Summer 2013

Distributed Beamforming and Nullforming: Frequency Synchronization Techniques, Phase Control Algorithms, and Proof-Of-Concept

Muhammad Mahboob Ur Rahman
University of Iowa

DISTRIBUTED BEAMFORMING AND NULLFORMING: FREQUENCY SYNCHRONIZATION TECHNIQUES, PHASE CONTROL ALGORITHMS, AND PROOF-OF-CONCEPT

by

Muhammad Mahboob Ur Rahman

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

August 2013

Thesis Supervisor: Prof. Raghu Mudumbai

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Muhammad Mahboob Ur Rahman

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Electrical and Computer Engineering at the August 2013 graduation.

Thesis Committee: _____
Raghu Mudumbai, Thesis Supervisor

_____
Soura Dasgupta

_____
Er-Wei Bai

_____
Anton Kruger

_____
Octav Chipara

To my family

ACKNOWLEDGEMENTS

In the name of ALLAH, the Most Gracious, the Most Merciful.

First and foremost, thanks to Almighty ALLAH for all HIS blessings on me.

I would then like to thank my adviser, Prof. Raghu Mudumbai, for his support, encouragement and confidence in my abilities. He has indeed spent quite a lot of time during last four years honing my professional and research skills. I am also indebtful to my co-adviser, Prof. Soura Dasgupta, for his time and efforts on me. In fact, I am grateful to both of them for their firm support and kind behavior in my difficult time. It means a lot to me!

I now want to acknowledge our collaborators at UCSB, Santa Barbara, Prof. Upamanyu Madhow and Dr. Francois Quitin for the helpful and valuable discussions. I am also thankful to my thesis committee members for their kindness, encourgement and support.

Of course, I would like to thank my friends (all over the world!) and labmates (at Iowa) for their enjoyable company and support. Need not to mention, my family, my wife and little Ahmad are the real assets for me, and my life is just nothing without them!

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

LIST OF ALGORITHMS

LIST OF THEOREMS

CHAPTER 1
INTRODUCTION

Recently, MIMO systems have attracted a lot of attention due to the famous result which states that the channel-capacity of a multi-antenna system scales linearly with the minimum of number of transmit, receive antennas[43],[11]. Physically, the increased channel capacity comes from the fact that MIMO systems provide spatial multiplexing gains. MIMO has already found application in next generation of cellular standards such as LTE as well as next generation of wireless access standards such as WiFi and Zig Bee.

Despite all the aforementioned benefits of MIMO systems, the form-factor constraint due to minimum antenna-separation requirement limits the maximum number of antennas that can be mounted on a transmitter or receiver. Distributed MIMO (DMIMO) offers one attractive way around this constraint: rather than place multiple antennas on a single device, in DMIMO systems, a number of devices in a wireless network collaboratively organize themselves into "virtual antenna arrays" and cooperatively obtain MIMO gains. Distributed MIMO techniques are especially attractive for wireless sensor networks (WSN) that by definition, consist of a large number of low-power sensor nodes organized into an ad-hoc network.

1.1 System model

This dissertation focuses on distributed MISO; this is a special subclass of distributed MIMO techniques where a virtual antenna array of transmitters com-

municates cooperatively to individual (single antenna) receivers. More specifically, we consider two canonical distributed MISO techniques: i) distributed beamforming, and ii) distributed nullforming. These techniques can be thought of as building blocks for more general MIMO precoding techniques. However, these techniques are also interesting and important in their own right.

1.1.1 Distributed beamforming

Distributed beamforming refers to a distributed MISO technique where the nodes in a VAA want to form a beam towards a distant receiver. This technique is especially attractive for WSNs because it allows inexpensive nodes with simple omnidirectional antennas to collaboratively emulate a highly directional antenna and focus their transmission in the direction of the intended receiver. This potentially offers large increases in energy efficiency: a VAA of N nodes can achieve an N^2-fold increase in the power at a receiver compared to a single node transmitting individually; conversely each node in a N-node VAA can reduce its transmit power by a factor of $\frac{1}{N^2}$ and still achieve the same overall signal power at the receiver compared to a single transmitter.

It is important to note that this is not just a reduction in the *per node* transmitted power simply because there are more nodes transmitting; this is also an increase in the *energy efficiency* of transmission: an N-node VAA can achieve the same received signal strength (RSS) at the receiver with as little as $\frac{1}{N}$ of the *total transmit power* required by a single node transmitting individually.

Physically this increased energy efficiency arises from the increased directivity of transmissions; signals from the individual transmitters combine constructively at the intended receiver and as a result a larger proportion of the transmitted power is concentrated in the direction of the intended receiver. This is illustrated in Fig. 1.1.

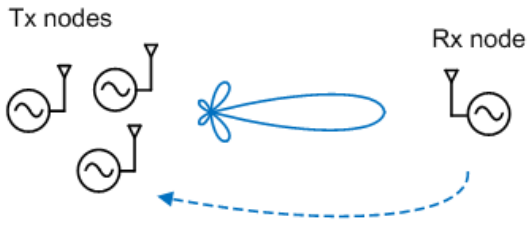

Figure 1.1: Energy efficient communication using distributed beamforming.

The key challenge in realizing the large potential gains from beamforming is in precisely synchronizing the RF signals. Each transmitter in general obtains its RF carrier signal from its own local oscillator. Even when two oscillators are set to the same nominal frequency, because of manufacturing tolerances and temperature variations, they would in general have a non-zero frequency offset with respect to each other. In addition all oscillators undergo random phase and frequency drifts modeled by Brownian motion in both, whose variance grows over time [71]. Finally, unlike a traditional phased array, a VAA made up of collaborating wireless sensor nodes does not have a regular and precisely known geometry; furthermore standard localization techniques such as GPS fall far short of the accuracy necessary to overcome this geo-

metric uncertainty for the purposes of beamforming. Thus, distributed beamforming requires highly sophisticated synchronization techniques that account for such uncertainties. We will discuss the framework for frequency synchronization and phase synchronization we have developed for distributed beamforming in Chapters 2-3 and Chapter 4 respectively.

Note that there are other cooperative transmission schemes that unlike distributed beamforming, do *not* require precise phase alignment. This includes all relaying and multi-hopping schemes where different transmitters use orthogonal space/-time/frequency channels so that their transmissions do not interfere with each other. In contrast, beamforming *depends* on transmitters interfering with each other in a carefully controlled way. Orthogonal cooperation schemes can provide diversity gains in fading channels, however, they cannot provide the energy efficiency gains achievable from beamforming.

1.1.2 Distributed nullforming

Distributed nullforming is a distributed MISO technique where VAA nodes want to steer a *null* towards a designated null target. To do this, VAA nodes cooperatively transmit a common message signal in such a way that their individual transmissions cancel each other at a designated null target. The technique of distributed nullforming has many potential applications including interference avoidance for increased spatial spectrum reuse [43], cognitive radio [67], physical-layer security [12] and so on.

Distributed nullforming requires precise control of the amplitude and phase of the radio-frequency signal transmitted by each cooperating transmitter to ensure that they cancel each other. This is an extremely challenging problem; as discussed above, each transmitter usually obtains its RF signal from a separate local oscillator (LO), and signals obtained from different LOs invariably have Brownian motion driven phase drifts due to manufacturing tolerances and temperature variations. The nullforming algorithm must estimate, track and compensate for the effect of these drifts. The synchronization requirements for distributed nullforming are even more challenging than for beamforming because (a) while beamforming is fairly robust and tolerant [34] to moderately large phase errors, nullforming is far more sensitive to even small phase errors, and (b) distributed beamforming can be accomplished with each transmitter having knowledge of only its own phase at the beam target, nullforming in general requires knowledge of all channels at all the transmitter nodes. The required frequency-locking mechanisms are discussed in detail in Chapters 2-3, while the phase control (gradient descent) algorithm is discussed in Chapter 5.

1.2 Organization of the thesis

The rest of this thesis is organized as follows.

Later chapters (except the last chapter) are devoted to 3 distinct topics:

- **Frequency synchronization.** Chapter 2 and 3 cover this topic in detail. Chapter 2 discusses in detail our proposed distributed consensus based algorithm for carrier synchronization. Chapter 3 discusses various novel DSP-centric

algorithms to carry out carrier frequency synchronization in baseband.

- **Distributed beamforming.** Chapter 4 summarizes the key points and presents significant results pertaining to our recent implementation of distributed beamforming on GNU-radio/USRP-based SDR platform.
- **Distributed nullforming.** Chapter 5 propoes a distributed gradient-descent based algorithm which causes VAA nodes to achieve a null at a designated null target.

Chapter 6 concludes with a discussion of possible topics for future work.

CHAPTER 2
CONSENSUS BASED RF CARRIER SYNCHRONIZATION

2.1 Motivation

2.1.1 Synchronization techniques in the literature

The synchronization problem has inspired researchers since long to develop multiagent consensus techniques [65], [63], [23], [31] and [53]. Moreover, naturally occuring biological phenomenon like the synchronized flashing of fireflies [33] have motivated other researchers to propose similar methods to achieve synchronization in wireless networks, [64]. Other examples include [46], [7] and [41], network time synchronization, [9], [8], and [15] and on-chip clock distribution, [21].

Before we present our proposed algorithm, we deem it necessary to discuss Master-slave architecure that has been a popular choice for frequency synchronization in the recent research work [34]. We have also used it during the early stages of our implementation of distributed beamforming on software-defined radio. Please refer to Chapter 3 and Chapter 4 for more details. Briefly speaking, in a Master-slave architecure, a designated "Master" node broadcasts a training signal; this signal is then used as a reference signal by the "Slave" nodes to correct for their respective frequency offsets. Master-slave architecture is widely used because of its simplicity and ease of implementation. However, it has several limitations, e.g., it is not scalable, a single "Master" node means its failure can cause network failure etc. Therefore, we present in this chapter, a distributed approach to synchronization problem which is

quite opposite to the centralized approach used by Master-slave architecture. Further, when only one node adjusts its frequency, as is done by slave nodes in a Master-Slave architecture, the frequency lock achieved by slave nodes which use PLLs is fundamentally local. Specifically, we enunciate a globally stable nonlinear consensus based algorithm that achieves carrier synchronization between two cooperating transmitters. We also present some preliminary results for the case when there are three, and in general N cooperating transmitters.

2.2 Two-node network: Analysis and results

Consider a two-node wireless network. Each node transmits to the other its sinusoidal carrier, and adjusts its frequency and phase to achieve global consensus. At that time, both carriers attain a common frequency that is an integer multiple of π/α, where α is a design parameter of our algorithm. In particular, depending on the value of α, arbitrary granularity for consensus frequency can be achieved at no practical expense. The steady state phase offset induced by the algorithm is either 0 or π. We provide a simple method, using which the transmitters can in principle determine whether a phase disparity of π exists between them and thus can correct it. Our algorithm thus represents a very substantial improvement on existing carrier synchronization methods that largely employ PLL technology, [30]. As is well known PLL's achieve only local carrier synchronization, so, PLL's will not converge from arbitrary initial errors. By contrast, despite having sinusoids in its update kernel our algorithm is globally stable.

2.2.1 The algorithm

Consider a two-node network with nodes in the set $\{i, j\} = \{1, 2\}$. Then assume that the i-th agent broadcasts a signal $\cos(\theta_i(t))$. Such an agent receives a signal $j \neq i$,

$$s_i(t) = A\cos(\theta_j(t)) + v_i(t), \tag{2.1}$$

where A reflects the attenuation suffered in the transmission from agent j to i and $v_i(t)$ is noise. In an uncluttered environment it is reasonable to assume that the attenuation suffered by both is the same.

The algorithm we propose is as follows. For $\beta, \alpha > 0$, $\{i, j\} = \{1, 2\}$ and $j \neq i$, the *i-th transmitter implements:*

$$\dot{\theta}_i(t) = \omega_i(t) \tag{2.2}$$

$$\dot{\omega}_i(t) = -\beta A \sin\left(\theta_i(t) - \theta_j(t) + \alpha\omega_i(t)\right). \tag{2.3}$$

Thus $\omega_i(t)$ represents the locally generated instantaneous frequency. The $\alpha\omega_i(t)$ term in the frequency update equation in (2.3) is extremely crucial to our synchronization algorithm, because it ensures the stability of the consensus solution. *Note the i-th node only has access to ω_i, θ_i and s_i.* An obvious discrete time counterpart of (2.2),(2.3) is as follows. For small time step $D(\omega)$:

$$\theta_i(t + D(\omega)) = \theta_i(t) + D(\omega)\omega_i(t), \tag{2.4}$$

$$\omega_i(t + D(\omega)) = \omega_i(t) - \beta D(\omega) A \sin\left(\theta_i(t) - \theta_j(t) + \alpha\omega_i(t)\right) \tag{2.5}$$

It is clear that the qualitative properties of (2.4, 2.5) approach those of (2.2, 2.3) for small time steps $D(\omega)$.

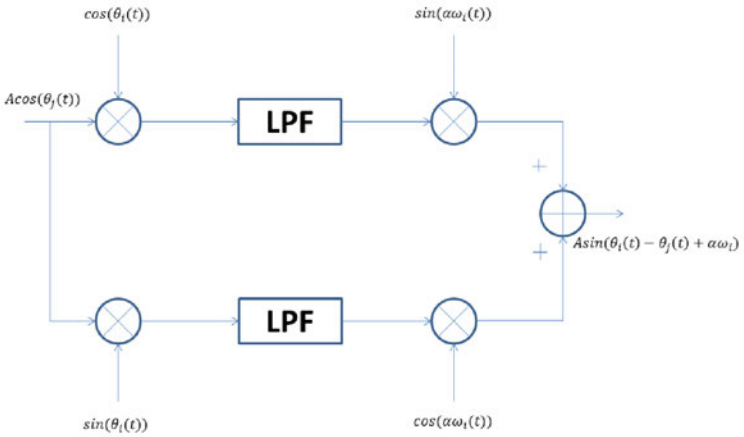

Figure 2.1: Implementation of the algorithm

2.2.2 The implementation considerations

We next turn to the implementation of the algorithm given that the information available to agent i, is the signal generated by the other agent, exemplified by (2.1), its instantaneous frequency $\omega_i(t)$ and the instantaneous phase $\theta_i(t)$. Observe the $\omega_i(t)$ are at RF, i.e. in hundreds of MHz, where as frequency disparities, induced say by oscillator drift are in few kHz. Thus, the difference between initial instantaneous frequencies are small compared to their values. In other words both $\sin(\theta_i(t) - \theta_j(t))$ and $\cos(\theta_i(t) - \theta_j(t))$ represent relatively low pass signals as compared to $\sin(\theta_i(t) + \theta_j(t))$ and $\cos(\theta_i(t) + \theta_j(t))$.

This emphasizes the need for stability that is not merely local, as initial frequency errors could be nontrivial, as high as few kHz.

Then consider the setting of Figure 2.1 where the blocks labeled LPF are low pass filters. Observe as $\theta_i(t)$ and $\omega_i(t)$ are available to agent i, one can generate:

$$\begin{aligned} g_i(t) &= 2s_i(t)\cos(\theta_i(t)) \\ &= A\left[\cos(\theta_i(t)-\theta_j(t))+\cos(\theta_i(t)+\theta_j(t))\right] \\ &+ 2v_i(t)\cos(\theta_i(t)). \end{aligned} \tag{2.6}$$

Thus, to within a noise perturbation, the low pass filtered version of g_i and by similar anlaysis, that of $2s_i(t)\sin(\theta_i(t))$ are respectively given by

$$A\cos(\theta_i(t)-\theta_j(t)) \text{ and } \quad A\sin(\theta_i(t)-\theta_j(t)).$$

Thus, as $\omega_i(t)$ is available, one can indeed as per Figure 2.1 generate the kernel in (2.3) to within a perturbing noise. *It is also clear that should the noise $v_i(t)$, be white Gaussian, so is the noise perturbing the kernel of (2.3).* Further the net noise is additive and is the original noise scaled by 2β.

In practice, (2.2, 2.3) or indeed (2.4, 2.5) will be implemented at baseband. Baseband techniques for RF carrier synchronization are discussed in detail in Chapter 3.

2.3 Comparison with Kuramoto

One well studied algorithm that can achieve frequency synchronization is the Kuramoto algorithm, [1]. Translated to a two node network it becomes for $\{i,j\} = \{1,2\}$, $i \neq j$,

$$\dot{\theta}_i(t) = \omega_i + K\sin(\theta_j(t)-\theta_i(t)), \tag{2.7}$$

where K is a coupling parameter, θ_i and ω_i are the instantaneous phase and the initial frequency estimate of node i's oscillator signal, respectively. Frequency synchronization is achieved if for all i, j

$$\dot{\theta}_i(t) = \dot{\theta}_j(t). \tag{2.8}$$

We now reveal a key difficulty with (2.7) to carrier frequency synchronization. For the $\dot{\theta}_i$ to synchronize we need

$$\omega_1 + K\sin(\theta_2 - \theta_1) = \omega_2 + K\sin(\theta_1 - \theta_2).$$

At the minimum this requires that

$$2K \geq |\omega_1 - \omega_2|. \tag{2.9}$$

In fact *the actual bound needed for stable synchronization is significantly higher, [24, 13].* Thus the coupling coefficient K must be large. The implementation of (2.7) would involve similar procedures as those for (2.2,2.3). As a consequence it can be verified that in this implementation the noise gets amplified by $2K$. Thus unless the initial frequencies are sufficiently close, not only will Kuramoto stabilize only at the expense of noise amplification, but will even lack a well defined consensus state, unless K is significantly large.

To be concrete consider the noisy version of the Kuramoto algorithm with for $\{i, j\} = \{1, 2\}$, $i \neq j$,

$$\dot{\theta}_i(t) = \omega_i + K\left(\sin(\theta_j(t) - \theta_i(t)) + e_i(t)\right), \tag{2.10}$$

where the $e_i(t)$ represent noise that obeys:

$$|e_i(t)| \leq \epsilon, \quad \forall\, t. \tag{2.11}$$

Then the following lemma lower bounds K for meaningful consensus.

Lemma 2.3.1. Consider (2.10) under (2.11). Suppose

$$K < \frac{|\omega_1 - \omega_2|}{2(1-\epsilon)}.$$

Then there is a pair of functions $e_i : \mathbb{R} \to \mathbb{R}$, $i \in \{1, 2\}$, obeying (2.11) such that for all t, there exists $t_1 > t$ such that

$$\left|\dot{\theta}_1(t_1) - \dot{\theta}_2(t_1)\right| \geq |\omega_1 - \omega_2|. \tag{2.12}$$

Proof. Without loss of generality choose $\omega_1 > \omega_2$. Also choose $e_1(t) = \epsilon$ and $e_2(t) = -\epsilon$ for all t. Then because of (2.16) one has:

$$\begin{aligned} \dot{\theta}_1(t) - \dot{\theta}_2(t) &= \omega_1 - \omega_2 + 2K\left(\epsilon - \sin(\theta_1(t) - \theta_2(t))\right) \\ &\geq |\omega_1 - \omega_2| - 2K(1-\epsilon) \\ &> 0. \end{aligned} \tag{2.13}$$

Thus for every t, there exists $t_1 > t$ such that $\sin(\theta_1(t_1) - \theta_2(t_1)) = 0$. Thus the result follows from (2.13). ■

Thus, unless

$$K \geq \frac{|\omega_1 - \omega_2|}{2(1-\epsilon)} \tag{2.14}$$

the frequency error repeatedly exceeds the initial value $|\omega_1 - \omega_2|$. Thus for meaning ful synchronization one must have (2.14). We will analyze the behavior when (2.14). To this end we first present a convergence result.

Lemma 2.3.2. Consider (2.7). Suppose

$$K \geq \frac{|\omega_1 - \omega_2|}{2}.$$

Then $\dot{\theta}_1(t) - \dot{\theta}_2(t)$ converges exponentially to zero, i.e. $\sin(\theta_1(t) - \theta_2(t))$ converges exponentially to

$$\frac{\omega_1 - \omega_2}{2K} \tag{2.15}$$

The next lemma shows that in this case the worst case synchronization error is proportional to the initial frequency disparity $|\omega_1 - \omega_2|$.

Lemma 2.3.3. Consider (2.10) under (2.11) and (2.14). Then there is a pair of functions $e_i : \mathbb{R} \to \mathbb{R}$, $i \in \{1, 2\}$, obeying (2.11), that has the following property. For every $\delta > 0$, and t, there exists a $t_1(\delta) > t$ such that:

$$\left|\dot{\theta}_1(t_1(\delta)) - \dot{\theta}_2(t_1(\delta))\right| \geq \frac{2|\omega_1 - \omega_2|\epsilon}{1 - \epsilon} - \delta. \tag{2.16}$$

Proof. We will choose $e_i(t) - e_2(t)$ to switch between $\pm\epsilon$. Suppose at a given t $e_i(t) - e_2(t) = \epsilon$ and $e_i(t)$ both constant. Apply Lemma 2.3.2 with ω_i replaced by $\omega_i - Ke_i$. Then for every δ, there exists a $t_2(\delta)$ such that

$$|2K \sin(\theta_1(t_2(\delta)) - \theta_2(t_2(\delta))) - (\omega_1 - \omega_2 + 2K\epsilon)| \leq \delta. \tag{2.17}$$

Now choose $e_i(t_2(\delta)) - e_2(t_2(\delta)) = -\epsilon$. Then from (2.10) and (2.14)

$$\begin{aligned}
\left|\dot{\theta}_1(t_2(\delta)) - \dot{\theta}_2(t_2(\delta))\right| &= |\omega_1 - \omega_2 + 2K\epsilon \\
&- 2K \sin(\theta_1(t_2(\delta)) - \theta_2(t_2(\delta)))| \\
&\geq 4K\epsilon - \delta \\
&\geq \frac{2|\omega_1 - \omega_2|\epsilon}{1 - \epsilon} - \delta.
\end{aligned}$$

On the other hand if at a given t $e_i(t) - e_2(t) = -\epsilon$ and $e_i(t)$ both constant. Apply Lemma 2.3.2 with ω_i replaced by $\omega_i + Ke_i$. Then for every δ, there exists a $t_3(\delta)$ such that

$$|2K\sin(\theta_1(t_3(\delta)) - \theta_2(t_3(\delta))) - (\omega_1 - \omega_2 - 2K\epsilon)| \leq \delta. \tag{2.18}$$

Now choose $e_i(t_3(\delta)) - e_2(t_3(\delta)) = -\epsilon$. Then from (2.10) and (2.14)

$$\begin{aligned} \left|\dot{\theta}_1(t_3(\delta)) - \dot{\theta}_2(t_3(\delta))\right| &= |\omega_1 - \omega_2 - 2K\epsilon \\ &- 2K\sin(\theta_1(t_2(\delta)) - \theta_2(t_2(\delta)))| \\ &\geq 4K\epsilon - \delta \\ &\geq \frac{2|\omega_1 - \omega_2|\epsilon}{1-\epsilon} - \delta. \end{aligned}$$

Thus, the result follows. ■

By contrast as shown in the next section, the equilibrium trajectories of (2.2, 2.3) are independent of β.

Now consider Figure 2.2. This depicts the frequency plot generated by our algorithm in a 2-node network with initial frequencies at 2000 and 2100 radians/sec, initial phase difference of $\pi/4$ radians, with $\beta = 1$, $\alpha = 1$ and demonstrates synchronization. Thus the noise amplification factor in this case is 2. On the other hand (2.7) would require a $K = 50$. Thus our algorithm synchronizes with a much smaller noise amplification factor than required by Kuramoto to even have a well defined consensus state, let alone stablity.

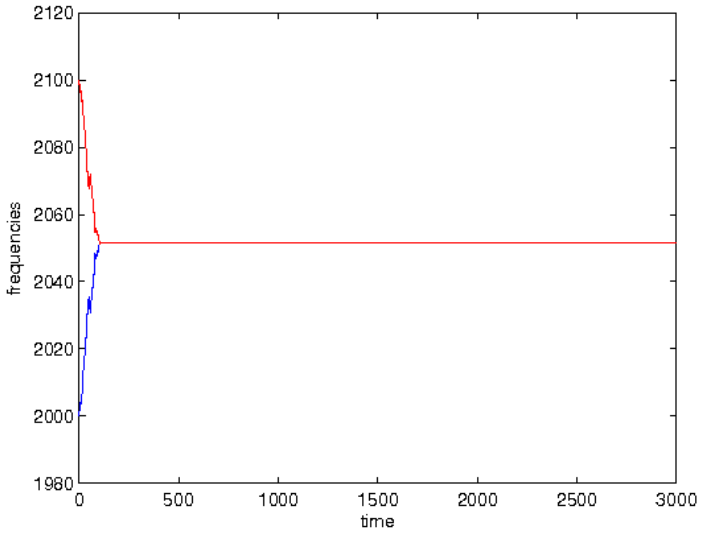

Figure 2.2: Frequency consensus in a 2-node network.

2.4 Locally stable consensus states

The two node algorithm is with $\alpha, \beta > 0$ to: for $i \in \{1, 2\}$ (2.2) and:

$$\dot{\omega}_1(t) = -\beta \sin\left(\theta_1(t) - \theta_2(t) + \alpha\omega_1(t)\right) \tag{2.19}$$

and

$$\dot{\omega}_2(t) = -\beta \sin\left(\theta_2(t) - \theta_1(t) + \alpha\omega_2(t)\right) \tag{2.20}$$

hold.

We say that consensus is achieved if there hold:

$$\sin\left(\theta_1(t) - \theta_2(t) + \alpha\omega_1(t)\right) = 0, \quad \forall t \tag{2.21}$$

and

$$\sin\left(\theta_2(t) - \theta_1(t) + \alpha\omega_2(t)\right) = 0, \quad \forall t, \tag{2.22}$$

We begin by characterizing the consensus states for the algorithm.

Theorem 2.1. *Consider (2.2) for $i \in \{1,2\}$, and (2.19) and (2.20) with $\beta, \alpha > 0$. The only equilibrium trajectories for this system are: for integers m and n, and any ϕ,*

$$\theta_1(t) = \frac{(m+n)\,\pi}{2\alpha}t + \phi + \frac{(m-n)\,\pi}{2}, \tag{2.23}$$

$$\theta_2(t) = \frac{(m+n)\,\pi}{2\alpha}t + \phi \tag{2.24}$$

$$\omega_1(t) = \omega_2(t) = \frac{(m+n)\,\pi}{2\alpha}. \tag{2.25}$$

Proof. Equilibrium requires that (2.21) and (2.22) hold and because of (2.19) and (2.20), for some ω_i^*, $i \in \{1,2\}$

$$\omega_i(t) = \omega_i^* \quad \forall t. \tag{2.26}$$

Thus, as $\theta_i(t)$ and $\omega_i(t)$ are both continuous, on this trajectory for some integer l, for all t, there holds:

$$\begin{aligned} &\theta_2(t) - \theta_1(t) + \alpha\omega_2^* = \theta_1(t) - \theta_2(t) + \alpha\omega_1^* + l\pi \\ \Leftrightarrow \quad &\theta_2(t) - \theta_1(t) = \frac{\alpha\,(\omega_1^* - \omega_2^*) + l\pi}{2}. \end{aligned} \tag{2.27}$$

Further because of (2.26) and (2.2) for all t and $i \in \{1,2\}$, on this trajectory for constant ϕ_i:

$$\theta_i(t) = \omega_i^* t + \phi_i. \tag{2.28}$$

Because of (2.27) this must mean that for some ω^*

$$\omega_1^* = \omega_2^* = \omega^*.$$

Finally for some integers m, n

$$\phi_2 - \phi_1 + \alpha\omega^* = n\pi$$

and

$$\phi_1 - \phi_2 + \alpha\omega^* = m\pi.$$

Thus (2.23) to (2.25) hold. ■

We next show through a linearized analysis that certain consensus frequency and phase combinations are locally unstable and some others are stable.

Theorem 2.2. *Consider (2.2) for $i \in \{1, 2\}$, (2.19) and (2.20) with $\beta, \alpha > 0$. Then the equilibrium trajectories characterized in Theorem 2.1 are locally exponentially stable iff m and n are both even. If either m and/or n is odd then the trajectory is unstable.*

Proof. For integer m, n, define

$$\tilde{\theta}(t) = \theta_1(t) - \theta_2(t) - \frac{(m-n)\,\pi}{2} \tag{2.29}$$

for $i \in \{1, 2\}$,

$$\tilde{\omega}_i(t) = \omega_i(t) - \frac{(m+n)\,\pi}{2\alpha} \tag{2.30}$$

and

$$\tilde{\omega}(t) = \omega_1(t) - \omega_2(t). \tag{2.31}$$

Evidently such a trajectory is a stationary trajectory if for all t

$$[\tilde{\theta}(t), \tilde{\omega}_1(t), \tilde{\omega}_2(t)]' = 0.$$

Further by subtracting (2.2) for $i = 2$ from (2.2) for $i = 1$, one obtains:

$$\dot{\tilde{\theta}}(t) = \tilde{\omega}(t), \tag{2.32}$$

Also from (2.19) one obtains:

$$\begin{aligned}
\dot{\tilde{\omega}}_1(t) &= -\beta \sin\left(\theta_1(t) - \theta_2(t) + \alpha\omega_1(t)\right) \\
&= -\beta \sin\Bigg(\theta_1(t) - \theta_2(t) - \frac{m-n}{2}\pi + \frac{m-n}{2}\pi \\
&+ \alpha\omega_1(t) - \frac{m+n}{2}\pi + \frac{m+n}{2}\pi\Bigg) \\
&= -\beta \sin\left(\tilde{\theta}(t) + m\pi + \alpha\tilde{\omega}_1(t)\right) \\
&= -(-1)^m \beta \sin\left(\tilde{\theta}(t) + \alpha\tilde{\omega}_1(t)\right)
\end{aligned} \tag{2.33}$$

Similarly, from (2.20) one obtains:

$$\begin{aligned}
\dot{\tilde{\omega}}_2(t) &= -\beta \sin\left(\theta_2(t) - \theta_1(t) + \alpha\omega_2(t)\right) \\
&= -\beta \sin\Bigg(\theta_2(t) - \theta_1(t) + \frac{m-n}{2}\pi - \frac{m-n}{2}\pi \\
&+ \alpha\omega_2(t) - \frac{m+n}{2}\pi + \frac{m+n}{2}\pi\Bigg) \\
&= -\beta \sin\left(-\tilde{\theta}(t) + n\pi + \alpha\tilde{\omega}_2(t)\right) \\
&= -(-1)^n \beta \sin\left(-\tilde{\theta}(t) + \alpha\tilde{\omega}_2(t)\right)
\end{aligned} \tag{2.34}$$

Linearizing (2.32), (2.33) and (2.34) around zero, we obtain (2.32),

$$\dot{\tilde{\omega}}_1(t) = -(-1)^m \beta \left(\tilde{\theta}(t) + \alpha\tilde{\omega}_1(t)\right) \tag{2.35}$$

and:

$$\dot{\tilde{\omega}}_2(t) = -(-1)^n \beta \left(-\tilde{\theta}(t) + \alpha\tilde{\omega}_2(t)\right). \tag{2.36}$$

We now consider two cases that exhaust all possibilities.

Case I: Either both m and n are even, or they are both odd. In this case subtracting (2.36) from (2.35) we get

$$\begin{bmatrix} \dot{\tilde{\theta}}(t) \\ \dot{\tilde{\omega}}(t) \end{bmatrix} = (-1)^m \begin{bmatrix} 0 & 1 \\ -2\beta & -\beta\alpha \end{bmatrix} \begin{bmatrix} \tilde{\theta}(t) \\ \tilde{\omega}(t) \end{bmatrix}. \tag{2.37}$$

Clearly (2.37) is exponentially stable iff m and thus n are both even, and is unstable if both are odd.

Case II: One among m and n is even and the other is odd. In this case because of the underlying symmetries we can without loss of generality assume that m is even and n is odd. Then (2.32), (2.35) and (2.36) become:

$$\begin{bmatrix} \dot{\tilde{\theta}}(t) \\ \dot{\tilde{\omega}}_1(t) \\ \dot{\tilde{\omega}}_2 1t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 \\ -\beta & -\beta\alpha & 0 \\ -\beta & 0 & \beta\alpha \end{bmatrix} \begin{bmatrix} \tilde{\theta}(t) \\ \tilde{\omega}_1(t) \\ \tilde{\omega}_2(t) \end{bmatrix}. \tag{2.38}$$

Now observe that

$$\begin{bmatrix} 0 & 1 & -1 \\ -\beta & -\beta\alpha & 0 \\ -\beta & 0 & \beta\alpha \end{bmatrix}$$

has determinant $2\beta^2\alpha \neq 0$. Further its trace is zero. Consequently it must have an eigenvalue in the open right half plane, and thus (2.38) is unstable.

■

Observe the stable frequencies are thus multiples of π/α. The potential steady state phase offsets (modulo 2π) are 0 and π. Sometimes, e.g. in a standard communications framework a phase difference that is an odd multiple of π is entirely acceptable. Nonetheless it would be useful to easily determine whether the achieved phase discrepancy is an odd multiple of π. The following lemma helps in detecting such a disparity, should it occur.

Lemma 2.4.1. With even integers m and n, consider

$$k = \frac{m+n}{2} \text{ and } l = \frac{m-n}{2}.$$

Then both k and l are integers and k is even iff l is even.

Proof. That with even m and n, k and l are integers is self evident. Now l is odd iff for some integer i,

$$\begin{aligned} & \frac{m-n}{2} = 2i+1 \\ \Leftrightarrow \quad & m-n = 2(2i+1) \\ \Leftrightarrow \quad & m = n+2(2i+1) \\ \Leftrightarrow \quad & k = \frac{m+n}{2} = n+(2i+1). \end{aligned}$$

Then the result follows as n is even. ■

In view of this lemma and Theorem 2.2 a phase offset that is an odd multiple of π will occur iff the locally consensus frequency one achieves is an odd multiple of π/α. Should that happen, one of the nodes can simply advance its phase by π and *de facto phase as well as frequency lock is achieved.*

2.5 Global Stability

In this section we prove the global stability of (2.2,2.3). As noted in the introduction this represents a substantial advancement over existing technology. To be specific current carrier synchronization between a transmitter and a receiver, is effected using standard PLL technology. In a PLL one node transmits its carrier to

a receiver, which adjusts its frequency/phase to match the transmitter's frequency. The transmitter does not adjust its carrier. Consequently, unless the phase and the frequency of the receiver are sufficiently close to that of the transmitter, frequency/phase lock will not eventuate. By contrast our algorithm requires both nodes to adjust their carriers, and forces them to achieve a consensus.

We observe that the system is autonomous. It is thus the type of system that is potentially amenable to analysis by Lasalle's Theorem, [19]. However, Lasalle's Theorem requires a positively invariant set that is compact. There are technical difficulties with this requirement, as even after consensus is achieved, the $\theta_i(t)$ do not belong to a compact set. To circumvent this difficulty we propose an alternative, related state space for which compactness is easier to prove. Indeed this is a fifth order state space for which the state elements z_i are as below.

$$z_1(t) = \sin(\theta_1(t) - \theta_2(t) + \alpha\omega_1(t)), \tag{2.39}$$

$$z_2(t) = \sin(\theta_2(t) - \theta_1(t) + \alpha\omega_2(t)), \tag{2.40}$$

$$z_3(t) = \cos(\theta_1(t) - \theta_2(t) + \alpha\omega_1(t)), \tag{2.41}$$

$$z_4(t) = \cos(\theta_2(t) - \theta_1(t) + \alpha\omega_2(t)), \tag{2.42}$$

and

$$z_5(t) = \omega_1(t) - \omega_2(t). \tag{2.43}$$

Under the two node system equations we obtain:

$$\begin{aligned} \dot{z}_1(t) &= \cos(\theta_1(t) - \theta_2(t) + \alpha\omega_1(t))\,(\omega_1(t) - \omega_2(t) + \alpha\dot{\omega}_1(t)) \\ &= z_3(t)\,(z_5(t) - \beta\alpha z_1(t))\,, \end{aligned} \tag{2.44}$$

$$\begin{aligned}\dot{z}_2(t) &= \cos(\theta_2(t) - \theta_1(t) + \alpha\omega_2(t))\,(\omega_2(t) - \omega_1(t) + \alpha\dot{\omega}_2(t)) \\ &= z_4(t)\,(-z_5(t) - \beta\alpha z_2(t))\,, \end{aligned} \tag{2.45}$$

$$\begin{aligned}\dot{z}_3(t) &= -\sin(\theta_1(t) - \theta_2(t) + \alpha\omega_1(t))\,(\omega_1(t) - \omega_2(t) + \alpha\dot{\omega}_1(t)) \\ &= -z_1(t)\,(z_5(t) - \beta\alpha z_1(t))\,, \end{aligned} \tag{2.46}$$

$$\begin{aligned}\dot{z}_4(t) &= -\sin(\theta_2(t) - \theta_1(t) + \alpha\omega_2(t))\,(\omega_2(t) - \omega_1(t) + \alpha\dot{\omega}_2(t)) \\ &= -z_2(t)\,(-z_5(t) - \beta\alpha z_2(t))\,, \end{aligned} \tag{2.47}$$

and

$$\dot{z}_5(t) = -\beta(z_1(t) - z_2(t)). \tag{2.48}$$

We now analyze the stability of the system represented by (2.44)-(2.48) regardless of its origins, i.e. the tie to our algorithm.

Lemma 2.5.1. With $z = [z_1, \cdots, z_5]' : \mathbb{R} \to \mathbb{R}^5$, consider the system represented by (2.49) to (2.53) below.

$$\dot{z}_1(t) = z_3(t)\,(z_5(t) - \beta\alpha z_1(t))\,, \tag{2.49}$$

$$\dot{z}_2(t) = z_4(t)\,(-z_5(t) - \beta\alpha z_2(t))\,, \tag{2.50}$$

$$\dot{z}_3(t) = -z_1(t)\,(z_5(t) - \beta\alpha z_1(t))\,, \tag{2.51}$$

$$\dot{z}_4(t) = -z_2(t)\,(-z_5(t) - \beta\alpha z_2(t))\,, \tag{2.52}$$

and

$$\dot{z}_5(t) = -\beta(z_1(t) - z_2(t)). \tag{2.53}$$

Then $z(t)$ is bounded and converges uniformly asymptotically to:

$$z_1 \equiv 0 \tag{2.54}$$

and

$$z_2 \equiv 0. \tag{2.55}$$

Proof. Since the system of equations under consideration is autonomous, asymptotic stability implies uniform asymptotic stability.

Observe first that (dropping the argument t)

$$\begin{aligned} \frac{d}{dt}\left(z_1^2 + z_3^2\right) &= 2\left(z_1\dot{z}_1 + z_3\dot{z}_3\right) \\ &= 2\left(z_1 z_3(t)\left(z_5(t) - \beta\alpha z_1(t)\right)\right. \\ &- \left. z_3 z_1(t)\left(z_5(t) - \beta\alpha z_1(t)\right)\right) \\ &= 0. \end{aligned}$$

Similarly:

$$\frac{d}{dt}\left(z_2^2 + z_4^2\right) = 0.$$

Thus $[z_1, \cdots, z_4]'$ is bounded. Thus the function

$$V(z(t)) = -\beta\left[z_3(t) + z_4(t)\right] + \frac{z_5^2(t)}{2} \tag{2.56}$$

is bounded from below. Further, there holds:

$$\begin{aligned}\dot{V}(z) &= -\beta\left(\dot{z}_3+\dot{z}_4\right)+z_5\dot{z}_5\\ &= -\beta\left(-z_1z_5+\beta\alpha z_1^2+z_2z_5+\beta\alpha z_2^2\right)-\beta z_5\left(z_1-z_2\right)\\ &= -\beta^2\alpha\left(z_1^2+z_2^2\right)\leq 0. \end{aligned} \tag{2.57}$$

Thus $V(z)$ and hence $z_5(t)$ is bounded. Consequently, z is in a compact set. Thus from Lasalle's Theorem, and (2.57) asymptotically z converges to the trajectory corresponding to $\dot{V}(z)\equiv 0$, i.e. to (2.54) and (2.55). ■

This brings us to our main theorem demonstrating global convergence.

Theorem 2.3. *Consider (2.2) for $i\in\{1,2\}$, (2.19) and (2.20) with $\beta,\alpha>0$. Then for some ϕ, for integers m and n, $[\theta_1(t),\theta_2(t),\omega_1(t),\omega_2(t)]'$ converges uniformly asymptotically to*

$$\begin{bmatrix} \frac{(m+n)\pi}{2\alpha}t+\phi+\frac{(m-n)\pi}{2}\\ \frac{(m+n)\pi}{2\alpha}t+\phi\\ \frac{(m+n)\pi}{2\alpha}\\ \frac{(m+n)\pi}{2\alpha}\end{bmatrix}.$$

Further for almost all initial conditions, m and n are even.

Proof. Under (2.39)-(2.43), (2.49)-(2.53) hold. Thus from Lemma 2.5.1, uniformly asymptotically, (2.21) and (2.22) hold. Then the result follows from Theorems 2.1 and 2.2. ■

Thus global consensus involving both phase and frequency lock is indeed achieved. Further in view of Theorem 2.2 this consensus ensues at an exponential rate. Uniform asymptotic stability also guarantees robustness to noise and delay.

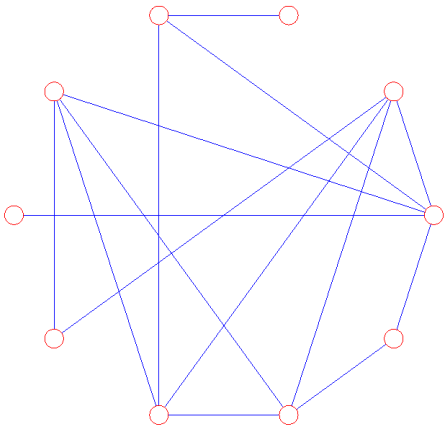

Figure 2.3: connectivity graph for simulations

2.6 N-node network: Preliminary results

2.6.1 Simulation results

We now present some simulation results to demonstrate the working of the consensus algorithm in an N-node network. For our simulations, we consider the $N = 10$ node network with the connectivity graph shown in Fig. 2.3. For simplicity we used a symmetric graph i.e. $A_{ij} = A_{ji}$, $\forall i, j$. Initially the frequencies are randomly chosen from the range 0 to 10 kHz. This is a typical range for the relative frequency offset of two oscillators with a frequency error of 10 ppm operating in the 1 GHz spectrum. The gains A_{ij} were chosen randomly from a 10 dB range.

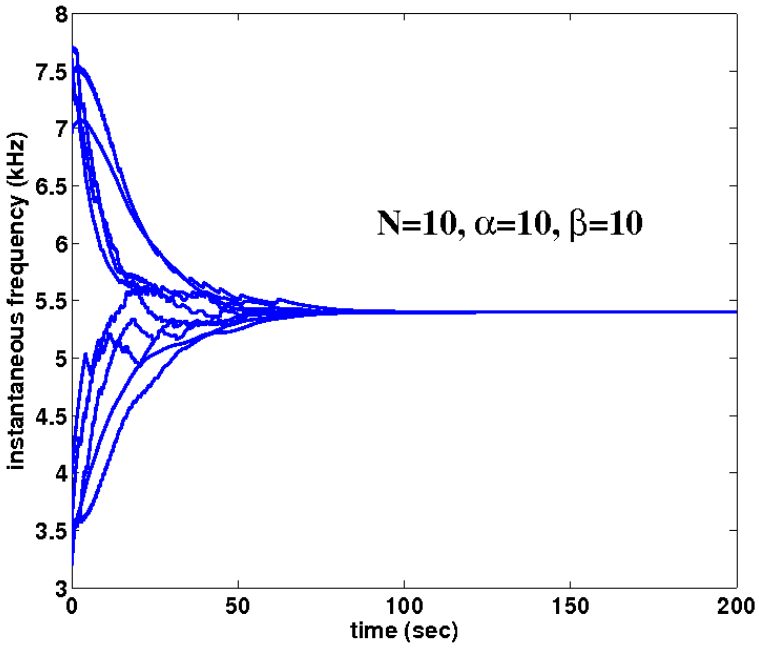

Figure 2.4: Frequency consensus in a 10-node network

2.7 Conclusion

We have proposed a consensus based RF carrier synchronization algorithm involving two transceiving units. Our algorithm achieves frequency lock globally and exponentially. Further it also achieves phase synchronization in the following sense. Asymptotically it induces the two transmitters to be either in phase, or out of phase by 180 degrees. This constitutes a significant advance over existing carrier synchronization technology which is largely based on Phase Locked Loops that only achieve lock locally.

CHAPTER 3
DSP-CENTRIC ALGORITHMS FOR CARRIER SYNCHRONIZATION

In this chapter, we consider the problem of carrier frequency synchronization among the nodes in a WSN from implementation viewpoint. Inline with the modern wireless transceiver architecture, we propose to estimate and compensate for the frequency offsets digitally in baseband and demonstrate it on software-defined radio testbed (see next chapter).

The key idea behind our implementation (to be discussed in more detail in Chapter 4) is that while the RF signals transmitted by the WSN nodes are themselves not suitable for digital processing, the clock offsets between oscillators that are nominally set to the same frequency are typically quite small. For instance, even very cheap crystal oscillators [42] have worst-case frequency deviations on the order of ± 10 parts per million of the nominal center frequency. In our experimental setup, we used center frequencies around 900 MHz, and thus our clock offsets can be expected to be no greater than 9 kHz or so. In fact, our measurements with the oscillators on the USRP boards showed clock drifts that seldom exceeded 4 kHz. Furthermore, these offsets remained roughly constant over time-scales on the order of hundreds of milliseconds.

Thus, as long as we are working with *relative offsets between two oscillators*, the frequencies are small enough and their time-variations slow enough that they can be tracked and compensated in software using low-rate DSP techniques.

Once carrier frequency synchronization is achieved, the nodes in WSN can act together as a virtual antenna array to either beamform or nullform to a designated receiver. While beamforming acheives energy-efficient communication and nullforming achieves the null at a designated null target, together the two can be thought of pre-requisite to ground-breaking spatial-multiplexing distributed MIMO techniques.

Different protocols have been developed that solve the problem of carrier frequency synchronization among WSN nodes in ways that represent different tradeoffs between in-network coordination, feedback from the receiver and so on. For instance, under schemes using a master-slave architecture [34], there is a designated master node that supplies the reference signal $c_0(t)$, whereas under round-trip synchronization schemes [45], the receiver (virtual array target) itself implicitly provides the reference signal. Alternatively, WSN nodes can use an external reference such as the signal from a GPS satellite if it is available. Transmitters can very well use BTS FCCH signals to lock to a stable reference signal [26]. Each of these alternatives have their advantages and disadvantages. For instance, uninterrupted availability of a GPS synchronization signal may not be a good assumption for indoor networks or where cost and form-factor constraints preclude using dedicated GPS modules on each node. Similarly having the receiver send a reference carrier signal eliminates the need for a separate Master node, but the reference signal from a distant receiver is likely to be more noisy as compared to a signal from a Master node co-located with the Slaves. Nevertheless, the DSP-centric architecture for carrier frequenyc synchronization proposed in this thesis is applicable to all of the schemes mentioned

above.

In general, the overheads associated with the synchronization process has costs that must be weighed against the benefits available from beamforming. One of the important goals of our implementation is precisely to show that these overhead costs are modest even without expensive custom designed hardware. Specifically we used the inexpensive oscillators [42] that come standard with the Universal Software Radio Peripherals (USRP); these have frequency offsets on the order of ± 10 parts per million. In contrast, high quality ovenized oscillators with frequency tolerance of around 20 parts per *billion* are now available [51] for around 400 dollars. Highly stable chip-scale atomic clocks [66] are also now coming closer to commercial feasibility. As these high-quality oscillators become more widely used in commodity wireless hardware, the overheads associated with carrier synchronization will become correspondingly smaller and this will make cooperative techniques such as distributed beamforming even more attractive over an increasing range of frequencies.

3.1 Two parallel sub-processes at slaves

In our setup, these are the Slave nodes (WSN nodes) that actually constitute the virtual antenna array; therefore, in our implementation, most of the DSP involved in frequency synchronization and beam/null steering occurs at the Slave nodes.

A key feature of our implementation is that the virtual array operation of WSN nodes is achieved by means of two decoupled sub-processes that run independently and concurrently. Roughly speaking, the first sub-process compensates for the frequency

offsets Δf_i among the WSN nodes; then a second sub-process is used to either form a beam or steer a null towards a designated receiver by adjusting the transmitters' phases $\Delta\phi_i$.

1. **Frequency synchronization.** In this sub-process, each transmitter locks its oscillator on to a shared reference signal. The purpose of this sub-process is to ensure that the transmitters all have RF signals with the same frequency and a fixed (but unknown) phase relationship with each other. The LO frequency offsets that can occur in typical software-defined radios can range up to several kHz, making the frequency synchronization of the transmit nodes especially challenging.

   We have exploited master-slave architecture to achieve carrier frequency synchronization in the earlier stages of our work. We then advanced our setup to round-trip carrier frequency synchronization mechanism. Specifically, we report the following contributions in this chapter:

   - **Open-loop frequency synchronization using costas loop.** In the early stages of our work, we used open-loop master-slave architecture to achieve frequency synchronization among cooperating transmitters. A designated (Master) transmitter provided a continuous training signal to which other nodes in WSN (Slaves) locked-into using so-called baseband costas loop.This method helped us to quickly prototype and demonstrate the energy-efficiency gains due to beamforming. Nevertheless, this architecture has its limitations. For example, we need a dedicated always-On

Master transmitter which is a not an efficient use of power and resources.

- **Closed-loop frequency synchronization using extended kalman filter.** Keeping in mind the limitations of master-slave architecture, we have ported our implementation to use closed-loop scheme for frequency synchronization. Specifically, the designated receiver sends periodic feedback packets to transmitters who then use extended kalman filters (EKF) to estimate and compensate for their frequency offsets w.r.t receiver itself in baseband. Evidently, this method of synchronization is superior to the open-loop method in terms of power-saving at receiver (receiver does not need to be always On), resources (no dedicated master transmitter is needed) and is immune to any instability issues (inherent in Costas loop).

The above sub-process ensures that the Slave nodes have carrier signals that are frequency-locked to some common reference; though they still have unknown but fixed relative phase offsets. We can now exploit the transmitters' phases to steer either a beam or a null to a designated receiver. A brief description of what will be coming in next two chapters is worth mentioning:

2. **Beam/Null steering.** This sub-process adjusts the phase relationship between the transmitters in such a way that their transmitted signals either add up constructively (called beamforming), or, destructively (called nullforming) at the intended receiver. Specifically, transmitters can choose between the following two algorithms at run-time to form a beam or a null towards a receiver:

- **Beamforming using 1-bit feedback algorithm.** The 1-bit feedback algorithm is a gradient ascent algorithm in nature which makes sure that the Slave nodes' transmissions are eventually aligned in phase at the Receiver. More details can be seen in Chapter 4.
- **Nullforming using gradient descent algorithm.** A gradient descent-based algorithm is used to achieve a null at the receiver using the received power as cost (objective) function. More details can be seen in Chapter 5.

The main motivation for the two decoupled sub-processes described above is simplicity. To be concrete, let us consider the case of distributed beamforming. The 1-bit algorithm is an elegant method which is easy to implement and has low overhead. While it can be modified to provide both frequency and phase synchronization [58], it cannot handle the significant frequency drifts that we encounter in our prototype, especially given the large latencies in the feedback channel. Thus, the frequency synchronization sub-process estimates and eliminates the frequency offsets among transmitters, and therefore, allows the simple 1-bit algorithm to achieve and maintain phase coherence among transmitters' signals as received at receiver.

3.2 Open-loop frequency synchronization using costas loop

Fig. 3.1 shows a schematic representation of our system. The goal of this frequency synchronization process is to lock the RF signals transmitted by the Slave nodes to a common reference clock signal supplied by the Master node. This serves to compensate for the clock offsets between the oscillators at the Slave nodes.

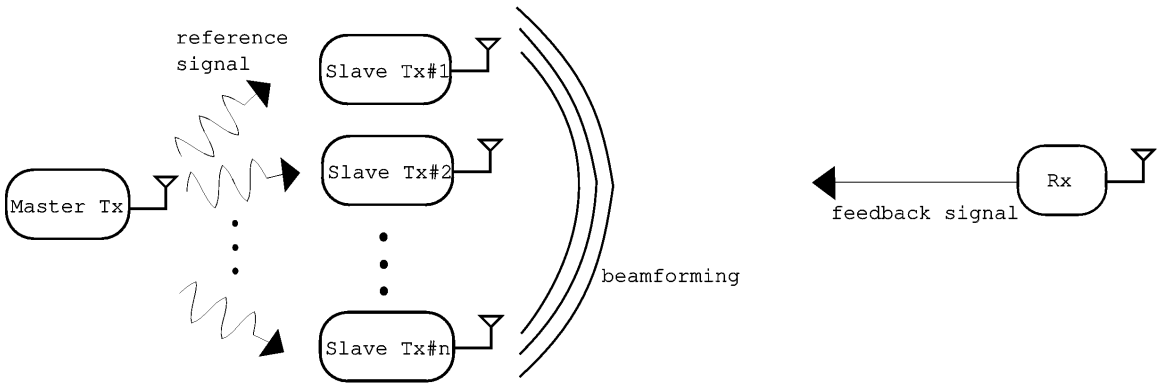

Figure 3.1: master-slave architecture for frequency locking.

Conceptually the frequency-locking problem can be formulated as follows. Given a reference signal $c_0(t) = \cos(2\pi f_1 t)$ from the Master node (i.e. a sinusoid at frequency f_1), and the pair of local oscillator signals $c_i(t) = \cos(2\pi(f_1 + \Delta f_i)t + \Delta\phi_i)$ and $s_i(t) = \sin(2\pi(f_1 + \Delta f_i)t + \Delta\phi_i)$ at Slave node i, we wish to digitally synthesize an RF signal $r_i(t) = \cos(2\pi f_2 t + \theta_i)$ at Slave i.

Note that the signals $r_i(t)$ at Slave i can have an arbitrary phase offset θ_i with each other, but must be locked to the same frequency f_2. The Slave nodes use the signals $r_i(t)$ to beamform/nullform to the receiver. As will be discussed in Section 4.3.1 of next chapter, because of the duplexing constraints, Slave nodes must transmit and receive on two different frequencies. So, slaves nodes transmit their common message to the receiver on frequency f_2 while they receive the reference signal from Master on frequency f_1.

In our setup, we used a modified baseband version of the classic Costas loop at the slave nodes to track the frequency *offset* between the reference signal from

the Master node and the Slave's local oscillator. This baseband loop is shown in Fig. 3.3 and it works as follows; the input to the baseband loop is the complex signal $\exp(j\phi(t))$ which represents the pair of signals $\cos\phi(t)$ and $\sin\phi(t)$, where for Slave node i, $\phi(t) = 2\pi\Delta f_i t + \Delta\phi_i$. These signals are obtained as the in-phase and quadrature components by downconverting the reference signal $c_0(t)$ using the local carrier signals $c_i(t))$ and $s_i(t)$ respectively as shown in Fig. 3.2.

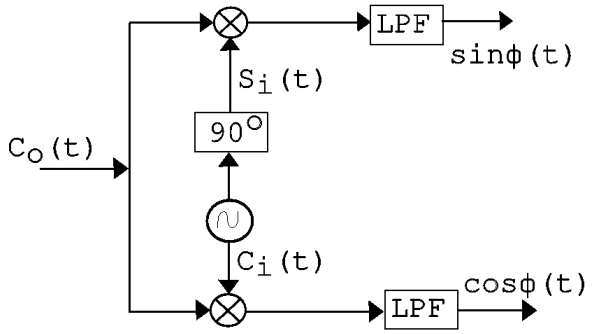

Figure 3.2: Slave i's oscillator offset with reference signal.

The complex signal $\exp(j\hat{\phi}(t))$ is the output of a digital VCO with the frequency sensitivity K_1, and therefore we have by definition

$$\hat{\phi}(t) = K_1 \int_{-\infty}^{t} e(\tau)d\tau \tag{3.1}$$

The "error signal" $e(t)$ is obtained from the difference of $\phi(t)$ and $\hat{\phi}(t)$ as shown

in Fig. 3.3, and this relationship can be written as

$$\begin{aligned} e(t) &= \cos\left(\phi(t) - \hat{\phi}(t)\right) \sin\left(\phi(t) - \hat{\phi}(t)\right) \\ &= \frac{1}{2} \sin\Big(2\big(\phi(t) - \hat{\phi}(t)\big)\Big) \end{aligned} \tag{3.2}$$

Equation (3.2) is mathematically equivalent to the classic Costas loop [10], though our implementation shown in Fig. 3.3 is quite different from the traditional RF loop. Over time, the loop makes the "error signal" $e(t)$ very small, and therefore makes $\hat{\phi}(t)$ close to $\phi(t) \equiv 2\pi\Delta f_i t + \Delta\phi_i$. In other words, this baseband loop at Slave i tracks the frequency offset Δf_i between the local oscillator signal of Slave i and the reference signal $c_0(t)$.

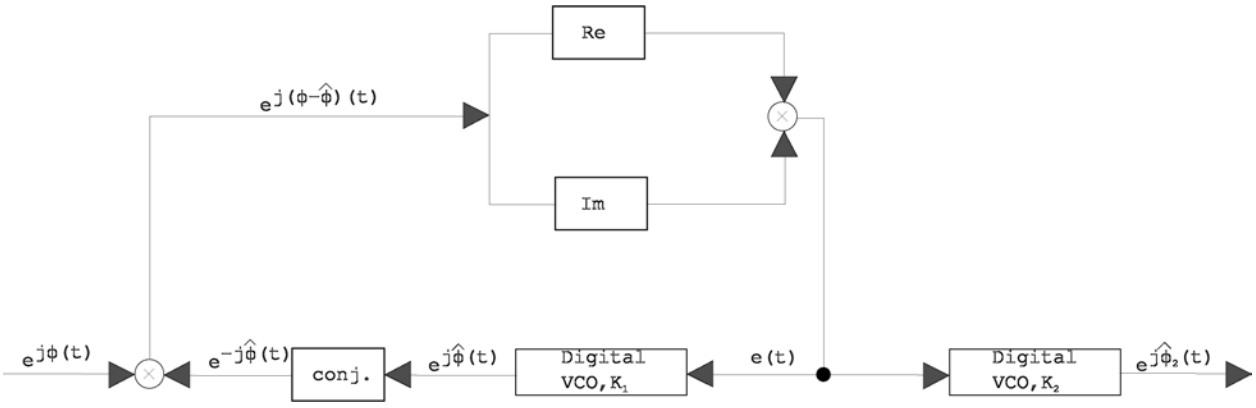

Figure 3.3: Modified baseband Costas loop for frequency-locking.

The Slave node i is now in a position to generate frequency-locked RF signals at frequency f_1 simply by upconverting $\cos\hat{\phi}(t)$ and $\sin\hat{\phi}(t)$ using the in-phase and quadrature local oscillator signals $c_i(t)$ and $s_i(t)$ respectively. However, to talk to

the receiver, slaves want to generate frequency-locked carrier signals not at the same frequency f_1 as the reference signal $c_0(t)$, but rather at a *different frequency* f_2 as discussed earlier.

In order to accomplish this, we use the fact that PLL-frequency synthesizers [52] used to obtain RF signals at different frequencies can be well-modeled as frequency-multiplying devices. Thus if Slave i generates an RF carrier signal at frequency f_2 from the same underlying oscillator used to generate the signals $c_i(t)$ and $s_i(t)$ at frequency f_1, the resulting signals will have frequency offsets given by $\frac{f_2}{f_1}\Delta f_i$. In order to correct for these offsets, we need to use $\cos\hat{\phi}_2(t)$ and $\sin\hat{\phi}_2(t)$ obtained from the *scaled* offset estimate $\hat{\phi}_2(t)$ from the second VCO as shown in Fig. 3.3; this scaled estimate can be written as

$$\hat{\phi}_2(t) = K_2 \int_{-\infty}^{t} e(\tau)d\tau \equiv \frac{K_2}{K_1}\hat{\phi}(t) \tag{3.3}$$

In the above, the VCO sensitivites $K_1,\ K_2$ must be chosen to satisfy $\frac{K_2}{K_1} = \frac{f_2}{f_1}$.

Note that this frequency-multiplication process may produce an unknown phase offset θ_i in the carrier signals at frequency f_2; however, this offset is constant and is easily compensated for by the beam/null steering algorithms which iteratively update the transmitters' phases.

3.3 Closed-loop frequency synchronization using extended kalman filter

Consider again the schematic representation of our system as shown in Fig. 1.1. The important distinction is that the receiver itself acts as a Master to provide a reference to the transmitters. Specifically, receiver node regularly broadcasts feedback

messages; the transmitter nodes use these messages to estimate and correct for their frequency and phase offsets Δf_i, $\Delta \phi_i$ w.r.t receiver. This approach is quite general, and is broadly applicable to WiFi, Zigbee and other packet wireless networks.

Figure 3.4 shows the the time-slotting model for the transmit nodes. Every time a feedback message is received from the receiver, the transmit node i will use the feedback message to make an estimation of its frequency and phase offset Δf_i and $\Delta \phi_i$ w.r.t receiver. These estimations will be used by the transmit nodes' extended kalman filters (EKFs) to predict and compensate for the LO offset of transmitter i w.r.t receiver until the next feedback message is received.

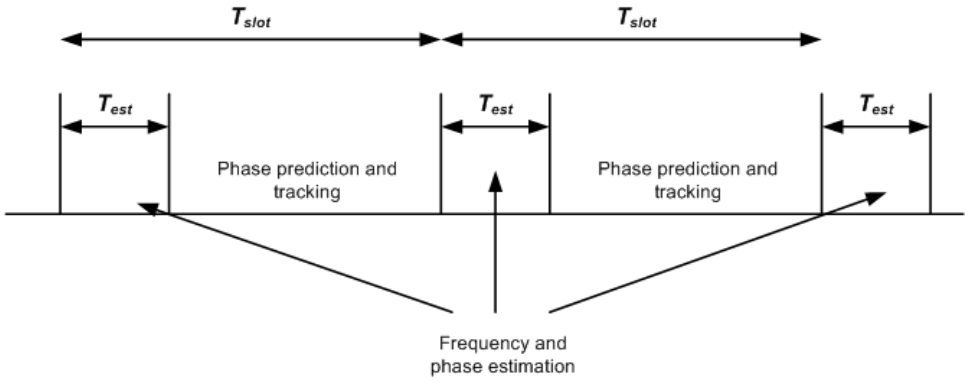

Figure 3.4: Time-slotting model for frequency and phase offset estimation at transmitters.

Again, the information link and feedback link can use the same frequency band using a medium access control mechanism enabling time sharing, e.g., TDMA, or we

can employ FDMA, with the feedback link and information link being at different frequencies. While our architecture applies to both scenarios, we employ frequency division multiplexing in our prototype. Thus, the transmit nodes use the reference signal (from receiver) at one frequency to synthesize a synchronized RF signal at a different frequency. Again, we assume that both carrier frequencies are synthesized from the same crystal oscillator, hence there is a known multiplicative relationship between them which also applies to the frequency offsets between two nodes that we wish to estimate and correct for.

3.3.1 Fundamental limits of frequency and phase estimation

We now focus our attention on quantifying the performance limits of frequency synchronization sub-process. Specifically, transmit nodes estimate their frequency and phase offsets w.r.t receiver using periodic feedback packets of duration T_{est} broadcast by the receiver; these noisy estimates are then passed to an EKF which filters out the noise. The time between these feedback packets is denoted by T_{slot}. Let us discuss what insights the Cramer-Rao Lower Bound (CRLB) for one-shot frequency/phase estimation provides regarding the desirable regime of operation for the frequency synchronization sub-process. These insights are then verified by simulations and experiments quantifying EKF performance [49].

Consider the process of obtaining one-shot frequency and phase estimates using a noise-corrupted reference signal received by a transmitter over the training epoch of duration T_{est} in one time-slot. Let $a(t) = A\exp(j\phi(t)) + n(t), t \in [0, T_{est}]$

which is the complex baseband waveform corresponding to one feedback packet upon demodulation using the LO signal of the transmit node. The post-integration SNR of this signal is defined as $\text{SNR} \equiv \frac{A^2 T_{est}}{2N_0}$, where N_0 is the power spectral density of the white noise process $n(t)$. The CRLBs for this one-shot phase and frequency estimation process are well-known in the literature [55, 27]: if ϕ_{err} and f_{err} respectively denote the one-shot phase and frequency estimation errors, we have

$$\begin{aligned} \sigma_\phi^2 \doteq E\left[\phi_{err}^2\right] &\geq \frac{2}{\text{SNR}} \\ \sigma_f^2 \doteq E\left[f_{err}^2\right] &\geq \frac{3}{2\pi^2 T_{est}^2 \text{SNR}} \end{aligned} \tag{3.4}$$

Consider now the phase error that results when transmitters use one-shot frequency and phase estimates from the training interval to predict and correct for the frequency and phase offsets of their oscillators over the subsequent time slot. The variance of the resulting error $\phi(t) - \hat{\phi}(t)$ between the predicted phase offset $\hat{\phi}(t)$ and actual phase offset $\phi(t)$ of the transmitter with the reference signal grows with time and its value at the end of the time-slot can be written as

$$\begin{aligned} E\left[\left(\phi(t) - \hat{\phi}(t)\right)^2\right]_{t=T_{slot}} &= \sigma_\phi^2 + T_{slot}^2 (2\pi\sigma_f)^2 \\ &\geq \frac{2}{SNR}\left(1 + \frac{3}{\eta^2}\right). \end{aligned} \tag{3.5}$$

When the duty cycle of the estimation process is small i.e. $\eta \equiv \frac{T_{est}}{T_{slot}} << 1$, then the second term in (3.5) dominates; in this setting, *one-shot frequency estimates are highly unreliable* as compared to the phase estimate.

Now consider an alternative approach to the frequency estimation problem. Instead of doing one-shot frequency estimates, we can also estimate frequency by

using two one-shot phase estimates in two successive training epochs T_{slot} seconds apart. In other words, we consider the frequency estimate $\tilde{f} \doteq \frac{\hat{\phi}(T_{slot})-\hat{\phi}(0)}{2\pi T_{slot}}$. This estimate has the variance

$$\mathrm{var}(\tilde{f}) = \frac{2\sigma_\phi^2}{(2\pi T_{slot})^2} \geq \frac{1}{\pi^2 \mathrm{SNR} T_{slot}^2}, \tag{3.6}$$

and this variance can be significantly smaller than the one-shot frequency variance σ_f^2 in (3.4). This suggests that we might be better off dispensing with one-shot frequency estimates altogether, and rely on averaging phase estimates over multiple time slots to get good frequency estimates. Indeed, this approach, implemented using a Kalman filter, is what is employed in [5]. However, using phase estimates alone for both phase and frequency tracking requires access to *unwrapped* phase estimates. This in turn requires that the frequency error in our estimate is small enough that 2π ambiguities in phase do not appear over the slot duration T_{slot} between successive training bursts. For the low-quality oscillators in our software-defined radios, the frequency drift is severe enough that satisfying this assumption would require excessive overhead.

In order to circumvent the preceding phase unwrapping problem, we employ crude one-shot frequency estimates to complement the phase estimates. These one-shot frequency estimates need only be good enough to avoid phase unwrapping errors over a single time-slot; in other words, we want $\sigma_f T_{slot}$ that is not too much larger than unity. Plugging this into (3.4), we obtain the following rule of thumb.

CRLB-based rule of thumb: $\frac{T_{slot}}{T_{est}} \approx k\sqrt{SNR}$, where $k = \sqrt{\frac{2}{3}}\pi$. Interestingly, this requirement only applies to the ratio $\frac{T_{slot}}{T_{est}}$ or equivalently to the duty-cycle of the training signal, not individually to T_{slot} or T_{est}. Note that this requirement is only

meant to provide very rough guidance. More detailed design insights as obtained via numerical simulations and experiments can be found in [49].

3.3.2 Frequency synchronization

The frequency synchronization sub-process is divided into three stages. In the first stage, the transmit node, upon receipt of each feedback packet, makes a measurement of its LO frequency and (wrapped) phase offset relative to the receiver using a blind estimation algorithm. In the second stage, the EKF uses these LO frequency and wrapped LO phase offset measurements to keep track of the unwrapped LO phase offset. In the third stage, the transmit node compensates for the LO offset based on the latest LO frequency and phase offset values as predicted by the EKF.

The blind estimation algorithm used in the first stage depends on the modulation format used for the feedback message. For most classical modulation formats (PSK, QAM, GMSK etc.), these algorithms transform the feedback message into a pilot tone, whose frequency can easily be estimated with classical frequency estimation theory. After compensating the feedback message for the LO frequency offset, the LO phase offset can easily be measured by correlating the feedback message with the (known) message header using matched filter. This will yield a *wrapped* measurement for the LO phase offset. A blind estimation algorithm for the case when receiver sends GMSK feedback messages is described in detail in Appendix of [49].

3.3.3 EKF state-space model

We use the following discrete state-space model for the LO offsets of each transmit node relative to the receiver.

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k \tag{3.7}$$

where $\mathbf{x}_k = [\phi_k, \omega_k]^T$ is the LO phase and angular frequency offset of the transmit node with respect to the receive node at time-slot k (where $\omega_k = 2\pi\Delta f_k$). The state update matrix $\mathbf{F}$ is defined by

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

and T_{slot} is the period of the feedback messages. Note that if aperiodic feedback messages are considered, T_{slot} is not fixed and the state update matrix $\mathbf{F}$ is allowed to be time-varying. The process noise vector $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}(T_{slot}))$ is the noise that causes the LO phase and frequency offset to deviate from their nominal value[1].

We use the following measurement model for the blind LO offset estimation algorithm which provides the inputs for the EKF.

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{3.8}$$

where

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

[1]The model described in (3.7) is accurate when modeling static scenarios, where the nodes do not move. In the case of mobile scenarios, a three-state model can be used to include the effects due to kinematics (which produces Doppler shift) in the LO model [5].

and $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$ is the additive white Gaussian measurement noise. Note that (3.8) defines a *non-linear* measurement model reflecting the fact that the blind estimation algorithm yields only an estimate of the *wrapped* phase offset.

In our work, we have borrowed the model for the process noise covariance matrix $\mathbf{Q}$ from [71, 5]. The state-space noise covariance matrix is defined by

$$\mathbf{Q}(T_s) = \omega_c^2 q_1^2 \begin{bmatrix} T_s & 0 \\ 0 & 0 \end{bmatrix} + \omega_c^2 q_2^2 \begin{bmatrix} \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^2}{2} & T_s \end{bmatrix} \tag{3.9}$$

where ω_c is the carrier frequency and T_s is the sample period. The parameters q_1^2 and q_2^2 are the process noise parameters that correspond to white frequency noise and random walk frequency noise, respectively. For a class of oscillators, these two parameters can be obtained by using the Allan variance.

The Allan variance is a tool to characterize the frequency stability of an oscillator, under the presence of various noise sources. It is mathematically defined as

$$\sigma_y^2(\tau) = \frac{1}{2\tau^2} \left\langle (\phi(t+2\tau) - 2\phi(t+\tau) + \phi(t))^2 \right\rangle_t \tag{3.10}$$

where $\phi(t)$ is the LO phase offset at time instant t with respect to some absolute reference. By applying equation (3.12) to the state-space model (3.7) and the noise model (3.11), it is shown in [71] that the following theoretical model can be obtained for the Allan variance:

$$\sigma_y^2(\tau) = \frac{q_1^2}{\tau} + \frac{q_2^2 \tau}{3} \tag{3.11}$$

The Allan variance can also be measured experimentally by sending a pilot tone with a transmitter, and by recording the received pilot tone (which will contain a certain

LO clock offset). By entering the unwrapped phase of the received pilot tone in (3.12) for various values of τ, it is possible to obtain an experimental curve for the Allan variance. By fitting experimental Allan variance measurements to the theoretical model (3.13), it is possible to obtain values for q_1^2 and q_2^2. For the software-defined radios used in our experimental setup (to be described in next chapter), the obtained parameters are $q_1^2 = 8.47 \times 10^{-22}$ and $q_1^2 = 5.51 \times 10^{-18}$.

Finally, the measurement noise matrix corresponding to our setup is $\mathbf{R} = \begin{bmatrix} 0.05 \cdot \pi/180 & 0 \\ 0 & 1.5 \cdot 2\pi \end{bmatrix}$.

The equations that determine the EKF evolution are split into two stages: an *update phase* and a *prediction phase*. The update phase corrects the current state estimate given the last measurement $\mathbf{z}_k$, and is mathematically defined as

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{h}\left(\mathbf{x}_{k|k-1}\right) \tag{3.12a}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R} \tag{3.12b}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{3.12c}$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k \tag{3.12d}$$

$$\mathbf{P}_{k|k} = \left(\mathbf{I}_2 - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_{k|k-1} \tag{3.12e}$$

The matrix $\mathbf{H}_k$ is the Jacobian of the function $\mathbf{h}$:

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

The prediction phase gives an estimation of the future state $\mathbf{x}_{k+1|k}$ to be used in the

update phase of next EKF cycle:

$$\mathbf{x}_{k+1|k} = \mathbf{F}\mathbf{x}_{k|k} \tag{3.13a}$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^T + \mathbf{Q} \tag{3.13b}$$

For each EKF cycle, the values contained in the vector $\mathbf{x}_{k|k}$ give a filtered estimate for the unwrapped LO phase offset and LO angular frequency offset.

The interplay between LO phase offset and LO frequency offset in equations (3.9)-(3.10) can be intuitively understood by considering the elements of $\mathbf{y}_k$. First, observe that the phase terms of $\mathbf{y}_k$ (the first two elements of $\mathbf{y}_k$) cannot exceed 2, whereas the frequency term of $\mathbf{y}_k$ (the third element of $\mathbf{y}_k$) can be arbitrarily large. In the early cycles of the EKF, the differences between the estimated and measured LO frequency offsets are often large. As a results, the phase terms of $\mathbf{y}_k$ will be negligible compared to the frequency term of $\mathbf{y}_k$, and the LO frequency offset will be the main driving element of the EKF. Once the estimated LO frequency offset approaches its measured values, the phase terms of $\mathbf{y}_k$ will no longer be negligible compared to the frequency term of $\mathbf{y}_k$. In this regime, the previously predicted LO frequency offset is used to determine the number of 2π-phase wraps that has occurred between the previous cycle and the current one. The current LO phase and frequency measurement are then used to adjust the previously predicted LO phase and frequency offset.

3.4 Conclusion

We have presented a novel signal processing architecture for digital synchronization of high-frequency RF signals. This architecture is based on the observation

that even at high frequencies on the order of 1 GHz, the relative frequency and phase offsets between a pair of oscillators are usually sufficiently small and slowly varying, that they can be estimated and corrected in software on standard CPUs. Specifically, we have: (a) proposed an algorithm based on a modified version of the classical Costas feedback loop [10] for frequency locking suitable for analog signaling schemes, (b) proposed an EKF based frequency locking scheme suitable for packet wireless networks, and (c) derived fundamental limits on the performance of one-shot blind estimation process.

CHAPTER 4
DISTRIBUTED BEAMFORMING: SDR IMPLEMENTATION AND RESULTS

In this chapter, we discuss the key ideas behind our recent implementation of distributed beamforming on software-defined radio platoform. First, we outline below the progress we have made so far with our implementation of distributed beamforming:

- **Beamforming implementation - version-I.** Open-loop carrier synchronization method (i.e., Master slave architecture) is used to achieve frequency lock among transmitters. Transmitters use costas loop to estimate and compensate for their frequency offsets w.r.t Master. This version uses analog signaling among the cooperating nodes. That is, transmitters send unmodulated tones to receiver and receiver reflects the feedback signal proportional to what it has received.
- **Beamforming implementation - version-II.** Round-trip carrier synchronization method is used to achieve frequency lock among transmitters. Transmitters use EKF to estimate and compensate for their frequency offsets w.r.t receiver. Transmitters still send unmodulated tones to the receiver but receiver now quantizes the feedback into 1-bit and sends BPSK/GMSK packet to the transmit nodes.
- **Beamforming implementation - version-III.** We add upon the version-II in that the transmit nodes now send data instead of just tones. To do this,

in addition of frequency and phase lock, timing synchronization among the transmit nodes is required. So, this is the first version which uses full digital signaling among the cooperating nodes.

In all three implementation versions, transmitters and receiver use 1-bit feedback algorithm to achieve phase coherence among the transmit signals at receiver.

All the nodes used in our experimental setup are based on the USRP RF and baseband boards [16] which is the most popular commercial SDR platform. For the first set of the experiments, we used the USRP-1 version of this platform; however, we have then ported our implementation to the most advanced version which are USRP N200 radios.

4.1 Background

Before we describe implementation details and discuss results, we present some background information and a brief survey of related work.

4.1.1 Cooperative transmission techniques

The large gains achievable through collaborative transmission schemes has been known to information theorists for many decades. Indeed the idea of cooperative beamforming is implicit in many early information theoretic works on multi-user channels [14]. The idea of distributed beamforming can also be further generalized to distributed MIMO [69], where nodes in a wireless network organize themselves into virtual arrays that use MIMO techniques such as spatial multiplexing and precoding to potentially achieve substantially better spatial reuse in addition to energy

efficiency. In fact, it has been shown recently [44] that wireless networks using distributed MIMO can effectively overcome the famous capacity scaling limits of wireless networks due to Gupta and Kumar [18]. This literature has, however, largely ignored the synchronization requirements for achieving these cooperation gains.

More recently the concept of *user cooperation diversity* where nearby users in a cellular system use cooperation to achieve decreased outage probability in the uplink was first suggested in [57] and further developed using space-time coding theory [29]. As noted earlier, cooperative diversity techniques have less stringent synchronization requirements [32] as compared to beamforming, but do not deliver the energy efficiency gains achievable with beamforming.

4.1.2 Experimental implementations of cooperative transmission techniques

Following up on the recent interest in cooperative communication, there have been several experimental implementations to study the practical feasibility of these ideas. This body of experimental work is summarized in a recent survey article [3], and has focused largely on cooperative diversity techniques. A recent experimental study of the amplify-and-forward relaying scheme [40] on Rice University's WARP platform [54] suggested that large gains are achievable even with a simple Alamouti space-time code. A DSP-based testbed was used for a comparative study of cooperative relaying schemes in [68]. A general testbed for systematically studying different MAC and PHY cooperative schemes was reported in [28]. Implementations of cooperative relaying have also been developed [2, 70] for software-defined radio platforms very

similar to the one used in our implementation.

Diversity schemes as pointed out earlier have substantially less stringent synchronization requirements than beamforming, which makes them easier to implement. However, there have also been several recent experimental studies of distributed beamforming [39, 58, 59]. All of the above implementations have been based on the 1-bit feedback algorithm.

Distributed beamforming is also at the heart of the Coordinated Multi-Point (CoMP) systems developed as part of the European EASY-C project [22, 25]; these make extensive use of various capabilities of cellular network infrastructure such as (a) uninterrupted availability of GPS signals, which are used to frequency-lock local oscillators and to supply symbol-level synchronization, (b) uplink channels with high bandwidths and low latencies to send detailed channel state feedback from the mobiles, and (c) a multi-gigabit backhaul network for Basestation coordination. In contrast, our work is aimed at the very different application setting of wireless sensor networks, where we cannot depend on the availability of such a sophisticated wired infrastructure.

4.2 Beam-steering using 1-bit feedback algorithm

The 1-bit feedback algorithm for beamforming was originally introduced in [37]. Under this algorithm, in every time-slot, each transmitter independently makes a random phase perturbation in its transmissions to the receiver; the receiver monitors the received signal strength (RSS), and broadcasts exactly 1 bit of feedback to the

transmitting nodes indicating whether the RSS in the preceding time-slot was greater than in previous time-slots. Using this 1 bit of feedback, the transmitters retain the favorable phase perturbations and discard the unfavorable ones.

Over time, it can be shown [38] that the transmitters converge to coherence *almost surely* under some mild conditions on the distribution of the phase perturbations. Furthermore the algorithm is extremely robust to noise, estimation errors, lost feedback signals and time-varying phases; these attractive properties make it possible to implement this algorithm on simple hardware, and indeed as noted earlier, distributed beamforming using variations of this basic algorithm has been demonstrated on multiple experimental prototypes [39, 58, 59] at various frequencies.

Nevertheless, this algorithm and its variants suffer from a number of shortcomings.

- *Slow convergence rate.* While the convergence rate of the 1-bit algorithm, with appropriately chosen parameters, has good scaling properties for large arrays (convergence time increasing no faster than linearly with number of transmitters [38]), in absolute terms, it requires a large number of time-slots.
- *Latency limitations.* The 1-bit algorithm neglects latency in the feedback channel; it assumes that the feedback signal is available instantaneously and simultaneously at all the transmitting nodes. In practice this may impose a high lower-bound on the time-slot duration which compounds the problem of slow convergence rate.
- *Poor performance with frequency offsets.* Non-zero frequency offsets between

transmitters manifest themselves as rapid time-variations in the phase. While variations of the 1-bit algorithm have been developed that can handle frequency offsets [58], these too require high feedback rates.

Recent work has shown that it is possible to overcome the above shortcomings of the 1-bit algorithm while retaining its attractive features by using richer feedback from the receiver [35]. In our experimental setup we have implemented the receiver feedback in a flexible way that allows for easy generalization to more advanced algorithms using multi-bit feedback.

The latency limitations mentioned above can be especially challenging for software-defined radio platform [56] that typically have multiple buffering stages in the data path, in addition to processing delays that depend on CPU loads and other uncontrollable factors. To get around this limitation, our current implementation uses a separate explicit mechanism for frequency locking the oscillators on the transmitters; this removes the frequency offsets and allows us to use the simple 1-bit algorithm for beamforming even with slow rates of feedback.

4.3 Beamforming implementation - version-I

The 1-bit feedback algorithm requires periodic feedback of 1 bit per time-slot from the receiver regarding the received signal strength (RSS) of the beamforming signal in the previous time-slot. In this version of our implementation, the receiver simply sends a continuous wave signal proportional to the amplitude of the received signal. This signal is broadcast wirelessly to all the beamforming nodes. This feedback

signal, of course, provides a lot more than 1 bit of feedback information, and indeed we designed our feedback channel in a flexible way to permit easy generalization of our implementation to more sophisticated algorithms [35] to take advantage of richer feedback information.

Each Slave node receives this feedback signal with a delay because of latencies in the software-defined radio system; we need to first estimate the round-trip (RT) latency between each Slave and the receiver in order to extract the 1-bit feedback required for the beamforming algorithm. In chapter 3, we described our implementation of the frequency-locking process in Section 3.2 which forms the first subprocess as explained in section 3.1. We now describe our implementation of the second subproblem i.e. the 1-bit beamforming algorithm for the case when feedback channel is analog. In short, the beamforming algorithm on each Slave node consists of an initialization procedure that measures the round-trip latency in the feedback channel, followed by the actual implementation of the beamforming algorithm.

The latency measurement algorithm is based on the following simple idea. Initially when none of the beamforming nodes are transmitting, the signal level at the receiver consists of just background noise which is quite small and therefore the amplitude of the feedback signal is also correspondingly small. Then when one of the Slaves starts transmitting, it can estimate its RT latency simply by counting the number of samples it takes before it sees an increase in the amplitude of the feedback signal from the receiver. This, of course, requires that each Slave node be calibrated individually. In our setup, we do this by using special flags in the software that can

be switched on and off in real-time to start and stop transmitting from each Slave node.

The pseudo-code for the initialization process and the beamforming algorithm are given in Algorithms 4.1 and 4.2 respectively. Key parameter values along with corresponding variable names referred to in the pseudo-code are in Table 4.1.

Parameter	**Variable name**	**Value**
Round-trip latency	r_t_latency	≈30 ms
Averaging start time	avg_st_time	(r_t_latency+1)ms
Averaging end time	avg_end_time	(r_t_latency+21)ms
Beamforming time-slot end time	bf_t_slot_end	(r_t_latency+22)ms
Low-pass filter bandwidth	-	30kHz
Low-pass filter transition width	-	20kHz
Frequency correction factor of Costas loop	-	892/964
VCO sensitivity of Costas loop	-	100k rad/s/V
Baseband sampling rate	samp_rate	2 Msps
FPGA Decimation	-	32
FPGA Interpolation	-	64
Random phase perturbation distribution	-	uniform
Random phase perturbation angle	rand_pert	±15 degrees
Past RSS window size	past_rss_win	4

Table 4.1: Key parameters: Beamforming implementation - version-I.

Specifically, each slave node starts computing the sample average (to obtain an estimate of current RSS) 1mS after its estimate of round-trip latency and it does the averaging for 20mS; this is indicated by averaging start time and averaging end time paramters in the table. Then, there is 1mS of guard time and subsequently next time-slot/iteration of algorithm starts.

Algorithm 4.1 Round-trip latency measurement: Beamforming implementation - version-I.

Initialization:

$initial\_flag \leftarrow true$

$samp\_count \leftarrow 0$

while $initial\_flag = true$ **do**

Average every 1000 samples to get an $RSS\,estimate$

Compare $RSS\,estimate$ with a pre-defined $threshold$

if $RSS\,estimate \geq threshold$ **then**

$initial\_flag \leftarrow false$

//Round-trip latency in number of samples:

$r\_t\_latency \leftarrow samp\_count$

$avg\_st\_time \leftarrow r\_t\_latency + (1mS \times samp\_rate)$

$avg\_end\_time \leftarrow r\_t\_latency + (21mS \times samp\_rate)$

$bf\_t\_slot\_end \leftarrow r\_t\_latency + (22mS \times samp\_rate)$

//Round-trip latency in milli-seconds:

$r\_t\_latency \leftarrow (samp\_count/samp\_rate) \times 1000$

end if

end while

Algorithm 4.2 1-bit feedback algorithm: Beamforming implementation - version-I

Initialization:

$samp\_count, past\_rss\_win, cum\_phase \leftarrow 0$

while $initial\_flag = false$ **do**

if $avg\_st\_time \leq samp\_count < avg\_end\_time$ **then**

Average the received signal samples to obtain $current\_rss$, the estimate of RSS of current time-slot

else if $samp\_count = avg\_end\_time$ **then**

Compare $current\_rss$ with $past\_rss\_win$

if $current\_rss > past\_rss\_win$ **then**

$feedback\_bit \leftarrow true$

else

$feedback\_bit \leftarrow false$

end if

From $\pm rand\_pert$, generate random phase perturbation as c_rand_pert

$cum\_phase \leftarrow cum\_phase + c\_rand\_pert$

if $feedback\_bit = false$ **then**

$cum\_phase \leftarrow cum\_phase - p\_rand\_pert$

end if

Shift the FIFO $past\_rss\_win$ by 1 to save $current\_rss$ in it

Save c_rand_pert as p_rand_pert

else if $samp\_count = bf\_t\_slot\_end$ **then**

$samp\_count \leftarrow 0$

end if

end while

4.3.1 Frequency division multiplexing scheme

One important thing to note about this setup is that there are three different RF signals being transmitted by various nodes in the network simultaneously: the reference tone from the Master node to the Slaves, the beamforming signal from the Slaves towards the Receiver, and the feedback signal from the Receiver to the Slaves. Specifically, we note that the Slave nodes receive both a reference tone from the Master node and a feedback signal from the Receiver.

Thus we need to design a suitable multiplexing scheme to make sure these signals do not interfere with each other, and can be extracted using relatively simple filtering operations implemented in software. In addition, we also need to ensure that *duplexing constraints* are satisfied i.e. a nodes' transmissions should not fall within the bandwidth of the same node's receiver, so there is sufficient amount of isolation between the transmit and receive hardware.

The frequency multiplexing scheme used in our experimental setup is illustrated in Fig. 4.1. The choice of the specific frequencies in this scheme reflects a balancing act between two conflicting objectives: on the one hand, we want to minimize the overall bandwidth of the signal received by the Slave node, so that the signal can be digitized with a relatively low sampling rate and therefore a small processing burden for the signal processing software. On the other hand, if we make the frequency separation between the reference signal from the master and the feedback signal from the receiver too small, then we will need sharp frequency-selective filters at the Slave nodes to separate the two signals, and this in turn increases the

processing burden for the Slave nodes.

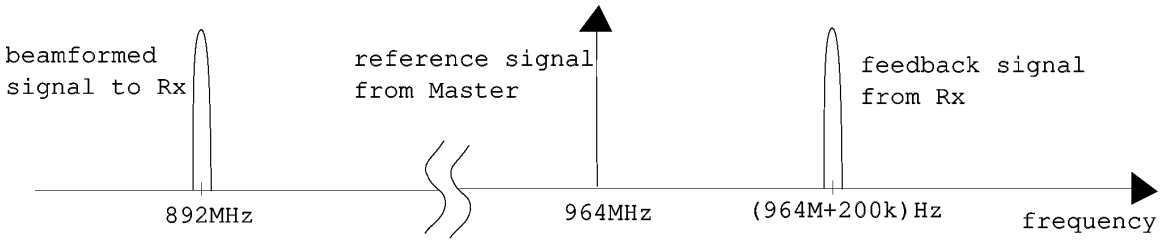

Figure 4.1: FDM scheme for beamforming implementation - version-I.

4.3.2 Results

We now show some experimental results from our implementation. Fig. 4.3 shows a photograph of the receiver node in our experimental setup which is where the measurements reported in this section were recorded. In addition to the "Flex 900" RF daughterboard that the receiver node uses for receiving the beamforming signal and for transmitting the feedback signal, we also connected an additional "Basic Tx" daughterboard to the receiver node to enable us to view the received signal strength at the receiver on an external oscilloscope. This setup is illustrated in Fig. 4.15.

Figs. 4.4, 4.5 show screenshots from the oscilloscope of two runs of the beamforming experiment using two and three Slaves respectively. Specifically, Figs 4.4 and 4.5 show the amplitude of the received signal from the beamforming Slaves, with each Slave node transmitting individually at first, and then transmitting together while implementing the beamforming algorithm. Fig. 4.4 also has an interval (T6)

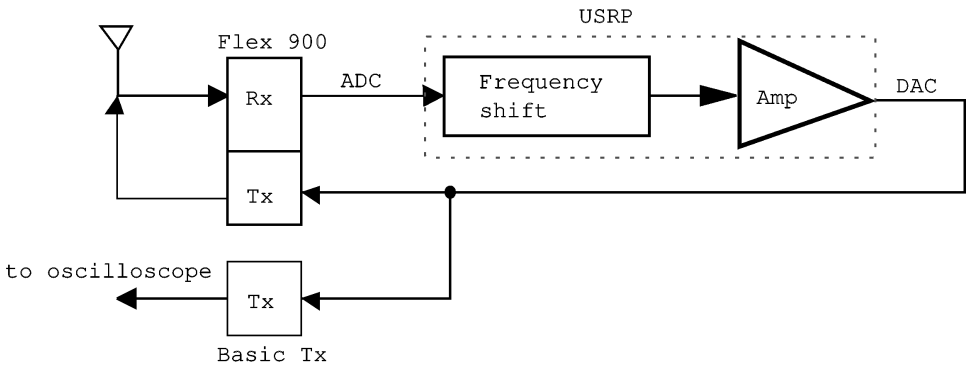

Figure 4.2: Measurement setup for beamforming experiment.

where the Slaves are transmitting together incoherently (i.e. without running the beamforming algorithm).

It is also possible to dispense with the external oscilloscope completely and simply save samples of the received signal at the receiver node for offline processing and plotting; a typical result is shown in Fig. 4.6 which represents a run of the beamforming experiment with the same sequence of steps as Fig. 4.5.

The coherent gains from beamforming are apparent from the above plots. In other words, the amplitude of the received signal is seen to be close to the sum of their individual amplitudes. It can also be seen from Fig. 4.4 that the beamforming gains quickly deteriorate when the two Slaves are transmitting together but incoherently i.e. with the beamforming algorithm disabled.

While the transmitted signal in Figs. 4.4, 4.5, 4.6 is just an unmodulated sinusoidal tone, it is straightforward to adapt this setup to send a data signal. We illustrate this in Fig. 4.7 where the beamforming transmitters use a simple ON/OFF

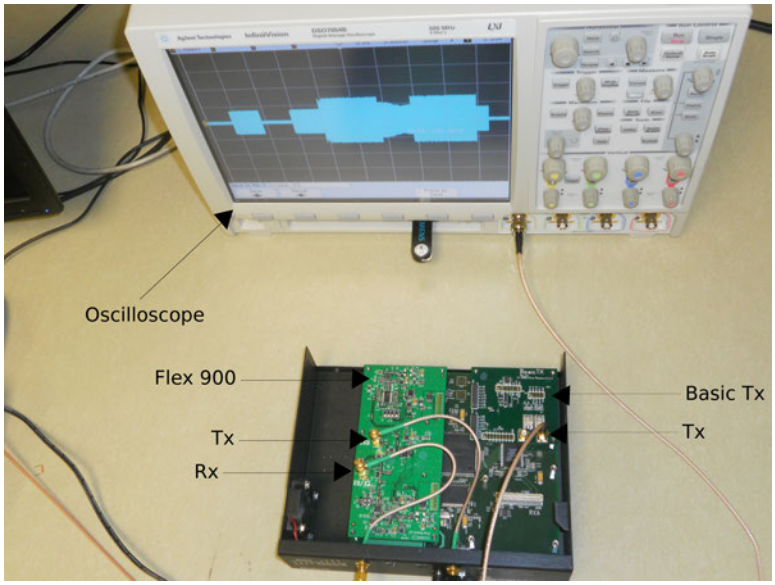

Figure 4.3: Photograph of measurement setup.

keying scheme to transmit a sequence of bits to the receiver. Specifically, Fig. 4.7 shows the envelope of two ON/OFF keyed received signals in two experimental runs: Experiment 1 with two beamforming transmitters and Experiment 2 with a single transmitter. We calibrated the transmitted power in Experiment 2 such that the total transmitted power is the same in both experiments; specifically, in Experiment 1, the two beamforming nodes transmit with power P each, and the single transmitter in Experiment 2 transmits with power 2P. The stronger received signal in Experiment 1 shows the beamforming gain.

Finally, the plot in Fig. 4.8 shows the "transient" of the beamforming process; specifically it shows the amplitude of the received signal, with one Slave transmitting individually at first, then the second Slave being turned on with the beamforming

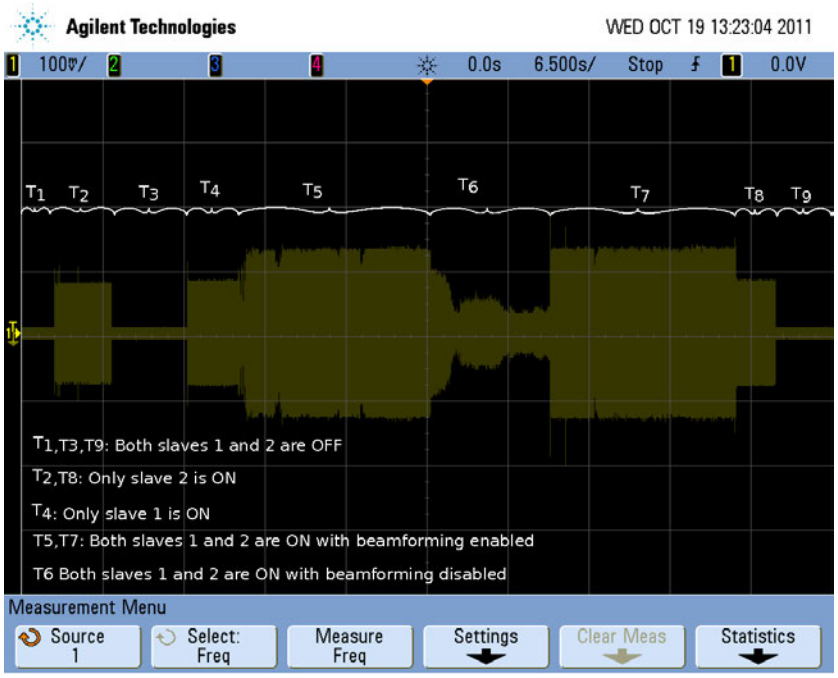

Figure 4.4: Received signal at the receiver with two transmitters.

algorithm activated on both nodes. It is seen that the convergence time of the beamforming algorithm is on the order of several hundred milliseconds, which represents around 15 timeslots.

4.4 Beamforming implementation - version-II

This work was lead by our collaborators at UCSB. In this version of beamforming implementation, there are two important advances compared to version-I.

1. The transmit nodes use blind estimation algorithm along with EKF to track and compensate for their frequency offsets w.r.t receiver. So, the network employs round-trip method for carrier synchronization which eliminates the need for a

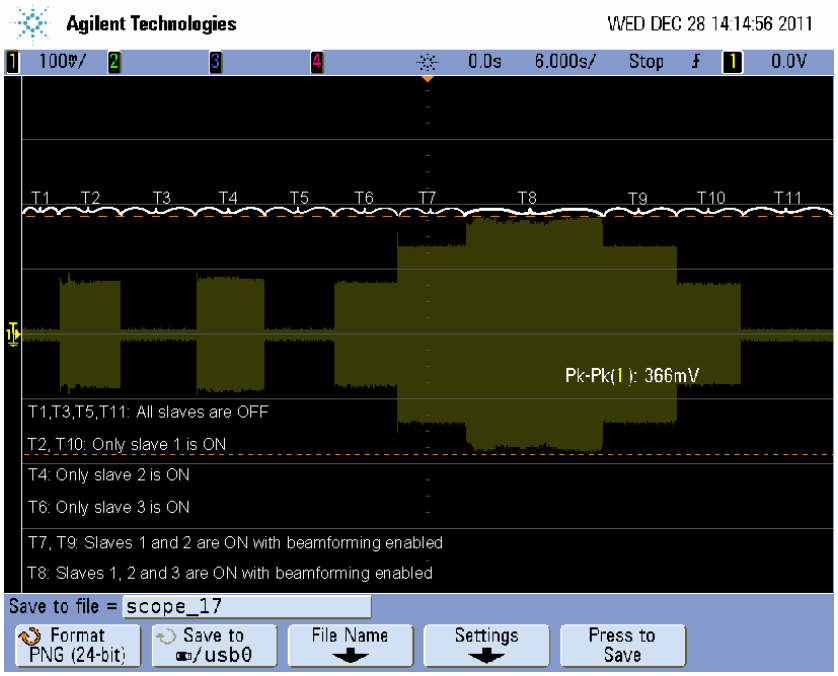

Figure 4.5: Received signal at the receiver with three transmitters.

dedicated master transmitter.

2. The feedback link is digital.

Essentially, the receiver sends periodic feedback to the transmitters. Specifically, during every time-slot, receiver broadcasts a GMSK packet which contains in its payload the 1-bit generated by 1-bit feedback algorithm at the receiver. The transmitters exploit the GMSK baseband waveform itself to estimate blindly their frequency offsets w.r.t receiver (see Fig. 4.9). The slaves then use the payload of the received feedback packet to guide their phases to steer the beam towards the receiver. More detailed description of the whole experimental setup can be found in [48], [47], [49].

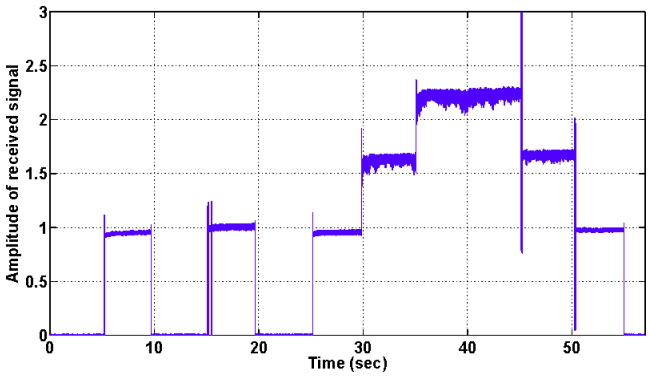

Figure 4.6: Received signal amplitude at the receiver with three transmitters.

4.4.1 Results

Fig. 4.10 has been borrowed from [49] which shows the beamforming gain when initially 2 and then 3 transmitters are switched on and they beamform towards a receiver. Again, more detailed results on beamforming gain analysis, EKF convergence etc. can be found in [48], [47], [49].

4.5 Beamforming implementation - version-III

This version of beamforming is perhaps the most interesting among all the 3 versions. It has the following distinguishing features:

1. In addition to feedback link, forward link is also digital. This means the transmitters now send data (not just the sinusoids) to the receiver.
2. If the transmitters want to send common message to the receiver, they must be time synchronized so as to make sure that there is no ISI due to timing mismatch between them. This method employs one such distributed timing

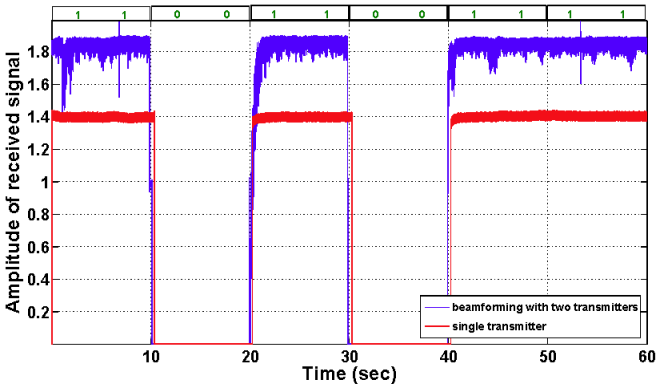

Figure 4.7: Data transmission using ON-OFF keying.

synchronized method.

As can bee seen in Fig. 4.11, the whole system now operates in packet-mode. That is the transmissions on forward and feedback channel occur in orthogonal time-slots; therefore, one can employ TDM instead of FDM mode. A few remarks about Fig. 4.11 are worth mentioning. The red-colored rectangles represent feedback packets while the data packets are represented by blue-colored rectangles. Similarly, yellow-colored chunk inside the red rectangle and the pink-colored chunk inside the blue rectangle represent the 1-bit of feeback and common message respectively.

Fig. 4.11 also explains the way transmitters align their transmission times. Specifically, each transmitter computes the time of packet arrival (TOPA) for every feedvack packet they receive from receiver. Transmitters compute TOPA by correlating the incoming feedback packet against a known header using matched filters. Each transmitter then adds some delay (which is same for all transmitters) to time its

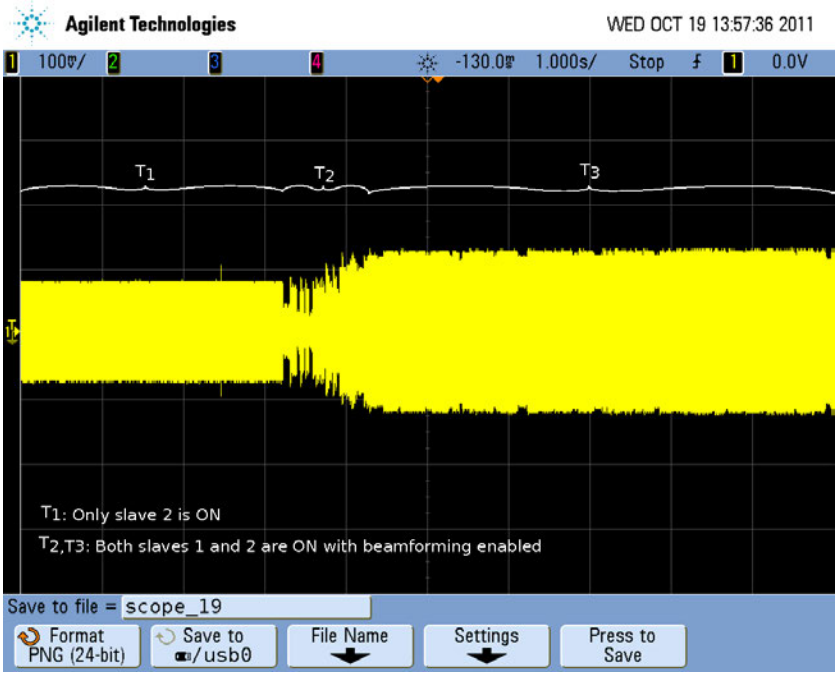

Figure 4.8: Transient of the beamforming process.

transmission of common message in near future time (which is 20mS in our implementation). The standing assumption is that TOPA is the same at all transmitters which is justified by the two facts: i) the propagation delay is negligible for all the paths between receiver and each transmitter ii) transmitter mark the TOPA using FPGA time which is the time when feedback packet hits the transmitters and is unaffected by the different latencies which arise due to ethernet connection etc.

In summary, the transmit nodes now have three disjoint sub-processes running in parallel, i.e., frequency synchronization, phase synchronization and timing synchronization (see Fig. 4.12). The remarkable fact about our implementation is that transmitters achieve all 3 kinds of synchronization/alignments in purely distributed

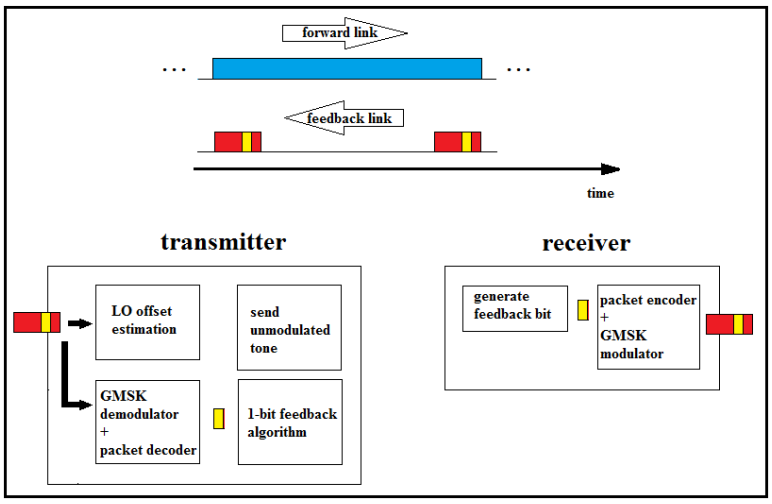

Figure 4.9: block diagram of the system - feedback link is digital.

fashion, i.e., without talking among themselves, rather with distant receiver only. Moreover, the overhead for these processes is really small. Same feedback packet is used for all three kinds of synchronization. That is, the complex baseband waveform of received feedback message provides a mean for blind frequency offset estimation; the payload in the packet contains the feedback bit for 1-bit feedback algorithm; finally, correlating packet with its header part gives us TOPA which is used to align transmission times of transmitters.

We conducted some experiments to quantify the accuracy of TOPA estimation method. It turns out that the estimation method is accurate on the order of tens of micro-seconds. Therefore, transmitters can enjoy a data rate on the order of hundreds of Kbps. Fig. 4.13 quantifies the timing spread/mismatch (in number of samples, for sampling rate of 200Ksps) when two transmitters want to align their timings using

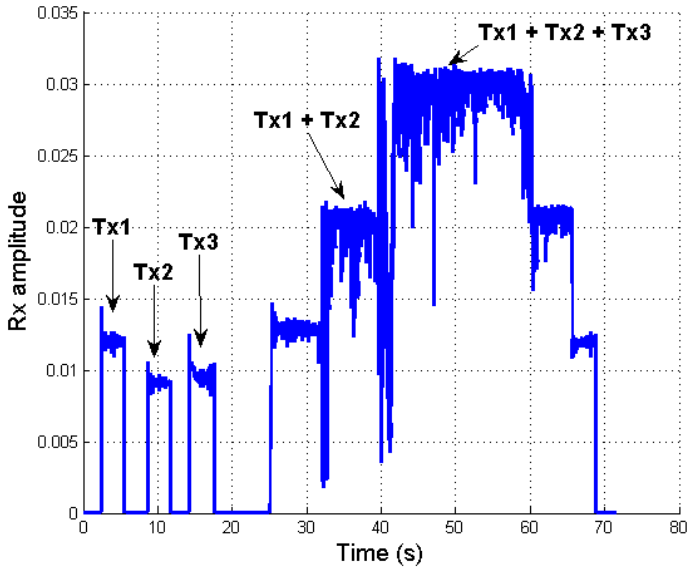

Figure 4.10: Gains in RSS due to beamforming version-II ([49]).

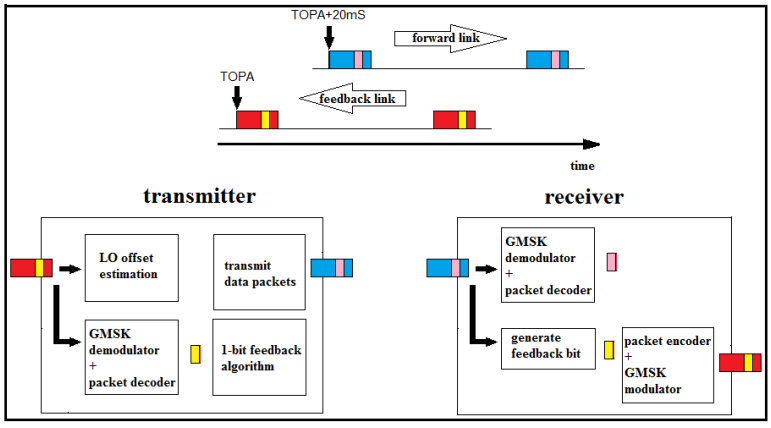

Figure 4.11: block diagram of the system - feedback link and forward link are both digital.

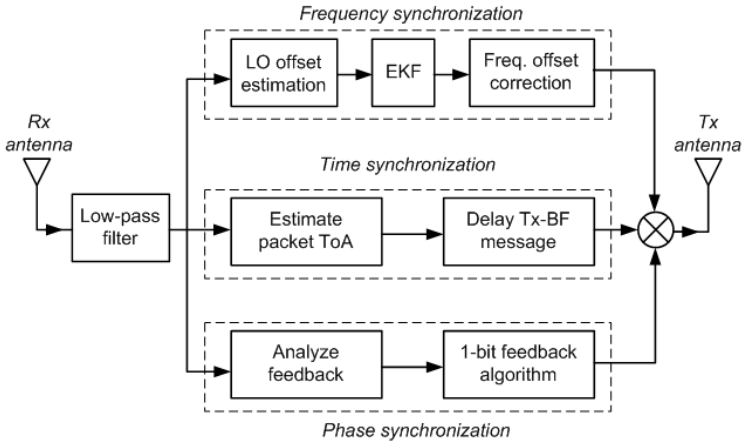

Figure 4.12: block diagram of transmit node.

aforementioned TOPA method.

4.5.1 Results

Figures 4.14 and 4.15 have been obtained using a phase perturbation of size 20 degrees on Tx.1 and a phase perturbation of size 30 degrees on Tx.2 (the beamforming didn't occur for small sizes of perturbation angle because of the residual frequency drift). We recorded 5 runs of the beamforming experiment while sending 500 data packets during each of the 5 experiment runs. The number of packets successfully received during the 5 experiments were 442,460,429,446,462 respectively.

In Fig. 4.14, the RSS is not steady during the initial part which is due to the fact that due to software limitations, some time feedback packets are not sent to transmitters; therefore, the frequency-locking assumption no more remains valid. However, as can be seen in Fig. 4.14, as soon as new measurements are available to

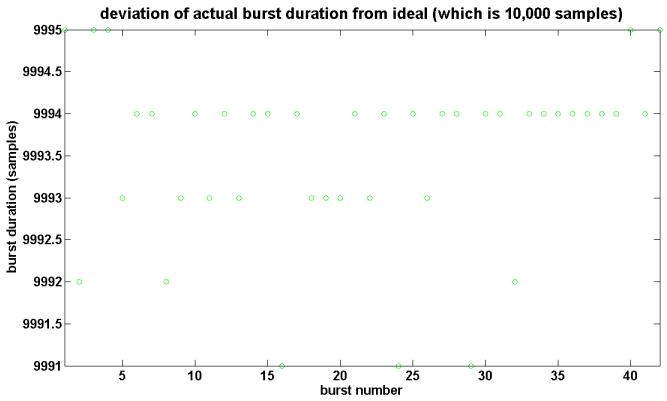

Figure 4.13: illustration of timing synchronization among the transmit nodes.

transmit nodes, the beamforming process quickly restores itself.

4.6 Conclusions

We described our implementation of distributed beamforming on an open-source software-defined radio platform. To the best of our knowledge, this is the first ever all-wireless implementation of distributed beamforming; previous experimental work in [39, 58, 59] all made use of reliable wired, secondary communication channels for channel feedback and/or to distribute a reference clock signal. This implementation is based on a novel signal processing architecture for the synchronization of high frequency RF signals entirely in software. Our results show that the synchronization requirements for beamforming can be satisfied with modest overheads on inexpensive commodity platforms without any hardware modifications and without any wired infrastructure. This opens up many interesting possibilities for future work in further developing open-source building blocks for bringing the large potential gains from

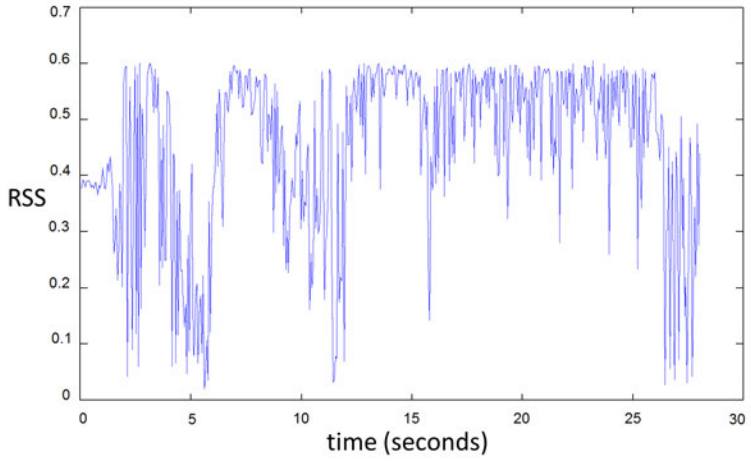

Figure 4.14: Gains in RSS due to beamforming - version-III.

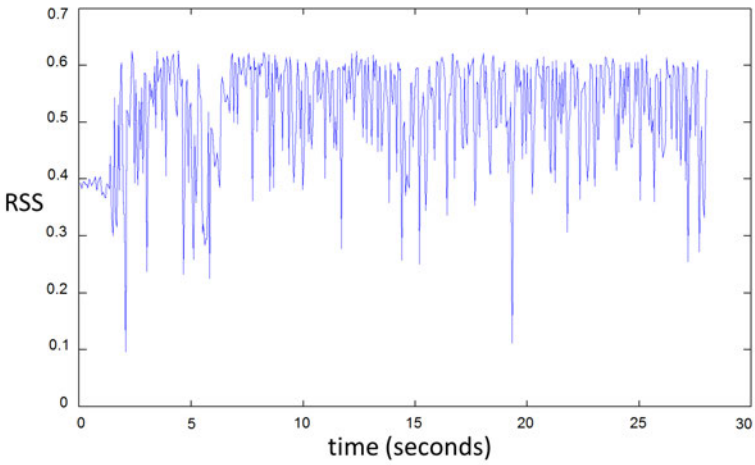

Figure 4.15: Gains in RSS due to beamforming - version-III.

virtual antenna arrays to real-world wireless networks. In addition, this poses a challenge of designing effective networking protocols to take advantage of cooperative communication schemes such as beamforming.

CHAPTER 5
DISTRIBUTED NULLFORMING: ALGORITHMS AND CONVERGENCE ANALYSIS

5.1 Introduction

In this chapter, we consider the problem of distributed nullforming where a set of transmitters in a wireless network cooperatively transmit a common message signal in such a way that their individual transmissions cancel each other at a designated receiver. In effect the transmitters form a *virtual antenna array* and shape the array's antenna pattern to create a null at the desired location. The technique of distributed nullforming has many potential applications including interference avoidance for increased spatial spectrum reuse [43], cognitive radio [67], physical-layer security [12] and so on.

Distributed nullforming requires precise control of the amplitude and phase of the radio-frequency signal transmitted by each cooperating transmitter to ensure that they cancel each other. This is an extremely challenging problem because each transmitter usually obtains its RF signal from a separate local oscillator (LO), and signals obtained from different LOs invariably have Brownian motion driven phase drifts due to manufacturing tolerances and temperature variations. The nullforming algorithm must estimate, track and compensate for the effect of these drifts.

While the idea of cooperative communication has been studied for decades [11], the early work in this area neglected the RF synchronization issues that are crucial for the practical implementation of these ideas. Recently, however, there has

been a significant amount of research activity on distributed transmit beamforming [34], [36], [58], including implementation on commodity hardware [50, 48].

While the synchronization techniques developed for distributed beamforming can be adapted for nullforming, there are two important differences that make nullforming significantly more challenging: (a) While beamforming gains are highly robust and insensitive to small phase errors (upto about 30 degrees [34]), nullforming is substantially more sensitive [4] to even modest errors. (b) One implication of this sensitivity to small phase errors is that the simple 1-bit feedback algorithm [37] that has proved to be effective for beamforming does not work for nullforming. However, we show in this paper that a gradient descent algorithm using multi-bit feedback similar to [35] works very well for nullforming. (c) For beamforming, each transmitter only needs the knowledge of the phase of its own transmitted signal at the receiver. In contrast for nullforming, the amplitude and phase of the transmitted signal at each node cannot be chosen independently of the amplitudes and phases of other nodes [4]. Nullforming essentially depends on a node's transmitted signal cancelling the signals from all other transmitters. Therefore state-of-the-art distributed nullforming algorithms, [6] and [4] assume that each transmitter knows *every transmitter's* complex channel gain to the receiver. This requirement poses a severe challenge for scalability.

In contrast to previous work on distributed nullforming [6, 4], in this work we assume that each transmitter knows *only its own channel gain* to the null location. Using this in Section 5.2 we formulate our gradient descent based algorithm, in which each node adjusts its transmitted phase knowing only its channel gain, and a common

feedback signal from the receiver at which the null is desired. This feedback signal is simply the complex baseband signal received by the receiver. Section 5.3 presents an analysis of the stability and convergence properties of the algorithm under simplifying assumptions. Section 5.4 provides simulations, that include the effect of channel phase offsets and oscillator drift. Section 5.5 concludes.

5.2 Scalable algorithm for nullforming

We now describe a scalable gradient descent algorithm for distributed nullforming in a node. As noted in the introduction, we assume that at the beginning of a nullforming epoch, each transmitter has access to its own complex channel gain to the receiver, using which it equalizes its channel to the receiver. This is in sharp contrast to [6] and [4] where each transmitter knows the Channel State Information (CSI) for every transmitter. We assume there are N transmitter nodes that have been synchronized in frequency, using the techniques of [6], [4] and [35].

Assume at time slot k, the i-th node transmits the baseband signal $e^{j\theta_i[k]}$. The total baseband signal at the receiver is thus:

$$s[k] = R[k] + jI[k], \tag{5.1}$$

where

$$R[k] = \sum_{i=1}^{N} r_i \cos\left(\theta_i[k] + \phi_i[k]\right), \tag{5.2}$$

$$I[k] = \sum_{i=1}^{N} r_i \sin\left(\theta_i[k] + \phi_i[k]\right), \tag{5.3}$$

r_i is the equalized channel gain from the i-th transmitter and $\phi_i[k]$ is a small uncompensated channel phase from the i-th transmitter. The receiver feeds back at each

time slot the signal $s[k]$. Consequently, at each time slot the i-th transmitter has access to $R[k]$, $I[k]$, r_i and $\theta_i[k]$; $\phi_i[k]$ is not available to any one. Define,

$$\theta[k] = [\theta_1[k], \cdots, \theta_N[k]]^\top. \tag{5.4}$$

The total received power in the k-th time slot is:

$$J(\theta[k]) = I^2[k] + R^2[k]. \tag{5.5}$$

Since the r_i are equalized gains, each receiver can always choose its r_i to equal 1, ensuring the existence of a choice of θ_i that achieve a null. Indeed throughout we make the following standing assumption:

Assumption 5.2.1. The equalized gains $r_i = 1$ for all $i \in \{1, \cdots, N\}$.

For a suitably small $\mu > 0$, in our algorithm the i-th transmitter updates its phase according to:

$$\theta_i[k+1] = \theta_i[k] + \mu \left(\sin\left(\theta_i[k]\right) R[k] - \cos\left(\theta_i[k]\right) I[k] \right). \tag{5.6}$$

Few features are of note. The algorithm is totally distributed, as each node only needs the common feedback signal $s[k]$ and r_i and $\theta_i[k]$, to implement it. This contrasts with [6], [4] where much more information is needed. Second, suppose in vector form the algorithm were expressed as:

$$\theta[k+1] = \theta[k] - f[k]. \tag{5.7}$$

Then, when the phase offsets $\phi_i[k]$ are all zero, the $f[k]$ corresponding to (5.6) is simply:

$$f[k] = \mu \left. \frac{\partial J(\theta)}{\partial \theta} \right|_{\theta=\theta[k]}. \tag{5.8}$$

In other words the algorithm attempts the gradient descent minimization of the received power. Finally, the fact that the algorithm works from a common feedback signal supplied by the receiver, makes it *totally scalable* as the feedback overhead does not grow with the size of the transmitter array.

5.3 Stability

Our stability analysis will be conducted under the idealized assumption of no noise and zero $\phi_i[k]$. The underlying philosophy is driven by total stability theory, [20], that states in essence that should the algorithm uniformly converge to desired stationary points in the idealized (zero noise, zero ϕ_i) case, uniformity being with respect to the initial time, then it will exhibit robustness to noise and small ϕ_i. Indeed we will demonstrate the *practical* uniform convergence of (5.6) under the following assumption:

Assumption 5.3.1. In (5.2) and (5.3) for all $i \in \{1, \cdots, N\}$ and all k, $\phi_i[k] = 0$.

Let us clarify what we mean by *practical* uniform convergence. As will be evident from the sequel, under Assumption 5.3.1 the algorithm in (5.6) has entire manifolds of stationary points at least to one of which the algorithm converges uniformly. Some stationary correspond to nulls. The rest, which we dub as being *spurious*, do not. We will show that the latter are locally unstable. Thus they are rarely attained, and even if attained not practically maintained as the slightest noise would drive the phase trajectories away from them. Thus, by showing the local stability of the stationary points corresponding to nulls, we would have demonstrated the practical

uniform converence of the algorithm to a null.

5.3.1 Characterizing nulls

We relax Assumption 5.3.1 to permit non-zero *but constant* ϕ_i. Under these conditions from (5.6) we obtain that the stationary points fall into the following categories.

- [A] $R[k] = I[k] = 0$.
- [B] If $R[k] \neq 0$, then for all i,

$$\tan \theta_i[k] = \frac{I[k]}{R[k]}.$$

- [C] If $I[k] \neq 0$, then for all i,

$$\cot \theta_i[k] = \frac{R[k]}{I[k]}.$$

Clearly [A] corresponds to stationary points reflecting nulls. Both [B] and [C] reflect the condition that for all i, l, there holds:

$$\tan \theta_i = \tan \theta_l. \tag{5.9}$$

Some of these may still correspond to nulls. The rest are spurious.

5.3.2 Assured uniform convergence to a stationary point

We will now invoke Assumption 5.3.1. We have the following Theorem.

Theorem 5.1. *Under Assumption 5.3.1, (5.2), (5.3), (5.6) and (5.5), there exists a* $\mu^* > 0$, *such that for all* $0 < \mu < \mu^*$, $\theta[k]$ *converges uniformly to one of the stationary points in (A-C) above.*

Proof. Observe, under Assumption 5.3.1, (5.7) and (5.8) hold. Consequently, the only stationary points of (5.6) obey

$$\frac{\partial J(\theta)}{\partial \theta} = 0. \tag{5.10}$$

These clearly correspond to those enumerated in (A-C). Thus it suffices to show that the following occurs uniformly:

$$\lim_{k\to\infty} \frac{\partial J(\theta)}{\partial \theta}\bigg|_{\theta=\theta[k]} = 0. \tag{5.11}$$

To prove this we first observe that there exists a number M that bounds the derivatives of all orders of $J(\cdot)$. Since (5.7) and (5.8) hold, we have that:

$$\begin{aligned} J(\theta[k+1]) &= J\left(\theta[k] - \mu \frac{\partial J(\theta)}{\partial \theta}\bigg|_{\theta=\theta[k]}\right) \\ &\leq J(\theta[k]) \\ &- \mu \left\| \frac{\partial J(\theta)}{\partial \theta}\bigg|_{\theta=\theta[k]} \right\|^2 \left(1 - \mu M \sum_{n=1}^{\infty} \frac{(\mu M)^{n-1}}{(n+1)!}\right) \\ &= J(\theta[k]) - \mu \left\| \frac{\partial J(\theta)}{\partial \theta}\bigg|_{\theta=\theta[k]} \right\|^2 \\ &\quad \left(1 - \frac{e^{\mu M} - 1 - \mu M}{\mu M}\right) \end{aligned}$$

Observe that

$$\lim_{x\to 0} \frac{e^x - 1 - x}{x} = 0.$$

Thus for every $\epsilon > 0$, there exists a $\mu^* > 0$, such that for all $0 < \mu < \mu^*$, there holds:

$$J(\theta[k+1]) = J(\theta[k]) - \mu(1-\epsilon) \left\| \frac{\partial J(\theta)}{\partial \theta}\bigg|_{\theta=\theta[k]} \right\|^2. \tag{5.12}$$

Then the result follows from standard considerations and the nonnegativity of $J(\cdot)$.

■

5.3.3 All spurious stationary points are locally unstable

Standard theory shows that the local instability of the algorithm in (5.6) is assured if the algorithm linearized around that stationary point has poles outside the unit circle. Under 5.3.1 this in turn is assured if the Hessian of $J(\cdot)$ evaluated at such a stationary point has a negative eigenvalue. The theorem below shows that this is indeed the case.

Theorem 5.2. *Consider (5.6), under (5.2) and (5.3) with assumptions 5.2.1 and 5.3.1 in force. Consider any θ at which $J(\theta) \neq 0$, but (5.10) holds. Then the Hessian, $H(\theta)$ whose i,l-th element is*

$$\frac{\partial^2 J(\theta)}{\partial\theta_i \partial\theta_l}$$

cannot be positive semidefinite.

Proof. First observe that under Assumptions 5.2.1 and 5.3.1 there holds:

$$\begin{aligned} J(\theta) &= \left(\sum_{i=1}^{N} \cos\theta_i\right)^2 + \left(\sum_{i=1}^{N} \sin\theta_i\right)^2 \\ &= N + 2\sum_{i=1}^{N}\sum_{\substack{l=1 \\ l\neq i}}^{N} (\cos\theta_i \cos\theta_l + \sin\theta_i \sin\theta_l) \\ &= N + 2\sum_{i=1}^{N}\sum_{\substack{l=1 \\ l\neq i}}^{N} \cos(\theta_i - \theta_l). \end{aligned} \tag{5.13}$$

Thus the i-th element of the gradient is given by:

$$\frac{\partial J(\theta)}{\partial\theta_i} = 2\sum_{\substack{l=1 \\ l\neq i}}^{N} \sin(\theta_l - \theta_i) \tag{5.14}$$

Thus, the (i,l)-th element of such an Hessian obeys:

$$[H(\theta)]_{il} = \begin{cases} -2\sum_{\substack{l=1 \\ l\neq i}}^{N} \cos(\theta_i - \theta_l) & i = l \\ 2\cos(\theta_i - \theta_l) & i \neq l \end{cases}$$

As $J(\theta) \neq 0$, for all $i, l \in \{1, \cdots, N\}$ (5.9) holds. Thus for all $i, l \in \{1, \cdots, N\}$

$$\cos(\theta_i - \theta_l) = \pm 1$$

Thus all off-diagonal elements of $H(\theta)$ are non-zero. To prove the result it thus suffices to show that at least one diagonal elements is non-positive. Without loss of generality, assume $\theta \in [0, 2\pi)^N$. As only the differences of θ_i appear in the expression for $H(\theta)$, in view of (5.9), again without loss of generailty, one can partition $\{1, \cdots, N\}$ into two sets $\mathcal{I}_0$ and $\mathcal{I}_\pi$ such that

$$\theta_i = \begin{cases} 0 & i \in \mathcal{I}_0 \\ \pi & i \in \mathcal{I}_\pi \end{cases}$$

Also observe that

$$\cos(\theta_i - \theta_l) = \begin{cases} 1 & \{i, l\} \subset \mathcal{I}_0 \text{ or } \{i, l\} \subset \mathcal{I}_\pi \\ -1 & \text{else} \end{cases}$$

We need to show that for at least one $i \in \{1, \cdots, N\}$ there holds:

$$\sum_{\substack{l=1 \\ l \neq i}}^{N} \cos(\theta_i - \theta_l) \geq 0. \tag{5.15}$$

We consider the following three cases:

Case I: $|\mathcal{I}_0| - 1 = |\mathcal{I}_\pi|$**.** Consider $i \in \mathcal{I}_0$. Then the number of summands in (5.15) that are 1, equals the number that are -1, and the sum equals zero.

Case II: $|\mathcal{I}_0| + 1 = |\mathcal{I}_\pi|$**.** Consider $i \in \mathcal{I}_\pi$. Then the number of summands in (5.15) that are 1, equals the number that are -1, and the sum equals zero.

Case III: Neither Case I nor Case II holds. Then there is at least one $i \in \{1, \cdots, n\}$ for which the number of summands in (5.15) that are 1, is greater than the number that are -1, and the sum is positive. ■

Thus indeed all spurious equlibria of (5.6) are locally unstable.

5.3.4 Practical uniform convergence to a null

In view of Section 5.3.1, Theorem 5.1 and Section 5.3.3 practical uniform convergence is guaranteed by showing that all stationary points corresponding to a true null are locally stable. To this end let us first examine the Hessian at these points corresponding to nulls. Assumption 5.2.1 guarantees the existence of stationary points. Under Assumption 5.3.1 at a stationary point corresponding to a null, i.e. when $R = I = 0$, there holds:

$$[H(\theta)]_{il} = \begin{cases} 2 & i = l \\ 2\cos(\theta_i - \theta_l) & i \neq l \end{cases}$$

It is readily seen that at such a stationary point, with $c = \begin{bmatrix}\cos\theta_1 & \cdots & \cos\theta_N\end{bmatrix}^\top$ and $s = \begin{bmatrix}\sin\theta_1 & \cdots & \sin\theta_N\end{bmatrix}^T$ the Hessian is $2cc^\top + 2ss^\top$. Thus the Hessian evaluated at a null is postive semidefinite, but with rank at most 2. There are several zero eigenvalues of the Hessian. Thus, the examination of the Hessian on its own is inconclusive. One approach to resolving this issue to use, as was done in [62], center manifold theory.

Instead, we adopt here a different approach. Specifically, we first show in the lemma below the absence of spurious equlibria in a neighborhood of the null manifold.

Lemma 5.3.1. Suppose assumptions 5.2.1 and 5.3.1 hold. Suppose $J(\theta) < 1$. Then

$$\frac{\partial J(\theta)}{\partial \theta} = 0 \Leftrightarrow J(\theta) = 0.$$

Proof. At a spurious stationary point, for all $i, l \in \{1, \cdots, n\}$, $\cos(\theta_i - \theta_l) = \pm 1$. Thus from (5.13), at a spurious stationary point $J(\theta)$ is a nonnegative integer. The result follows. ■

We thus have our main result.

Theorem 5.3. *Consider (5.6), under (5.2) and (5.3) with assumptions 5.2.1 and 5.3.1 in force. Then there exists a neighborhood of the manifold defined by $J(\theta) = 0$, such that for all $\theta[0]$ in this neighborhood, there exists a μ^* such that for all $0 < \mu < \mu^*$:*

$$\lim_{k\to\infty} J(\theta[k]) = 0.$$

Proof. Consider for any $0 < a < 1$, and the neighborhood

$$\mathcal{N}(a) = \left\{ \theta \in [0, 2\pi)^N \middle| \, J(\theta) \leq a \right\}$$

of the manifold defined by $J(\theta) = 0$. Then from (5.12), $\theta[0] \in \mathcal{N}(a)$ guarantees that for all $k \geq 0$, $\theta[k] \in \mathcal{N}(a)$. Then the result follows from Theorem 5.1 and Lemma 5.3.1. ■

5.4 Simulations

We now provide simulations that attest to the efficacy of the algorithm. All simulations involve 10 transmitters. In the following discussion, SNR is defined as the ratio of the per-node received power to the noise power.

Fig.5.1 shows a simulation plot of time-averaged total power at null target as a function of SNR when there are no phase drifts at the oscillators, but each of the ten transmitters sees a phase offset ϕ_i, that is uniformly distributed between 0 and $\pi/2$. The SNR limits the accuracy of the individual phase estimate and this in turn leads to fluctuations in the estimated gradient and therefore the overall received

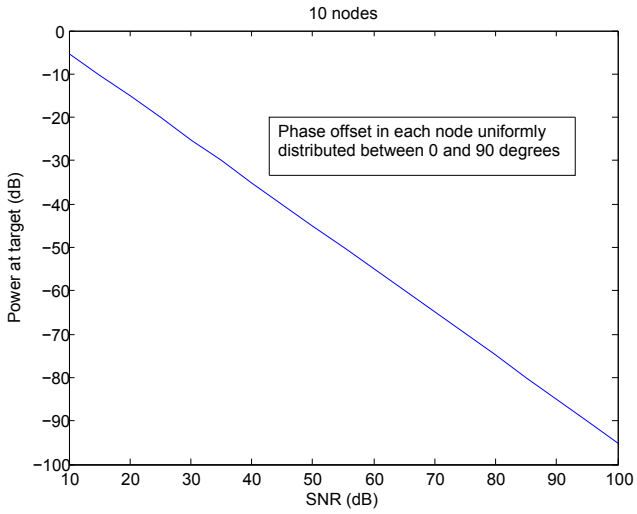

Figure 5.1: Power at null target vs. SNR.

signal strength at the null target. As expected the power at the null target decreases monotonically with increase in SNR.

Fig. 5.2 shows the variation of time-averaged total power at null target as a function of the Brownian motion phase drift for different SNRs. It can be seen that for very small Brownian motion drifts, the null power is determined by the SNR. However once drift increases to about a tenth of a degree between two iterations of the gradient descent, the null is largely limited by the drift and is more or less independent of the SNR. Observe that the highest phase drift of two degrees between phase updates corresponds to the very low feedback rate of 5 Hz, for even the cheapest of oscillators.

Fig. 5.3 is very similar to Fig. 5.2, except that unlike Fig. 5.2, that involves a

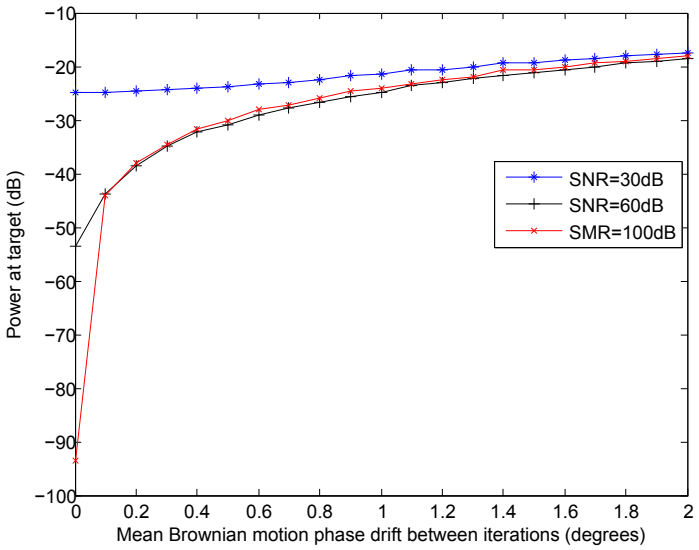

Figure 5.2: Power at null target vs. phase drift for equal channel gains.

setting where all gains are 1, in Fig. 5.3 the actual gains are obtained from a Rayleigh distribution and then equalized to one. As can be seen Fig. 5.3, the resulting potential noise amplification, has virtually no effect on the performance of the gradient descent nullforming algorithm.

Fig. 5.4 shows one simulation run of the proposed algorithm. It is evident that the algorithm converges to the practical null very rapidly, i.e., it takes less than 50 iterations to reach a power level of -36dB. The channel estimation errors at the transmitters (i.e. ϕ_i's) are assumed i.i.d U[0,π/3). The plot was obtained assuming a brownian motion phase drift between two succsessive iterations of mean zero and standard deviation 0.2 degrees. The dotted red line shows the mean power at null target after the gradient of the objective function has become very small (less than

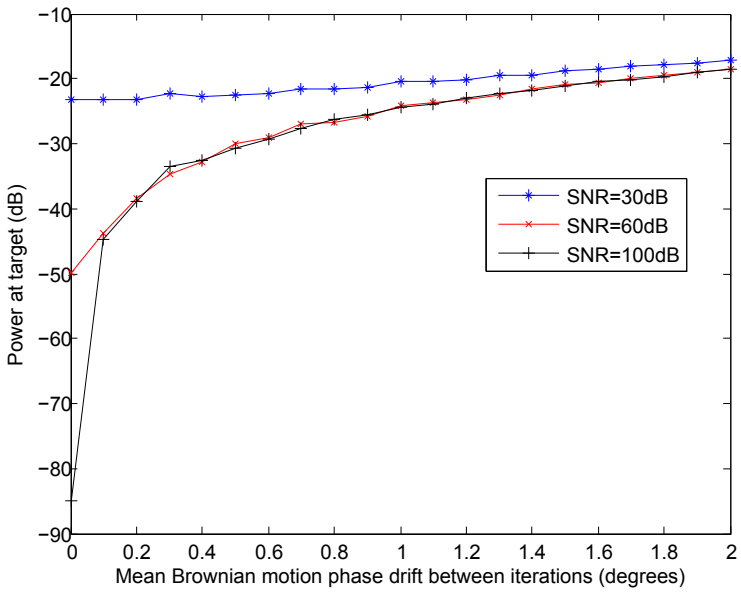

Figure 5.3: Power at null target vs. phase drift for unequal channel gains.

1e-6).

5.5 Conclusion

We have provided a new gradient descent based distributed nullforming algorithm that requires far less feedback than all its predecessors, in that each transmitter is required by this algorithm to only know its channel state information to the receiver. In constrast, previous algorithms required that channel state information to the receiver from each transmitter be known to each other transmitter in the virtual array. This coupled with the fact that it requires an additional *common* signal fed back by to all transmitters by the receiver, ensures its scalability. We have proved practical uniform convergence of the algorithm to a null. This ensures robustness to noise and channel phase estimation errors., verified by simulations, that involve

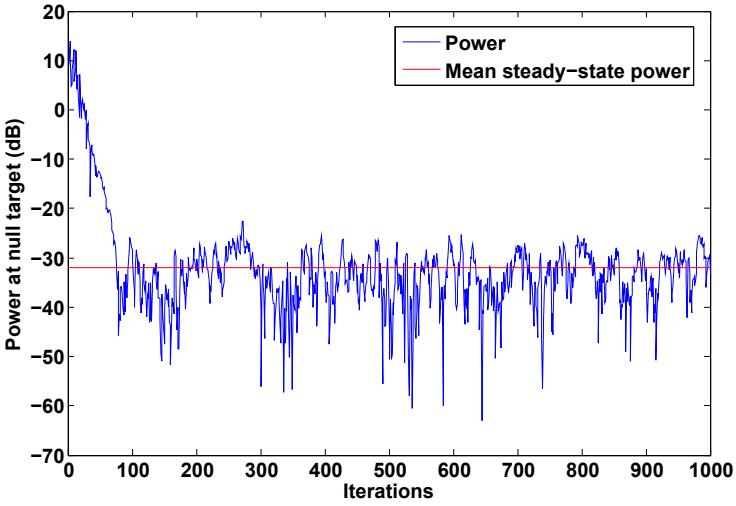

Figure 5.4: Power at null target vs. iterations.

nontrivial channel phase estimation errors compounded by Brownian motion driven oscillator drift.

CHAPTER 6
CONCLUSION AND FUTURE WORK

In this thesis, we have motivated the concept of distributed MIMO for energy-efficient and high data rate communication. Specifically, we have considered distributed MISO systems whereby a group of transmitters organizes themselves into a VAA to talk to a distant receiver. We have focused on two specific applications of VAAs, i.e., distributed transmit beamforming and distributed transmit nullforming.

The underlying challenge in realizing a VAA is that it requires frequency synchronization among the cooperating nodes; moreover, it requires precise control over the individual phases of their to-be transmitted signals. We have proposed a novel distributed consensus-based carrier synchronization algorithm that achieves frequency lock among the cooperating transmitters in a 2-node VAA globally and exponentially. We have then demonstrated the feasibility of distributed beamforming in real-time settings on GNU-radio/USRP based SDR platform. For this, we have introduced DSP-centric approach to frequency synchronization problem where VAA nodes use Extended Kalman Filters to estimate and compensate their frequency offsets w.r.t a designated receiver. To achieve phase coherence among the signals of VAA nodes at intended receiver, we have used 1-bit feedback control algorithm. Finally, we have proposed a scalable, distributed gradient-descent algorithm by means of which the VAA nodes can achieve a null at a designated null target.

6.1 Open problems

We list some open problems that arise from our work.

- **Consensus-based carrier synchronization algorithm**

  1. **Global stability analysis of N-node wireless network.** We have presented some analytical results in Chapter 2 which prove the global stability of the distributed consensus based algorithm for 2-node wireless network. Though we also have few preliminary results for 3-node network and N-node network (see Chapter 2), we leave as future work the rigorous analytical investigation of stability of N-node network.

  2. **Cross-layer protocol design for implementation.** Traditionally, different varients of master-slave architecture have been employed in wireless sensor networks whereby slave nodes use PLLs to achieve the frequency lock. It is well-known fact that PLLs acquire frequency lock only locally. On the other hand, the distributed consensus-based approach to carrier synchronization promises to achieve frequency lock among the cooperating nodes globally (as proved for N=2 nodes case in Chapter 2). Therefore, the proposed algorithm has the potential to transform the existing synchronization techniques for WSN, especially wireless ad-hoc networks. Nevertheless, the implementation of the proposed consensus-based algorithm will require novel cross-layer (MAC+PHY) protocols to coordinate the transmissions going on among every node and its neighbors in an ad-

hoc network.

- **Investigation of spatial multiplexing and its proof of concept.** The concepts of distributed transmit (and receive [17]) beamforming and distributed transmit (and receive) nullforming can be all combined together to establish a complete distributed MIMO system. Then, both the diversity gains and the gains due to spatial multiplexing can be achieved [60] to combat the fading and increase the throughput of the distributed MIMO system respectively.
- **Investigation and experimental evaluation of MIMO-OFDM techniques.** Recently, there has been a growing interest in designing efficient MIMO-OFDM systems due to achieve high data rates and reliable communication in wideband systems with frequency-selective channels [61]. Design of novel techniques for distributed MIMO-OFDM systems and their experimental evaluation is of enormous interest to the wireless research community, especially the researchers working on upcoming standards of WiFi and LTE.
- **Localization using beamforming.** Distributed transmit beamforming can be used to localize a VAA. Such localization is potentially useful in many military and WSN applications.
- **Novel techniques for phase calibration in a WSN to turn it into an VAA.** Novel techniques such as 3-way message-exchange scheme in [34] can be used to organize a set of single-antenna transmitters into a VAA.

- **Using EKF state-space model for mobile channels.** One can incorporate the time-varying channels into our existing beamforming implementation framework, i.e., when receiver and transmitters are moving w.r.t each other. [5] is a good starting point in this regard.

REFERENCES

[1] Juan A. Acebrón, L. L. Bonilla, Conrad J. Pérez Vicente, Félix Ritort, and Renato Spigler. The kuramoto model: A simple paradigm for synchronization phenomena. *Rev. Mod. Phys.*, 77:137–185, Apr 2005.

[2] G.J. Bradford and J.N. Laneman. An experimental framework for the evaluation of cooperative diversity. In *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, pages 641 –645, march 2009.

[3] G.J. Bradford and J.N. Laneman. A survey of implementation efforts and experimental design for cooperative communications. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 5602 –5605, march 2010.

[4] D.R. Brown, P. Bidigare, S. Dasgupta, and U. Madhow. Receiver-coordinated zero-forcing distributed transmit nullforming. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 269–272, 2012.

[5] D.R. Brown, P. Bidigare, and U. Madhow. Receiver-coordinated distributed transmit beamforming with kinematic tracking. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5209–5212, 2012.

[6] D.R. Brown, U. Madhow, P. Bidigare, and S. Dasgupta. Receiver-coordinated distributed transmit nullforming with channel state uncertainty. In *Information Sciences and Systems (CISS), 2012 46th Annual Conference on*, pages 1–6, 2012.

[7] J. Buck. Synchronous rhythmic flashing of fireflies. *The Quarterly Review of Biology*, pages 265289, 63(3), 1988.

[8] E. Butterline and S.L. Frodge. Gps- synchronizing our telecommunications networks. In *Proceedings of the 12th International Technical Meeting of the Satellite Division of ION GPS 1999*, page 597606. ION GPS, 1999., September 1999.

[9] D. Cavendish. Evolution of optical transport technologies: from sonet/sdh to wdm. *Communications Magazine, IEEE*, 38(6):164 –172, jun 2000.

[10] J.P. Costas. Synchronous communications. *Proceedings of the IRE*, 44(12):1713 –1718, dec. 1956.

[11] T. Cover and A.E. Gamal. Capacity theorems for the relay channel. *Information Theory, IEEE Transactions on*, 25(5):572–584, 1979.

[12] Lun Dong, Zhu Han, A.P. Petropulu, and H.V. Poor. Cooperative jamming for wireless physical layer security. In *Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on*, pages 417–420, 2009.

[13] F. Dorfler and F. Bullo. Synchronization and transient stability in power networks and non-uniform Kuramoto oscillators. 50(3):1616–1642, 2012.

[14] A. El Gamal and T.M. Cover. Multiple user information theory. *Proceedings of the IEEE*, 68(12):1466 – 1483, dec. 1980.

[15] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Oper. Syst. Rev.*, 36:147–163, December 2002.

[16] USRP products. http://www.ettus.com/products, 2011.

[17] U. Madhow F. Quitin, A. Irish. Distributed receive beamforming: a scalable architecture and its proof of concept. In *77th IEEE Vehicular Technology Conference, 2013*, 2012.

[18] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388 –404, mar 2000.

[19] Khalil H. Prentice Hall, 3rd edition edition, 2002.

[20] W. Hahn. Springer, 1st edition edition, 1967.

[21] Ahmed Hemani, Axel Jantsch, Shashi Kumar, Adam Postula, Johnny Oberg, Mikael Millberg, and Dan Lindqvist. *Network on chip: An architecture for billion transistor era*, page 166173. Citeseer, 2000.

[22] 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*, 49(2):102–111, february 2011.

[23] A. Jadbabaie, Jie Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988 – 1001, june 2003.

[24] A. Jadbabaie, N. Motee, and M. Barahona. On the stability of the kuramoto model of coupled nonlinear oscillators. In *American Control Conference, 2004. Proceedings of the 2004*, volume 5, pages 4296–4301 vol.5, 2004.

[25] V. Jungnickel, L. Thiele, T. Wirth, T. Haustein, S. Schiffermuller, A. Forck, S. Wahls, S. Jaeckel, S. Schubert, H. Gabler, C. Juchems, F. Luhn, R. Zavrtak, H. Droste, G. Kadel, W. Kreher, J. Mueller, W. Stoermer, and G. Wannemacher. Coordinated multipoint trials in the downlink. In *GLOBECOM Workshops, 2009 IEEE*, pages 1 –7, Dec, 2009 2009.

[26] Kalibrate. http://thre.at/kalibrate, 2010.

[27] S. Kay. A fast and accurate single frequency estimator. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(12):1987 –1990, Dec 1989.

[28] T. Korakis, M. Knox, E. Erkip, and S. Panwar. Cooperative network implementation using open-source platforms. *Communications Magazine, IEEE*, 47(2):134 –141, february 2009.

[29] J.N. Laneman and G.W. Wornell. Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks. *IEEE Transactions on Information Theory*, 49(10):2415 – 2425, oct. 2003.

[30] B.P. Lathi and Ding Z. Oxford Unniversity Press, 2008.

[31] J. Lin, A.S. Morse, and B.D.O. Anderson. The multi-agent rendezvous problem - the asynchronous case. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 2, pages 1926 – 1931 Vol.2, dec. 2004.

[32] J. Mietzner, J. Eick, and P.A. Hoeher. On distributed space-time coding techniques for cooperative wireless networks and their sensitivity to frequency offsets. In *Smart Antennas, 2004. ITG Workshop on*, pages 114 – 121, march 2004.

[33] Renato E. Mirollo and Steven H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM J. Appl. Math.*, 50:1645–1662, November 1990.

[34] R. Mudumbai, G. Barriac, and U. Madhow. On the feasibility of distributed beamforming in wireless networks. *IEEE Trans. on Wireless Communication*, 6(5):1754–1763, May 2007.

[35] R. Mudumbai, P. Bidigare, S. Pruessing, S. Dasgupta, M. Oyarzun, and D. Raeman. Scalable feedback algorithms for distributed transmit beamforming in wireless networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5213–5216, 2012.

[36] R. Mudumbai, D.R. Brown, U. Madhow, and H.V. Poor. Distributed transmit beamforming: challenges and recent progress. *Communications Magazine, IEEE*, 47(2):102 –110, February 2009.

[37] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac. Scalable feedback control for distributed beamforming in sensor networks. In *IEEE International Symp. on Information Theory (ISIT)*, pages 137–141, Adelaide, Australia, September 2005.

[38] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac. Distributed transmit beamforming using feedback control. *IEEE Trans. on Inform. Theory*, 56(1):411–426, January 2010.

[39] R. Mudumbai, B. Wild, U. Madhow, and K. Ramchandran. Distributed beamforming using 1 bit feedback: from concept to realization. In *44th Allerton Conf. on Comm., Control, and Computing*, pages 1020 – 1027, Monticello, IL, Sep. 2006.

[40] P. Murphy and A. Sabharwal. Design, implementation, and characterization of a cooperative communications system. *IEEE Transactions on Vehicular Technology*, 60(6):2534 –2544, july 2011.

[41] V N Murthy and E E Fetz. Synchronization of neurons during local field potential oscillations in sensorimotor cortex of awake monkeys. *Journal of Neurophysiology*, 76(6):3968–3982, 1996.

[42] Datasheet for EC2620ETTS-64.000M oscillator. http://www.ecliptek.com, 2011.

[43] A. Ozgur, O. Leveque, and D.N.C. Tse. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *Information Theory, IEEE Transactions on*, 53(10):3549–3572, 2007.

[44] A. Ozgur, O. Leveque, and D.N.C. Tse. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *IEEE Transactions on Information Theory*, 53(10):3549–3572, Oct. 2007.

[45] Ipek Ozil and D.R. Brown III. Time-slotted round-trip carrier synchronization. In *Proceedings of the 41st Asilomar Conference on Signals, Systems, and Computers*, pages 1781 – 1785, Pacific Grove, CA, November 4-7, 2007.

[46] C.S. Pittendrigh. Circadian rhythms and the circadian organization of living systems. In *In Cold Spring Harbor Symposia on Quantitative Biology*, volume 25, page 159. Cold Spring Harbor Laboratory Press, 1960., 1960.

[47] F. Quitin, U. Madhow, M.M.U. Rahman, and R. Mudumbai. Demonstrating distributed transmit beamforming with software-defined radios. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–3, 2012.

[48] F. Quitin, M.M.U. Rahman, R. Mudumbai, and U. Madhow. Distributed beamforming with software-defined radios: frequency synchronization and digital feedback. In *Globecom, 2012*, 2012.

[49] Francois Quitin, Muhammad Mahboob Ur Rahman, Raghuraman Mudumbai, and Upamanyu Madhow. A scalable architecture for distributed transmit beamforming with commodity radios: Design and proof of concept. *Wireless Communications, IEEE Transactions on*, 12(3):1418–1428, 2013.

[50] Muhammad Mahboob Ur Rahman, Hernest Ernest Baidoo-Williams, Soura Dasgupta, and Raghuraman 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.

[51] Rakon RFPO45 SMD oven controlled crystal oscillator datasheet. http://www.rakon.com, 2009.

[52] B. Razavi. Challenges in the design of frequency synthesizers for wireless applications. In *Proceedings of the IEEE Custom Integrated Circuits Conference*, pages 395 –402, may 1997.

[53] Wei Ren, R.W. Beard, and E.M. Atkins. Information consensus in multivehicle cooperative control. *Control Systems Magazine, IEEE*, 27(2):71 –82, april 2007.

[54] Warp: Wireless open access research platform. http://warp.rice.edu/trac, 2011.

[55] D. Rife and R. Boorstyn. Single tone parameter estimation from discrete-time observations. *Information Theory, IEEE Transactions on*, 20(5):591 – 598, Sep 1974.

[56] Thomas Schmid, Oussama Sekkat, and Mani B. Srivastava. An experimental study of network performance impact of increased latency in software defined radios. In *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, WinTECH '07, pages 59–66, 2007.

[57] A. Sendonaris, E. Erkip, and B. Aazhang. Increasing uplink capacity via user cooperation diversity. In *Proc. 1998 IEEE International Symp. on Information Theory*, page 156, Cambridge, MA, August 1998.

[58] Munkyo 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*, pages 683 –686, june 2008.

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

[60] Q.H. Spencer, A.L. Swindlehurst, and M. Haardt. Zero-forcing methods for downlink spatial multiplexing in multiuser mimo channels. *Signal Processing, IEEE Transactions on*, 52(2):461–471, 2004.

[61] G.L. Stuber, R. Barry, S.W. McLaughlin, Ye Li, D. Brunelli, and T.G. Pratt. Broadband mimo-ofdm wireless communications. *Proceedings of the IEEE*, 92(2):271–294, 2004.

[62] T.H. Summers, Changbin Yu, S. Dasgupta, and B. D O Anderson. Control of minimally persistent leader-remote-follower and coleader formations in the plane. *Automatic Control, IEEE Transactions on*, 56(12):2778–2792, 2011.

[63] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1229, Aug 1995.

[64] Geoffrey Werner-Allen, Geetika Tewari, Ankit Patel, Matt Welsh, and Radhika Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, pages 142–153, New York, NY, USA, 2005. ACM.

[65] H. Yamaguchi and G. Beni. Distributed anutonomous formation control of mobile robot groups by swarmbased pattern generation. In *In Proc. of DARS-96*, pages 141–155. Springer Verlag, 1996., 1996.

[66] D.W. Youngner, L. Lust, D.R. Carlson, S.T. Lu, L.J. Forner, H.M. Chanhvongsak, and T.D. Stark. A manufacturable chip-scale atomic clock. In *International Solid-State Sensors, Actuators and Microsystems Conference, TRANSDUCERS*, pages 39 –44, june 2007.

[67] T. Yucek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys Tutorials, IEEE*, 11(1):116–130, 2009.

[68] Per Zetterberg, Christos Mavrokefalidis, Aris S. Lalos, and Emmanouil Matigakis. Experimental investigation of cooperative schemes on a real-time dsp-based testbed. *EURASIP J. Wireless Comm. and Networking*, 2009.

[69] H. Zhang and H. Dai. On the capacity of distributed mimo systems. In *Proc. Conference on Information Sciences and Systems*, 2004.

[70] Jin Zhang, Juncheng Jia, Qian Zhang, and E.M.K. Lo. Implementation and evaluation of cooperative communication schemes in software-defined radio testbed. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, march 2010.

[71] C. Zucca and P. Tavella. The clock model and its relationship with the allan and related variances. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 52(2):289 –296, feb. 2005.