Technical Report for Superposition Coding Strategies: Design and Experimental Evaluation

Z. Gong, M. Haenggi, S. Srinivasa, K. Stamatiou, S. Vanka, P. Vizi

Abstract

We design and implement a software-radio sytem for Superposition Coding (SC), a multiuser transmission scheme that deliberately introduces interference among user signals at the transmitter, using a library of off-the-shelf point-to-point channel codes. We experimentally determine the set of rate-pairs achieved by this transmission scheme under a packet-error constraint. Our results suggest that SC can provide substantial gains in spectral efficiencies over those achieved by orthogonal schemes such as Time Division Multiplexing. Our findings also question the practical utility of the Gaussian approximation for the inter-user interference in Superposition-Coded systems.

Keywords—Superposition Coding, Software-Defined Radio, GNURadio, Universal Software Radio Peripheral.

I. Introduction

The problem of communicating with many receivers arises in many "downlink" scenarios such as communication from an access point to stations in WiFi or from a base station in cellular systems. The conventional approach is to set up orthogonal channels to each user by time/frequency/code-division multiplexing. Although this approach eliminates interference between transmissions, it does not in general achieve the highest possible transmission rates for a given packet error rate (or reliability) [1]. In fact, Superposition Coding (SC) [2] is a well-known non-orthogonal scheme that achieves the capacity on a scalar Gaussian broadcast channel.

We motivate the use of SC for the two-receiver case. Consider a cellular downlink with several active users. Given the user density in typical networks, it is always possible to pick two users N (the "near" user) and F (the "far" user), as shown in Figure 1. The key observation here is that N being geographically closer to the base station (BS) has a "stronger" (less noisy) link to the BS than F; thus any packet that can be decoded at F can most probably be decoded at N as well (but not vice versa). The idea behind SC is to optimally exploit this channel ordering.

A BS that uses two-receiver SC can transmit *superimposed* F and N packets (or more precisely, the far and near user codewords) in *both* F's and N's time slots (see Figure 1). Thus both links enjoy the *combined* degrees of freedom available to N and F, while sharing the transmit power. For large blocklengths, it can be shown that it is possible to encode F's packets such that they can be decoded in the presence of interference from N's packets. Since N has a stronger link to the BS, N can replicate this step to regenerate and thereby cancel F's signal from its received signal. It can then decode its own packet. This is the well-known *Successive Decoding* (SD) or *Successive Interference Cancellation* (SIC) procedure [1].

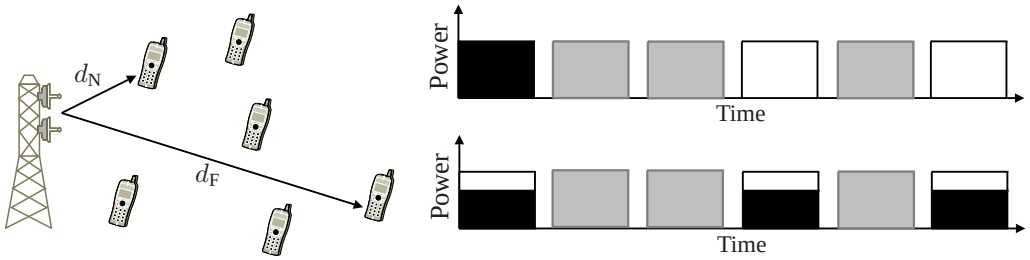

Figure 1. Illustration of two-user SC. (Left) The users N and F picked are at distances d_{N} and d_{F} respectively with $d_{\mathrm{N}} < d_{\mathrm{F}}$. (Right) Typical transmission timelines with and without SC. The gray slots represent transmissions to other active users which can remain unchanged. With Time-Division (TD, top), N and F are served in different slots (black and white). With SC (bottom), the BS transmits a linear combination of individually-coded user waveforms.

We can extend this two-user scheme to any number of users. In fact, SC (combined with SD) achieves the capacity on a scalar Gaussian broadcast channel. This implies that *any* TD-achievable rate-pair (i.e., the pair of *spectral efficiencies* on a Gaussian channel) can also be achieved using SC, with the rate gain over TD increasing with the disparity in the user link qualities.

While information theory sufficiently motivates the use of SC, it is largely silent on practical issues such as finite block length codes, finite encoding and decoding complexity, hardware non-idealities (e.g., carrier frequency offset, phase noise) that one would encounter while designing such a system. This motivates the experimental study of SC.

For rapid prototyping and streamlining the design effort, we adopt a software-defined radio (SDR) [3] paradigm using the well-known open-source GNU Radio platform in conjunction with the Universal Software Radio Peripheral (USRP) hardware board that serves as an analog and RF front-end [4]. A well-known prototyping system [5], [6], it has been recently used in testbed design, including UT Austin's Hydra [7] and by Bell Labs and Microsoft Research [8].

A. Main Contributions

Building on our previous work [9], the main contributions of this work are as follows:

1) We propose a design technique for SC using a finite library of finite-blocklength point-to-point codes designed for finite constellations.
2) We design and implement a system architecture and implement it on GNURadio/USRP and determine the experimentally-achieved set of spectral efficiencies for a packet-error constraint. To the best of our knowledge, ours is the first attempt at systematically designing and characterizing an SC physical layer that, along with its attendant hardware, forms a functioning system.
3) We study the implications of using a finite constellation along with a demodulate-and-decode receiver architecture on the statistics of the interference-plus-noise term.

B. Report Overview

The remainder of the report is organized as follows. In Section II we briefly summarize how SC achieves capacity and discuss some implications of restricting the library of codes to be a finite set of finite-blocklength codes. In Section III, while retaining the rate-centric approach, we propose a design technique for SC with such a finite code library and specialize this technique to a library comprised of a well-known family of codes designed using the Bit Interleaved Coded Modulation (BICM) technique [11], and predict the theoretically achievable rate region. In Section IV, we describe the system architecture that uses these BICM codes to implement SC. The transmitter and receiver design of the superposition coding system is introduced in Section V. In Section VI, we introduce the GNU Radio blocks we have implemented for SC. Hardware evaluations including transmit power calibration, noise figure measurement, and SNR estimation are presented in Section VII. In Section VIII, we present an experimental setup that emulates a Gaussian BC and use it to experimentally determine the achievable spectral efficiency pairs for a two-receiver BC under a packet-error constraint. The resulting rate region is the finite-library analog of the information-theoretic rate region. We also discuss some practical issues that arise in the design of superposition-coded systems, including the validity of treating inter-user interference as Gaussian noise. In Section IX we conclude the report and suggest possible avenues for future work.

II. Superposition Coding: From Theory to Practice

We will briefly summarize relevant results from [1], [2] on achieving the capacity of the (scalar) Gaussian Broadcast Channel (BC) using SC with SD. In addition to making this report self-contained, this discussion identifies the key architectural building blocks of a superposition-coded system. A closer examination of the blocks allows us to identify some key practical issues in implementing this ideal scheme.

We use calligraphic fonts (e.g., $\mathcal{C}$) to represent sets and sans-serif fonts (e.g., $\mathsf{f}(\cdot)$) to denote the encoding/decoding maps. Also, we use $[M]$ to represent $\{1, \cdots, M\}$ for $M \in \mathbb{Z}^+$, and occasionally use the short-hand Tx for a transmitter and Rx for a receiver.

A. An Information-Theoretic View of Superposition Coding on the Gaussian Channel

Consider a BS that wants to communicate with two receivers N and F. The broadcast nature of the wireless medium is captured by the broadcast channel model $X \to (Y_{\mathrm{N}}, Y_{\mathrm{F}})$ where X denotes the channel input and Y_{N} and Y_{F} are the channel outputs at N and F[1]. Let $(X(n))$ be a sequence of channel inputs indexed by the channel use $n \in [L]$. Clearly $(X(n))$ must encode information relevant to each user. The *capacity region* of this channel is the closure of the set of all possible pairs of transmission rates at which the BS can reliably send two independent information streams, one each to N and F (allowing $L \to \infty$).

For a Gaussian BC, we have

$$Y_{\mathrm{N}}(n) = h_{\mathrm{N}} X(n) + Z_{\mathrm{N}}(n); \quad Y_{\mathrm{F}}(n) = h_{\mathrm{F}} X(n) + Z_{\mathrm{F}}(n) \tag{1}$$

[1]In practical terms, $X(n)$ can be understood as a (coded) symbol stream from the BS, and the Y's as the corresponding noisy and/or distorted observations of this symbol stream at N and F.

where N (resp. F) has a complex channel gain h_{N} (resp. h_{F}) and $Z_{\mathrm{u}}, \mathrm{u} \in \{\mathrm{N}, \mathrm{F}\}$ denote the WGN processes. We assume the BS operates with an average power constraint P [W] and a (baseband) bandwidth W [Hz], and denote the noise power spectral density by N_0 [W/Hz]. From the above, the power constraint per channel use is P/W and $\mathbb{E}[|Z(n)|^2] = N_0 W$. From the definition of N and F, $|h_{\mathrm{N}}|^2 < |h_{\mathrm{F}}|^2$.

One way for $(X(n))$ to encode information is to communicate with each user in turns by partitioning the total number of channel uses into time slots (as in TD). For a given n, $X(n)$ contains information pertaining to just one user. This is the well-studied point-to-point communication problem, for which good practical encoding and decoding schemes exist. However, for a BC it is known that TDM is suboptimal in general; the root cause lies in its inability to fully exploit the fact that $|h_{\mathrm{N}}| > |h_{\mathrm{F}}|$: N has a "stronger" channel to BS, and hence can always decode information that can be decoded at F. This makes the scenario ideal for the SC scheme which achieves every pair of transmission rates in the capacity region.

The key architectural elements of an SC system are:

1) A *superposition encoder* f that consists of
   a) Two point-to-point encoders, $\mathsf{f}_{\mathrm{N}} \colon \{0,1\}^{\lfloor 2^{LR_{\mathrm{N}}} \rfloor} \to \mathbb{C}^L$ (which we call the *near-encoder*) and $\mathsf{f}_{\mathrm{F}} \colon \{0,1\}^{\lfloor 2^{LR_{\mathrm{F}}} \rfloor} \to \mathbb{C}^L$ (which we call the *far-encoder*), that map their respective inputs (the *near-* and *far-*messages) to complex-valued sequences $(X_{\mathrm{N}}(n))$ and $(X_{\mathrm{F}}(n))$, each of *block length* L. Here R_{N} and R_{F} denote the bandwidth-normalized transmission rates (or spectral efficiencies) of N and F (the *near-* and *far-rates* for short).
   b) A summation device that outputs a sequence
   $$X(n) = \sqrt{1-\alpha}\, X_{\mathrm{F}}(n) + \sqrt{\alpha}\, X_{\mathrm{N}}(n), \tag{2}$$
   where a fraction $\alpha \in [0,1]$ of the power is assigned to N(the *near-fraction* for short).
2) A *single-user* decoder $\mathsf{g}_{\mathrm{F}} : \mathbb{C}^L \to \{0,1\}^{2^{LR_{\mathrm{F}}}}$ that estimates the far packet from the observations $(Y_{\mathrm{F}}(n))$ by treating $(X_{\mathrm{N}}(n))$ as Gaussian noise.
3) A *successive cancellation* decoder $\mathsf{g}_{\mathrm{F,N}} : \mathbb{C}^L \to \{0,1\}^{2^{LR_{\mathrm{N}}}}$ that is used to recover N's packet in the following steps:
   a) Decode F's packet using the single-user decoder g_{F}.
   b) Cancel $\sqrt{1-\alpha} h_{\mathrm{N}} X_{\mathrm{F}}(n)$ from $Y_{\mathrm{N}}(n)$ by regenerating $X_{\mathrm{F}}(n)$ using the far-encoder f_{F} and the knowledge of h_{N} and α:
   $$Y'_{\mathrm{N}}(n) = Y_{\mathrm{N}}(n) - h_{\mathrm{N}}\sqrt{1-\alpha}\, X_{\mathrm{F}}(n) = h_{\mathrm{N}}\sqrt{\alpha}\, X_{\mathrm{N}}(n) + Z_{\mathrm{N}}(n). \tag{3}$$
   c) Decode N's packet using the single-user decoder $\mathsf{g}_{\mathrm{N}} : \mathbb{C}^L \to \{0,1\}^{2^{LR_{\mathrm{N}}}}$.

It is well known that as $L \to \infty$, for all α there exist $\mathsf{f}_{\mathrm{N}}, \mathsf{f}_{\mathrm{F}}, \mathsf{g}_{\mathrm{F}}, \mathsf{g}_{\mathrm{N}}$ such that communication can occur arbitrarily reliably for all pairs of transmission rates satisfying

$$R_{\mathrm{N}} < W \log_2 (1 + \alpha\gamma_{\mathrm{N}}); \quad R_{\mathrm{F}} < W \log_2 \left(1 + \frac{(1-\alpha)\gamma_{\mathrm{F}}}{\alpha\gamma_{\mathrm{F}} + 1}\right), \qquad \alpha \in [0,1], \tag{4}$$

where

$$\gamma_{\mathrm{N}} \triangleq \frac{P|h_{\mathrm{N}}|^2}{N_0 W} \quad \text{and} \quad \gamma_{\mathrm{F}} \triangleq \frac{P|h_{\mathrm{F}}|^2}{N_0 W}$$

represent the *near-* and *far-SNRs* respectively. We are interested in the spectral efficiencies

$$r_{\mathrm{u}} = R_{\mathrm{u}}/W, \qquad \mathrm{u} \in \{\mathrm{N}, \mathrm{F}\} \tag{5}$$

which we will simply call *rates*. Clearly, making α discrete would also make the rate region boundary discrete. The following subsections elaborate on this issue.

B. Practical Design Issues

As noted above, a discrete α results in a discrete set of rate-pairs defining the corner points of the achievable rate region[2]. The lowest (resp. highest) value of α restricts the minimum (resp. maximum) power that can be assigned to a user for superposed transmission. For finite block lengths there is also a non-zero probability of decoding error.

With these practical constraints factored in, the rate benefits from SC over TD will depend on α, the chosen discrete set of codes and the system implementation. Using a combination of theory, simulations and experiments, we show that it is indeed possible to build efficient superposition-coded systems using off-the-shelf single-user coding and decoding techniques.

III. Designing a Superposition-Coded System

Based on our observations in the previous section, we first describe our design approach in its full generality in Section III-A. In Section III-B we illustrate this approach for a well-known single-user coded modulation technique known as Bit-Interleaved Coded Modulation (BICM). Indeed, good practical codes designed using BICM techniques form the basis for error correction in many real-world wireless networks (see, e.g., [11] and the references therein).

A. Practical SC using a Finite Code Library

We assume all codes have a *block length* $L < \infty$ [channel uses] and fix a *target Packet Error Rate (PER)* $\epsilon \ll 1$. The latter is the probability that a user cannot decode its packet[3]. Define a *code library* as a collection $\mathcal{C}$ of $M > 1$ *single-user* encoder-decoder function pairs[4] $(\mathsf{f}(\cdot), \mathsf{g}(\cdot))$ (a "code" for short). These codes are ordered by their rates $r_1 < r_2 < \cdots < r_M$ [bps/Hz]. We label each code by its *rate index* $i \in [M]$. We say a *rate index* i is ($\epsilon-$)*feasible* on a link if the receiver can decode a packet encoded at rate r_i with a PER no larger than ϵ. In the following, denote the user rate, the corresponding rate index and the PER by r_{u}, i_{u} and $\mathrm{PER}_{\mathrm{u}}$ for $\mathrm{u} \in \{\mathrm{N}, \mathrm{F}\}$.

[2]Time-sharing can be used to convexify this boundary

[3]We asssume one packet is encoded as one codeword. Hence packet error is equivalent to codeword error.

[4]In practical terms, an encoding function is a mapping from the packet bits to the signal waveforms induced by the encoder and the modulator. A decoding function is the detection rule to perform the inverse operation, and is induced by the demodulator and decoder.

To maximize the potential benefit from SC, we assume that all rates from the library are feasible on the near link; in particular, the rate index i in single-user mode is feasible with SNR no more than $\beta_{\mathrm{N}}(i)\gamma_{\mathrm{N}}$, $\beta_{\mathrm{N}}(i) \leq 1$ for $i \in [M]$[5]. However, F must be chosen with some care. If it is too far away from the BS, even a small amount of near-user interference (from superposition) can render even r_1 infeasible. If F is too close, there is not enough disparity between the links to N and F to take the full advantage of superposition[6]. Keeping this in mind, we pick F such that the far-link supports a rate index $K < M$ in the single-user mode. Suppose the single-user mode is feasible on the far link with SNRs $\beta_{\mathrm{F}}(i)\gamma_{\mathrm{F}}$, $\beta_{\mathrm{F}}(i) \leq 1$ for $i \in [K]$. When the BS serves N and F using TD, it uses rate index M to serve N and rate index K to serve F. In this case, the fraction of time slots alloted to each link determines the effective rate. We now fix the values of α. Since $\mathcal{C}$ has only M possible rate indices, all of which are feasible at N, there are only M meaningful values of α. Assuming small levels of residual interference after canceling F, $\alpha(k) \gtrapprox \beta_{\mathrm{N}}(k), k \in [M]$[7]. So for each $m \in [M]$, we start with an initial guess $\alpha(m) = \beta_N(m)$ and find the largest far-user rate r_{F} such that the PER constraints at N and F are simultaneously satisfied. Thus we have a set of M separate optimization problems, whose m^{th} problem is posed as

$$\max_{\{r_1,\cdots,r_M\}} r_{\mathrm{F}} \quad \text{s.t.} \quad r_{\mathrm{N}} = r_k, \mathrm{PER}_{\mathrm{N}}(m) \leq \epsilon, \mathrm{PER}_{\mathrm{F}}(m) \leq \epsilon. \tag{6}$$

We denote the rate indices of the solutions of (6) by $(i^*_{\mathrm{F}}(k))$. To meet the constraint on $\mathrm{PER}_{\mathrm{N}}(m)$, $\alpha(m)$ may need to be increased slightly from $\beta_{\mathrm{N}}(m)$. Obtaining $i^*_{\mathrm{F}}(m)$ yields a possible $\alpha^*(m)$. Even after perfect cancellation of F's interference, N requires an SNR of at least $\gamma_{\mathrm{N}}\beta_{\mathrm{N}}(k)$; hence

$$\alpha^*(m) \geq \beta_{\mathrm{N}}(m), \qquad m \in [M]. \tag{7}$$

Also observe that since at most a fraction $1 - \beta_{\mathrm{N}}(m)$ of the transmit power is available for F,

$$i^*_{\mathrm{F}}(m) \leq \max\{l : 1 - \beta_{\mathrm{N}}(m) > \beta_{\mathrm{F}}(l)\}$$

even if we ignored N's interference at F.

The solution set of (6) is the set of optimal rate pairs $\{(r_m, r_{i^*_{\mathrm{F}}(m)}) : m \in [M]\}$. Combining the achievable endpoints $(r_M, 0)$ and $(0, r_K)$, we obtain the *rate-region* of this code library $\mathcal{R}^*(\mathcal{C}; \epsilon, L, \gamma_{\mathrm{N}}, \gamma_{\mathrm{F}}) \triangleq \mathcal{R}^*$ as

$$\mathcal{R}^* \triangleq \mathrm{Conv}(\{(0,0), (r_M, 0), \{(r_k, r_{i^*_{\mathrm{F}}(k)}) : m \in [M]\}, (0, r_K)\})). \tag{8}$$

where $\mathrm{Conv}(\cdot)$ denotes the convex hull operator. Figure 2 graphically summarizes the design procedure.

In the following subsection we pose (6) for a code library consisting of point-to-point codes based on BICM.

[5]A reasonable assumption, given the number of proximate receivers in typical urban WiFi or cellular networks

[6]We study the implications of a poor choice of F in Section VIII-D1.

[7]We find that this is indeed the case in Sec. V-C2

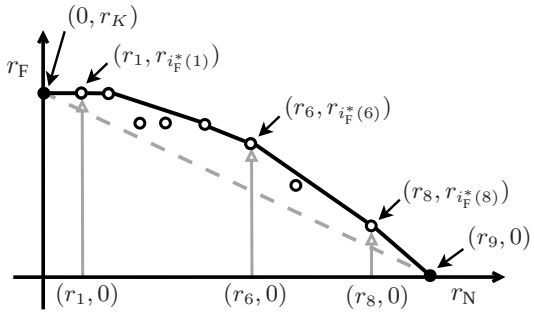

Figure 2. A graphical illustration of the rate region obtained by the design process in Section III-A for a hypothetical code library with $M = 9$ codes. The single-user points (filled dark circles) are $(r_9, 0)$ and $(0, r_K)$, for some $K \leq 9$. The gray dashed line joining these two points represents the rate region achievable by TD. The open circles represent the optimal rate-pairs obtained by solving (6) as N's rate is varied from r_1 through r_8. To obtain $\mathcal{R}$ using (8) we find the convex hull of this optimal set (along with the single-user points and the origin) to obtain a convex polygon with vertices $\{(r_k, r_{i_{\mathrm{F}}^*(k)}) : k \in \{2, 5, 6, 8\}\}$ inside the positive quadrant. The solid black lines represent the rate region boundary in this quadrant.

B. SC using a BICM Code Library

Based on the framework developed so far, we revisit the key elements discussed in Section II-A to map them to specific subsystems in a Superposition-Coded-BICM (SC-BICM) system; we also introduce some terminology specific to BICM. Subsequently we solve (6) numerically for BICM codes in the high-reliability regime ($\epsilon \to 0$). For reliabilities of practical interest (e.g., $\epsilon \lessapprox 0.1$), we solve this problem via Monte Carlo simulations for a specific class of BICM codes we have implemented on our testbed.

1) The SC-BICM System: The channel coding in a canonical BICM system [13] is specified by:

- c: Convolutional code with code-rate ρ_{c}, free distance d_{c} and number of free-distance error events w_{c}.
- x: A gray-coded constellation mapper that maps b_{x} interleaved coded bits mapped to each symbol. The symbols have unit average energy and a minimum intersymbol Euclidean distance of d_{x}[8]. We will refer to the range $\mathcal{S}_{\mathsf{x}}$ of x as its *constellation.* For example, the constellation of a BPSK mapper is $\{-1, 1\}$.
- Π: An interleaver matrix that specifies the order in which code bits are read by the constellation mapper. The corresponding de-interleaver is denoted by Π^{-1}.

Combining a convolutional code c with a modulator x results in a spectral efficiency of $r = \rho_{\mathsf{c}} b_{\mathsf{x}}/W$ [bps/Hz]. Thus the near-encoder f_{N} is the composition $\mathsf{x}_{\mathrm{N}} \circ \Pi_{\mathrm{N}} \circ \mathsf{c}_{\mathrm{N}}$. A block length L can encode $B = Lr$ information bits per codeword or packet. With a slight abuse of notation, we denote the Viterbi decoder and the constellation demapper by c^{-1} and x^{-1} respectively.

We can now adequately describe the key functional units in an SC-BICM system: the superposition encoder f finds the weighted sum of the outputs of two separate BICM encoders f_{N} and f_{F} to construct X which we will now call the *composite* symbol stream. The composite symbol stream is mapped to a waveform and transmitted over the wireless medium. At F, Y_{F} in (1) represents the noisy observations of X seen by the constellation demapper (that is

[8]The gain of the subsequent Tx stages controls the overall transmit power.

the first stage in g_{F}). From Y_{F} the demapper estimates the reliabilities of each encoded bit of the far-packet, treating the symbol stream $(X_{\mathrm{N}}(n))$ as interference. The remaining steps are the same as in standard BICM decoding.

Since g_{F} is part of the successive cancellation decoder $\mathsf{g}_{\mathrm{F,N}}$, these steps are reproduced at N; thereafter the far-encoder f_{F} (at N) reconstructs X_{F} from this decoded packet which is then subtracted from Y_{N} to yield Y'_{N} as in (3). In the final step g_{N} estimates the near-packet from Y'_{N} using standard BICM decoding.

2) A Theoretical Estimate of the Rate Region of an SC-BICM System in the High Reliability Regime: Given $\mathcal{C}$, (6) can be solved by checking the feasibility[9] of every candidate rate pair via (time-consuming) Monte Carlo simulation. One could reduce this computational overhead by reducing the search space of possible far-rates for a given near-rate. In fact, a formula to compute the PER for a given rate pair and near-fraction would obviate the need for simulations.

Unfortunately, accurate formulas for the PER are difficult to obtain even for the point-to-point case, although there exist well-known upper bounds that are asymptotically tight in the high-reliability regime (PER $\to 0$) [12]. In the high-reliability regime, these upper bounds can be suitably modified, as we will show.

The key difference between SC-BICM and point-to-point BICM lies in the far-demodulator $\mathsf{x}_{\mathrm{F}}^{-1}$ that estimates the reliabilities of the far-code bits from observations of the form

$$Y_{\mathrm{F}}(n) = \underbrace{h_{\mathrm{F}}\sqrt{1-\alpha}X_{\mathrm{F}}(n)}_{\text{Signal}} + \underbrace{h_{\mathrm{F}}\sqrt{\alpha}X_{\mathrm{N}}(n) + Z_{\mathrm{F}}(n)}_{\text{Perturbation}}, \quad n \in [L]. \tag{9}$$

Unlike in point-to-point BICM, the perturbation term is not Gaussian; its statistics depend on N's constellation, which we assume is known to the demodulator[10]. Whether or not x_{N} is known, the finiteness of the interference constellation raises an interesting question about the validity of the Interference-As-Gaussian-Noise (IAGN) model that assumes such Gaussianity. We investigate this question in greater detail in Section VIII-D3.

When N's constellation is known, it is useful to treat each composite symbol $X(n) = \sqrt{1-\alpha}X_{\mathrm{F}}(n) + \sqrt{\alpha}X_{\mathrm{N}}(n)$ as a member of a superconstellation with $2^{b_{\mathsf{x}_{\mathrm{N}}}+b_{\mathsf{x}_{\mathrm{F}}}}$ points (see also Figure. 17). Viewed from the demodulator, interference perturbs each original far-symbol (the *parent* point) to a randomly chosen *daughter* point. For each parent point, define the set of all possible daughter points to be its *potential daughter cluster* ("cluster" for short). The size and shape of this cluster depends on the interferer's constellation. Thus a maximum-likelihood demodulator interested only in the far-packet infers the most probable parent point of the observed (noisy) daughter point by identifying the most probable cluster to which an observation belongs. Identifying successively less probable clusters helps the demodulator refine its reliability estimate of each detected code bit of F. Analogous to the single-user case, the reliability of the k^{th} bit in the n^{th} symbol can be approximated using the max-log-MAP approximation:

$$L_n^{(k)} \approx N_0 \min_{s \in \sqrt{1-\alpha}\mathcal{S}_{\mathsf{x}_{\mathrm{F}}}^{(k-)} \times \sqrt{\alpha}\mathcal{S}_{\mathsf{x}_{\mathrm{N}}}} |Y_{\mathrm{F}}(n) - h_{\mathrm{F}}s|^2 - \min_{s \in \sqrt{1-\alpha}\mathcal{S}_{\mathsf{x}_{\mathrm{F}}}^{(k+)} \times \sqrt{\alpha}\mathcal{S}_{\mathsf{x}_{\mathrm{F}}}} |Y_{\mathrm{F}}(n) - h_{\mathrm{F}}s|^2 \tag{10}$$

[9]To any desired confidence interval.

[10]In practice sending this information entails a small overhead, which we neglect in this report.

where $\mathcal{S}_{\mathsf{x}_\mathrm{F}}^{(k-)} \subset \mathcal{S}_{\mathsf{x}_\mathrm{F}}$ and $\mathcal{S}_{\mathsf{x}_\mathrm{F}}^{(k+)} \subset \mathcal{S}_{\mathsf{x}_\mathrm{F}}$ comprise symbols whose k^{th} bits are 0 and 1, respectively.

In the high-reliability regime, the dominant error events in such a demodulator are events where a daughter point is incorrectly identified with a neighboring cluster. Analogously to the point-to-point case (when these "clusters" are just points), the probability of these error events is controlled by the effective cluster separation

$$d_{\text{eff}}^{(\mathrm{F,N})} \equiv d_{\text{eff}}(\alpha, \mathsf{x}_\mathrm{F}, \mathsf{x}_\mathrm{N}) = \min_{\substack{p_1,p_2 \in \sqrt{1-\alpha}\mathsf{x}_\mathrm{F} \\ p_1 \neq p_2}} \; \min_{d_1,d_2 \in \sqrt{\alpha}\mathsf{x}_\mathrm{N}} |p_1 + d_1 - p_2 - d_2|. \tag{11}$$

Here the superscript (F,N) on the left hand side emphasizes that the parent points are drawn from x_F and the interferer is drawn from x_N. Using arguments similar to those in [12], [13], PER_F can be approximated as

$$\mathrm{PER}_\mathrm{F} \approx B_\mathrm{F} w_{\mathsf{c}_\mathrm{F}} Q\left(d_{\text{eff}}^{(\mathrm{F,N})} \sqrt{\frac{d_{\mathsf{c}_\mathrm{F}} \gamma_\mathrm{F}}{2}} \right) \tag{12}$$

in the high-reliability regime, where $Q(x)$, $x > 0$ is the Q-function. Assuming perfect cancellation at N,

$$\mathrm{PER}_\mathrm{N} \quad \approx B_\mathrm{N} w_{\mathsf{c}_\mathrm{N}} Q\left(d_{\mathsf{x}_\mathrm{N}} \sqrt{\frac{d_{\mathsf{c}_\mathrm{N}} \gamma_\mathrm{N}}{2}} \right), \tag{13}$$

which can be plugged into (13) to approximate PER_N.

However, for PERs of practical interest (say PER $\lessapprox 0.1$) these Q-function bounds are too loose to predict the correct solutions to (6). In this regime we use these bounds as estimates that reduce the search space of the achievable far-rates, and refine them further via simulation. In the following subsection we illustrate these ideas by means of a design example. The code library chosen in the example is the same as that used in our system design.

C. A Design Example

We use a BICM code library with a decoder structure explained in Section III-B2. These BICM codes were implemented in our testbed in a point-to-point setting.

The library consists of all possible pairings of 4 convolutional codes $\{\mathsf{c}^{(1)}, \mathsf{c}^{(2)}, \mathsf{c}^{(3)}, \mathsf{c}^{(4)}\}$ with three constellation mappers. There is no interleaving[11]. Table I summarizes the details of these convolutional codes. Note that $\mathsf{c}^{(2)}, \mathsf{c}^{(3)}$ and $\mathsf{c}^{(4)}$ are obtained by appropriate puncturing of $\mathsf{c}^{(1)}$, which is the standard rate-1/2 constraint length 7 convolutional code with the generator matrix $[133, 171]$. The constellations are $\mathsf{x} \in \{\mathsf{BPSK}, \mathsf{QPSK}, \mathsf{16QAM}\}$. These are all QAM constellation with even b_x, so $d_\mathsf{x} = \sqrt{6/(2^{b_\mathsf{x}} - 1)}$ [10]. With this code library the available set of spectral efficiencies is the sequence $(r_1, r_2, \ldots, r_{12})$ obtained by ordering the elements of the set $\{1, 2, 4\} \times \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}\} = \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}, 1, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}, 2, \frac{8}{3}, 3, \frac{10}{3}\}$ in ascending order.

We now solve (6) via Monte Carlo simulation for $\epsilon = 0.1$ and $L = 1536$[12] with perfect receiver CSI and perfect interference cancellation. Besides illustrating the design procedure for a concrete example, the far-rates from this

[11]Indeed, code performance can be optimized by suitably tailoring the interleaver Π. However its absence does not change the main message of this example (or indeed, that of the report), which is to show substantial gains from SC even for finite blocklengths and constellations.

[12]For our implementation this choice of L strikes a balance between code performance and implementation constraints.

Code	Rate ρ_{c}	Free Distance d_{c}	#Free Dist. Error Events w_{c}
$\mathsf{c}^{(1)}$	1/2	10	11
$\mathsf{c}^{(2)}$	2/3	6	1
$\mathsf{c}^{(3)}$	3/4	5	8
$\mathsf{c}^{(4)}$	5/6	4	14

Table I
KEY PARAMETERS OF THE CONVOLUTIONAL CODES IN THE CODE LIBRARY.

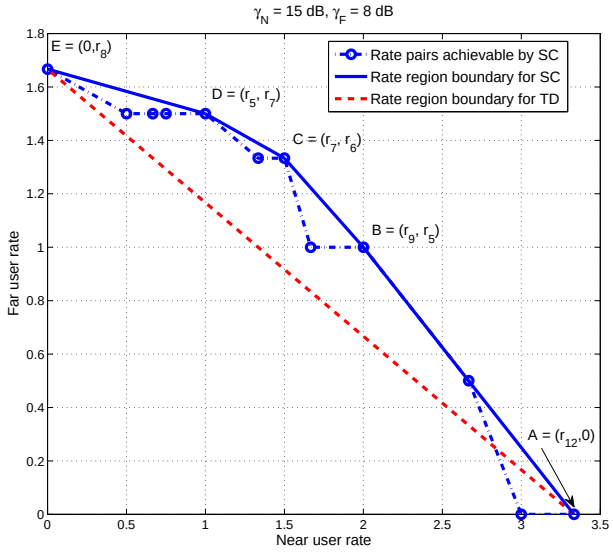

Figure 3. Optimal rate pairs (solution to (6)) for $\gamma_{\mathrm{N}} = 15$ dB and $\gamma_{\mathrm{F}} = 8$ dB for the library of $M = 12$ codes described in Section III-C. The TD rates are obtained by time-sharing between the single-user operating points. The values of α at corner points A-E are $1, 0.2138, 0.1259, 0.0631, 0$ respectively.

simulation provide upper bounds for a practical system (where neither of these conditions holds). Figure 3 shows the simulated rate region for $\gamma_{\mathrm{N}} = 15$ dB and $\gamma_{\mathrm{F}} = 8$ dB.

IV. IMPLEMENTING A SUPERPOSITION-CODED SYSTEM

A. The Platform

We implemented all physical layer processing steps at BS, N and F by suitably modifying an existing point-to-point wireless testbed. The testbed uses Orthogonal Frequency Division Multiplexing (OFDM), and its design parameters are similar to those in the IEEE 802.11a standard. The testbed runs on GNU Radio (revision 10923) on a Linux PC. GNURadio provides driver functions that interface the PC with the USRP board that functions as the analog frontend and the RF.

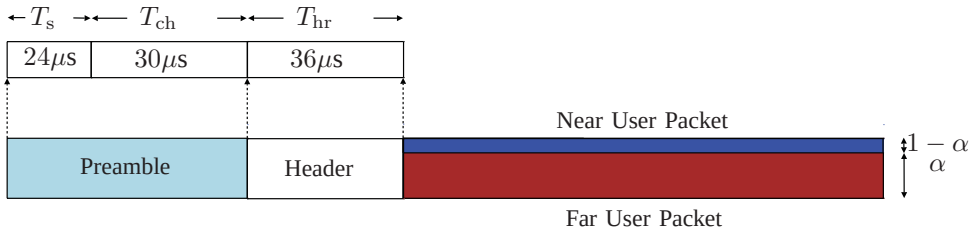

Figure 4. Packet structure.

Center Frequency	900 MHz
Message Bandwidth	2 MHz
Modulation	16-tone OFDM (8 data, 4 pilot, 4 null)
CP Length	1μs

Table II
PARAMETERS USED IN THE EXPERIMENT.

B. Packet Structure

Transmissions occur in *frames*. As in WiFi, each frame consists of a preamble followed by a header and a payload. The preamble assists the receiver in frame acquisition and channel estimation. The header encodes the BICM code type and α. The preamble and the header designs are left unchanged from the single-user case [13]. The only "multiuser" section in the frame is the payload.

The OFDM packet structure used is depicted in Figure 4. The preamble sequence is used primarily for frequency and timing synchronization, and is designed by repeating a pseudo-random training sequence of length 8 symbols six times. In the experiment, the bandwidth is 2 MHz. Hence the duration of the preamble $T_{\rm s} = 6 \times 8/2 \times 10^6 = 24$ μs. The channel estimation symbols (which are the second part of the preamble) are used for performing equalization and is generated by repeating a pseudo-random sequence of length 16 symbols thrice. The header conveys the information about code type and power allocation factor α. The data portion of the packet is followed by the header. The far-user power is α and near-user power is $1-\alpha$. The length of the cyclic prefix was chosen to be 4 symbols (2 μs).

The block of $L = 1536$ coded symbols is transmitted over 192 OFDM symbols with 8 subcarriers for the payload in each OFDM symbol. We retain the 4 pilot subcarriers used by the testbed for frequency and phase tracking and 4 null subcarriers for spectral shaping. This brings the total number of subcarriers to $8+4+4=16$. The message bandwidth of 2 MHz is limited by the USRP, and the cyclic prefix is made commensurate with the (relatively) flat frequency response of the channel for this bandwidth. We summarize the system parameters in Table II.

[13]Consequently, known techniques (e.g., the use of windowed correlators as in [15]) and low-rate header encoding (as in IEEE 802.11a) can be used.

C. Single-User Characterization

Creating experimental conditions that ensure a time-invariant wireless propagation loss can be complicated: although the indoor radio channel and the USRP boards have a reasonably flat response over a 2 MHz bandwidth, the propagation loss is quite sensitive to changes in the environment (e.g., those caused by motion). One approach would be to compensate for such changes by appropriate power control at the BS. Doing so would require SNR feedback from the users on a control channel; designing such feedback links would be worth the effort only if SC could provide substantial rate-gains in a *perfectly power-controlled* environment. We focus on the latter question in the report.

While manual power adjustments work at smaller time scales lasting a few minutes (see our recent submission [17]), they are quite cumbersome for longer duration experiments (e.g., measuring the PER for all the rates in the library, or solving (6),see Section VIII-A).

To circumvent this problem, we emulate perfect power control by connecting the Tx and Rx with a coaxial cable (see Figure 5), resulting in controllable experimental conditions and reproducible outcomes [14]. With this coaxial setup, we measure the point-to-point PER as a function of the transmit power P (which implicitly determines the SNR γ) for each rate index in the code library using parameters from Table II.

Starting from a value of P chosen such that the received power (at the antenna port, as measured by a spectrum analyzer) is at a fixed level above the theoretically predicted thermal noise floor, we change the transmit power in 1 dB steps and measure the PER for each rate index [15]. For comparison we overlay the PER plots for a simulated Gaussian system with the same theoretical noise floor but without additional sources of noise and distortion in Figure 6. Measurement data is presented in Table III. For a rate-index i, let $P_{\min}(i;\epsilon) \equiv P_{\min}(i)$ as the smallest experimentally obtained power level that makes a link $\epsilon-$feasible for rate index i. Denote the corresponding power level by $\tilde{P}_{\min}(i)$ on the simulated Gaussian link.Observe that the slopes of the ideal and experimental waterfall curves are similar upto $\gtrapprox 1\%$ PER. At $\lessapprox 10\%$ PER, the combined non-ideality of the hardware and implementaion result in a maximum power loss of $10\log\frac{P_{\min}(i)}{\tilde{P}_{\min}(i)} \leq 3.5\,\text{dB},\ i \in [12]$ $(= 3 \times 4)$ from the ideal results. Thus we take this "coaxial channel" to be a reasonable approximation of a Gaussian channel in our experiments.Using these results we obtain the smallest feasible near-fractions $\beta_{\text{N}}(i)$ for N as

$$\beta_{\text{N}}(i) = \frac{P_{\min}(i)}{P_{\min}(12)}, \qquad i \in [12]. \tag{14}$$

We use $(\beta_{\text{N}}(i))$ as starting points to find $\alpha(i)$ in the rate region experiment in Section VIII-C.

V. Modifications to Implement an SC-BICM System

The SC encoder f is implemented as two instances of a point-to-point BICM (sub-)encoder and a combiner. The encoded data is then transmitted via a standard OFDM modulator. The receiver is implemented as a single

[14]Indeed, the same practical considerations motivate the use of channel emulators.

[15]This mimics a system with a 1 dB granularity in power control. Also note that the effective signal distortion seen at the receiver is usually higher than that due to thermal noise alone, due to additional sources of noise and distortion such as imperfect receive implementation and hardware imperfections. In our experiments an initial power level of 20 dB above the theoretical value was found to be adequate.

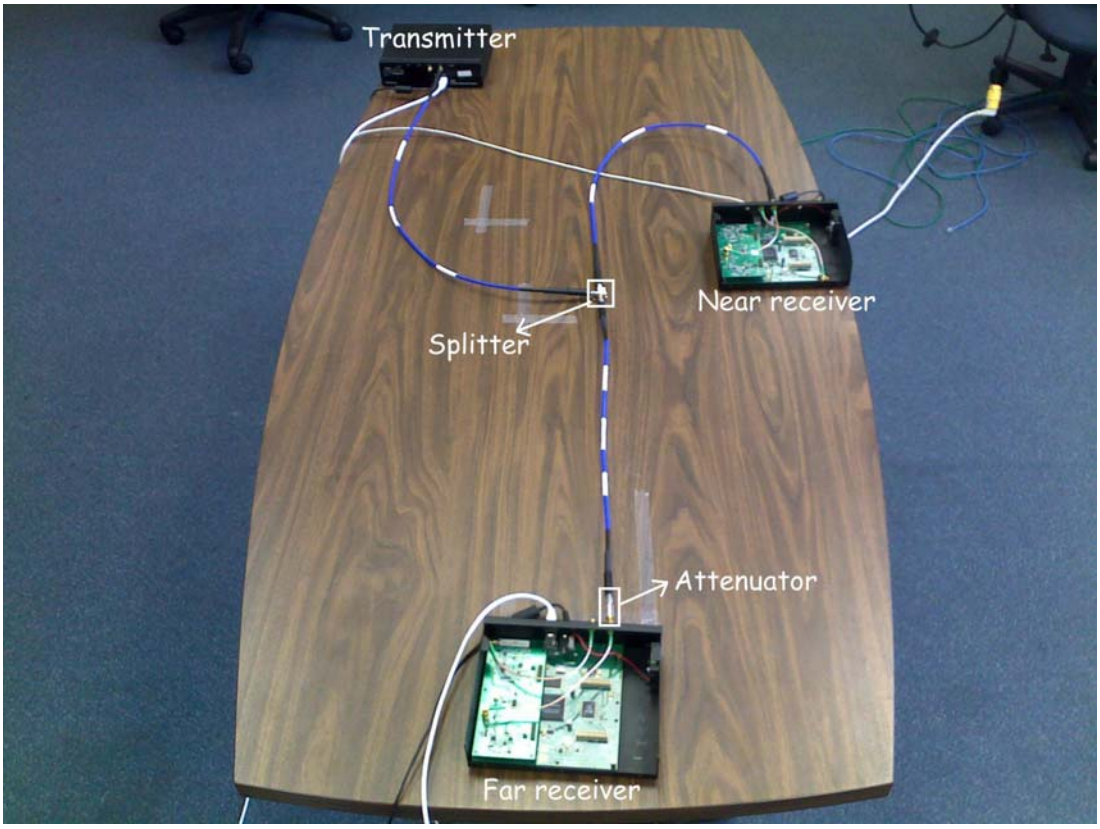

Figure 5. The setup used to approximate a Gaussian BC. The USRP boards are connected via cables. A splitter is used to split the transmitted signal to the two receivers, while an attenuator is used to (virtually) create the presence of a far user.

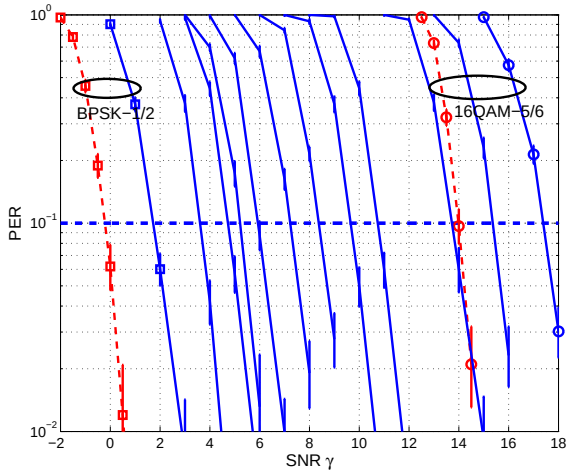

Figure 6. Single-user PER versus SNR, along with the 95%confidence intervals for all the 12 rates in the library. The solid lines depict the experimental results; the ideal curves for BPSK-1/2 and 16QAM-5/6 are also shown (dashed lines). Observe that at 10% PER the implementation loss is $2 - 3.5$dB.

	SNR	PER(%)		SNR	PER(%)		SNR	PER(%)
BPSK 1/2	-1	90.2	QPSK 1/2	2	100	16QAM 1/2	8	98.4
	0	37.2		3	69.8		9	44.4
	1	6		4	17.3		10	6
	2	0.8		5	1.3		11	0.5
BPSK 2/3	1	93.8	QPSK 2/3	4	100	16QAM 2/3	11	94.8
	2	37.7		5	66.9		12	38
	3	4.2		6	16.3		13	6
	4	0.2		7	1.9		14	0.9
BPSK 3/4	2	96.9	QPSK 3/4	5	100	16QAM 3/4	12	99.8
	3	44.2		6	84.3		13	73.3
	4	5.7		7	21.5		14	23
	5	0.5		8	2.8		15	2.3
BPSK 5/6	3	99	QPSK 5/6	7	93.1	16QAM 5/6	14	97.4
	4	61.9		8	37.3		15	57.4
	5	8.7		9	5		16	21.4
	6	1		10	0.4		17	3

Table III
SINGLE USER PER VERSUS SNR. DATA TABLE OF FIGURE 6.

GNU Radio signal processing block, and contains a successive decoding block to decode the near user packet. A point-to-point transceiver is a special case of the SC system we design here.

A. Transmitter Operation

The transmitter block diagram is shown in Figure 7. The payloads (near- and far-user data in bits) are provided to the physical layer by the link layer. Each user's payload is assigned a modulation type, code rate, and the fraction of the transmit power. As shown in Figure 7, these payloads are separately encoded and modulated based on their respective parameters prior to time-domain conversion. At this stage, the modulator is identical to two single-user transmitter. In the time domain, the signals are weighted by the fraction of power allocation factors ($\sqrt{1-\alpha}$ for the near-user signal and $\sqrt{\alpha}$ for the far-user signal), and added. The pilot stream is inserted for channel estimation at receiver, and the signal is processed by IFFT routine. The cyclic prefix is then inserted in each OFDM symbol. After the preamble and channel estimation sequences are added, the signal is upconverted for transmission. Due to the non-idealities of the USRP1, software interpolation is implemented in order to get a flatter frequency response of signals. The link layer interacts with the physical layer through Unix sockets, while the physical layer is implemented using GNU Radio signal processing blocks, which are introduced in Section VI. The transmitter interacts with the USRP via GNU Radio driver software.

B. Receiver Operation

The receiver block diagram is shown in Figure 8. The receiver front end is implemented in hardware using the USRP, while the physical layer is implemented using a single GNU Radio signal processing block. The USRP

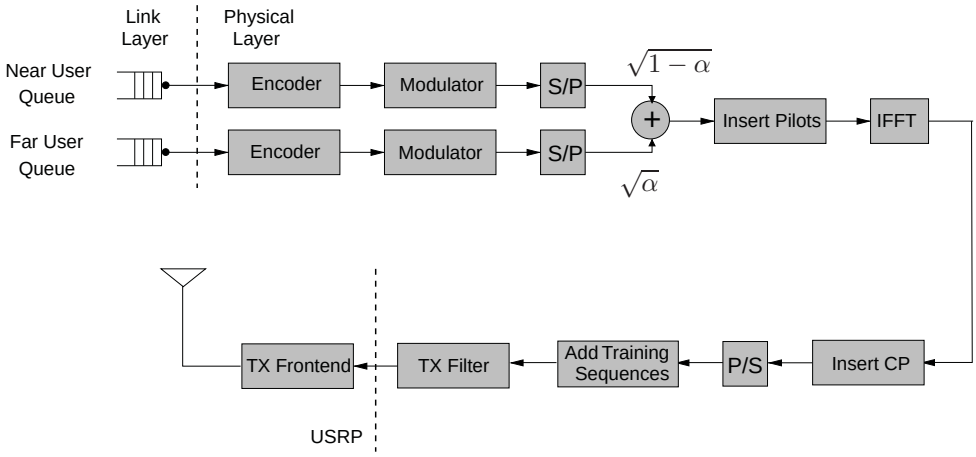

Figure 7. Transmitter block diagram.

downconverts and digitizes signals in the frequency band of operation. Initially, all blocks in the receiver except the timing and frequency synchronization block are inactive. This block essentially consists of a windowed correlator. Our algorithm is similar to the well-known Schmidl-Cox algorithm [15]. If the signal in the correlator window satisfies the correlation properties, this block declares a valid packet along with the location of the channel estimation training sequence (see Figure 8). Moreover, our preamble structure allows us to use the phase of the correlator to estimate the frequency offset with respect to the transmitter. Since this estimate can have a large variance, we call this the Coarse Frequency Estimate. The coarse frequency correction unit uses this coarse estimate supplied by the synchronization unit to correct all future samples in the packet for frequency offset. Using the location of the channel estimation training sequence, the coarse-offset corrected samples are now fed to the channel estimator, which estimates the channel. In this implementation we have assumed that the receiver knows the modulation, coding and power allocation information. Hence after the channel estimation is complete, the receiver is ready to decode the payload(s).

Since the locations of the OFDM symbols are known in time, the cyclic prefix (CP) is removed from the incoming samples, and blocks of 16 (equal to the number of subcarriers) samples are buffered and processed by the FFT routine. Once in the frequency domain, the pilot subcarriers are identified, and using the channel estimates obtained earlier, the pilot tones are used to estimate and correct the payload symbols for any residual (i.e., “fine”) frequency offset. These offset-corrected symbols are equalized using the channel estimates to calculate log likelihood ratio (LLR) values for channel decoding. Until this point, the receiver operation is quite similar to a typical IEEE 802.11a/g receiver.

The far user decodes its payload bits by simply treating the near user signal as interference. Far user receiver structure is identical to a single-user OFDM receiver. In contrast, the near user uses the successive decoding procedure to decode its payload bits (see Figure 8). This procedure involves decoding the far user bits, re-encoding and modulating the payload to reconstruct the far user component for each subcarrier of the received signal. This is

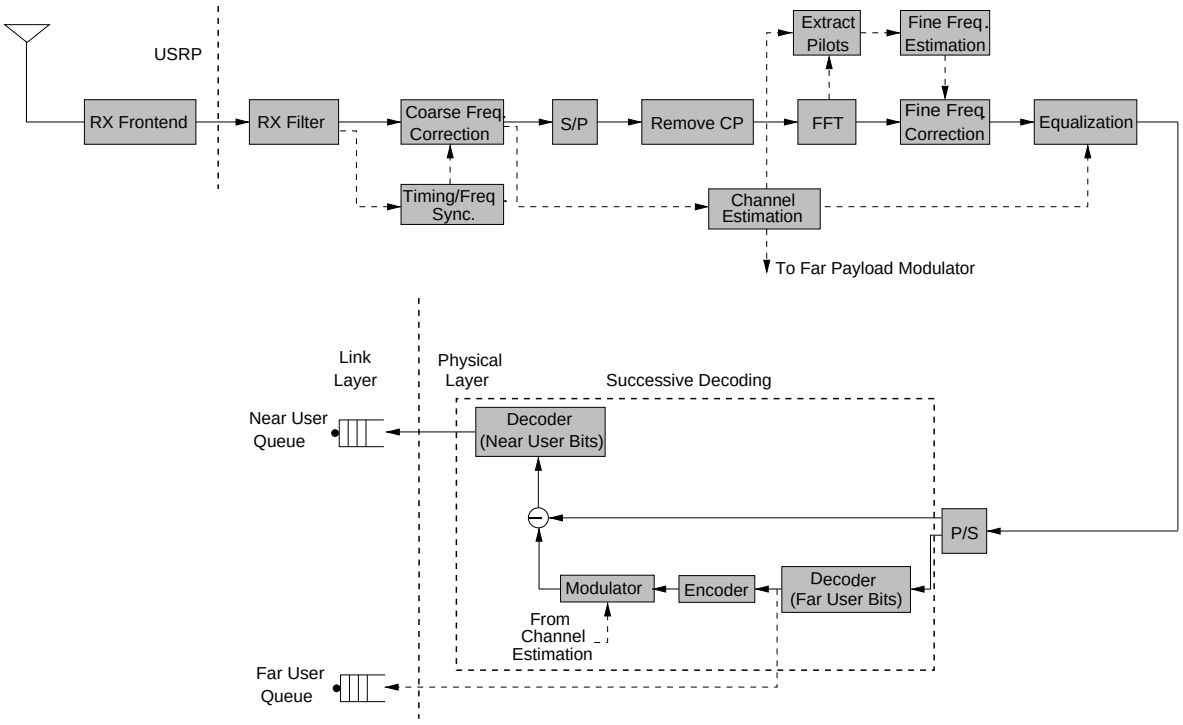

Figure 8. Receiver block diagram.

subtracted subcarrier-wise from the received signal; its payload is then extracted from the residual signal. In both cases, the decoded packets are sent to their respective queues. The link layer checks the CRC for each received packet and declares an error if the CRC fails.

VI. GNU Radio Block Diagrams

A. SC Transmitter Blocks

The GNU Radio block diagrams of the SC transmitter are shown in Figure 9 (part I) and 10 (part II), respectively[16]. The block names starting with "spc" are the signal processing blocks specific to SC implementation. Other blocks are standard GNU Radio signal processing blocks, whose descriptions can be found in [21]. As we see from the two figures, two packets (far and near packets) are modulated and encoded separately as two single-user systems. The two packets are then weighted according to the power allocation and added. The superimposed packet is next passed though an OFDM modulator. After the preamble, the channel estimation tones, and the header are inserted, the packet is pushed to USRP for transmission. The header path, which is the green path in Figure 9, conveys the power allocation information for both users. The information is first modulated and encoded, and then inserted into the packet as the header. In this section, we introduce the modified SC blocks in details. These blocks are designed specifically to SC implementation.

[16]Enlarged figures are available at http://www.nd.edu/~mhaenggi/pubs/spc_transmitter_part1.pdf and http://www.nd.edu/~mhaenggi/pubs/spc_transmitter_part2.pdf.

- **spc_packet_source**

Type	Function
Description	This block generates a packet for either far or near user. The block generates a new packet when the number of packets in queue is less than **msgq_length**.
Parameters	**msgq_length**: maximum number of packets in queue
I/O	**data_out**: output packet **control_out**: output control flow

- **spc_header_source**

Type	Function
Description	This block generates the header of a packet. The header includes the power allocation information for far and near users. This block also generates two vectors of far and near power output. (The standard GNU Radio block **gr_multiply_cc** only accepts two vectors with the same length. Hence, the vector length of far or near power output is equal to the length of data stream.)
Parameters	**msgq_length**: maximum number of packets in queue
I/O	**header_out**: power allocation information for far and near users **far_power_out**: vector of far power output **near_power_out**: vector of near power output

- **spc_usage_map_xx**

Type	Function
Description	This block maps the data and pilot streams according to the usage map stream. In our OFDM design, not all sub-carriers are used for data transmission. Some are reserved for pilot stream and some keep unused. This block maps the data stream into OFDM symbols according to a preset sub-carrier usage pattern. **spc_usage_map_cc**: SC usage map with complex input and complex output.
Parameters	**umap**: sub-carrier usage pattern **packet_start_sym**: symbol that indicates the start of a packet **unused_symbol**: symbol mapped to unused sub-carriers **pilot_sequence**: sequence of pseudo-random pilot stream
I/O	**data_in**: input data **control_in**: input control flow **data_out**: output data **control_out**: output control flow

- **spc_pad_packet_xxx**

Type	Function
Description	This block pads zeros to the end of a packet so that the total length of a packet is a multiple of the number of used sub-carriers in a single OFDM symbol. **spc_pad_packet_ccc**: pads packet with complex input, complex output, and complex pads.
Parameters	**pad_to_multiple_of_how_many_samples**: total number of samples in a packet after padding **packet_start_sym:** symbol that indicates the start of a packet **packet_stop_sym:** symbol that indicates the stop of a packet **padvalues:** the value that is padded to a packet **control_pad_values:** control flow of pad values
I/O	**data_in**: input data **control_in**: input control flow **data_out**: output data **control_out**: output control flow

- **spc_insert_into_stream_xxx**

Type	Function
Description	Insert a stream into a data flow. **spc_insert_into_stream_ccc**: inserts a stream into a data flow with complex input, complex output, and complex pads. **spc_insert_into_stream_bbb**: inserts a stream into a data flow with binary input, binary output, and binary pads. **spc_insert_into_stream_ccb**: inserts a stream into a data flow with complex input, complex output, and binary pads.
Parameters	**insert_what**: a stream to be inserted into a data flow **control_sym_where:** control symbol of the stream **how_to_insert:** the way to insert the stream: either before a data flow or after a data flow
I/O	**data_in**: input data **control_in**: input control flow **data_out**: output data

- **spc_packet_accumulator_xx**

Type	Function
Description	Buffers generated packets. **spc_insert_into_stream_cc**: buffers generated packet with complex input and complex output.
Parameters	**buffering_start_sym**: control symbol indicating the start of buffer **buffering_end_sym**: control symbol indicating the end of buffer
I/O	**data_in**: input data **control_in**: input control flow **data_out**: output data **control_out**: output control flow

B. SC Receiver Block

The receiver design, which differs from the transmitter design, only includes a single signal processing block ("**spc_receiver_fsm**") which contains all the functionalities shown in Figure 5. Such design benefits high efficiency in the experiment, since the receiver's computation complexity is much higher than the transmitter's complexity.

- **spc_receiver_fsm**

Type	Function
Description	Demodulates and decodes a received packet. If the far user decodes a packet, it only decodes its own information; if the near user decodes a packet, it first decodes the far-user's information and then implements SD to decode its own information.
Parameters	**chanesttones:** channel estimation sequence **noise_power_offset:** the gap between the end of a packet and the start of noise estimation samples **noise_estimation_length:** the length of samples used for noise estimation **pilot_sequence:** pilot streams **preambletones:** preamble sequence **umap:** usage map sequence
I/O	**data_in**: input data

VII. Hardware Characterization

Before we proceed to the actual experiment, it is necessary to experimentally characterize the RF front-end we use. In this section, we present the transmit power and noise figure measurement of USRP boards. The SNR estimation of a received packet is also presented.

A. Transmit Power Characterization

In GNU Radio and USRP, the transmit power is controlled by the amplitude gain setting at the transmitter. The larger the amplitude gain has, the higher the transmit power is. To measure the transmit power quantitively, we only transmit the data portion of a packet instead of the entire packet [17].

Figure 11 shows the averaged transmit power on each subcarrier as a function of the amplitude gain [18]. We observe that the USRP transmitter saturates when the amplitude multiplier is set larger than 3000. Figure 12 shows the frequency response of OFDM symbols with the amplitude multiplier 221. By implementing the software interpolater (see Section V-A), the frequency response of the transmitted signal is flat.

[17]The term "entire" represents a packet with preamble, channel estimation sequence, and pilot stream as presented in SectionV.

[18]The relationship between amplitude gain and output power varies from different daughterboards. The results refer to FLEX900 daughterboards.

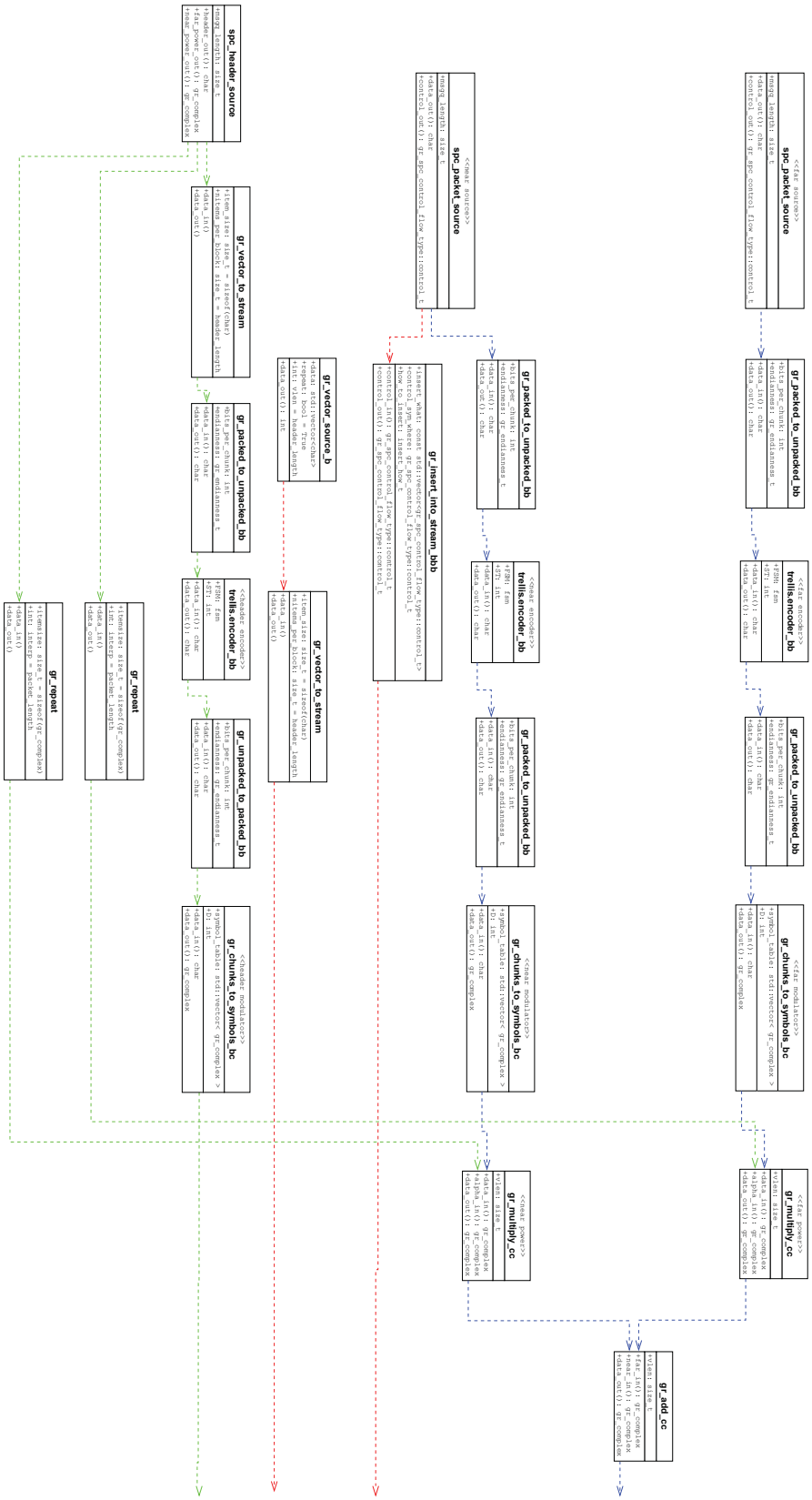

Figure 9. GNU Radio block diagrams of SC transmitter (Part I).

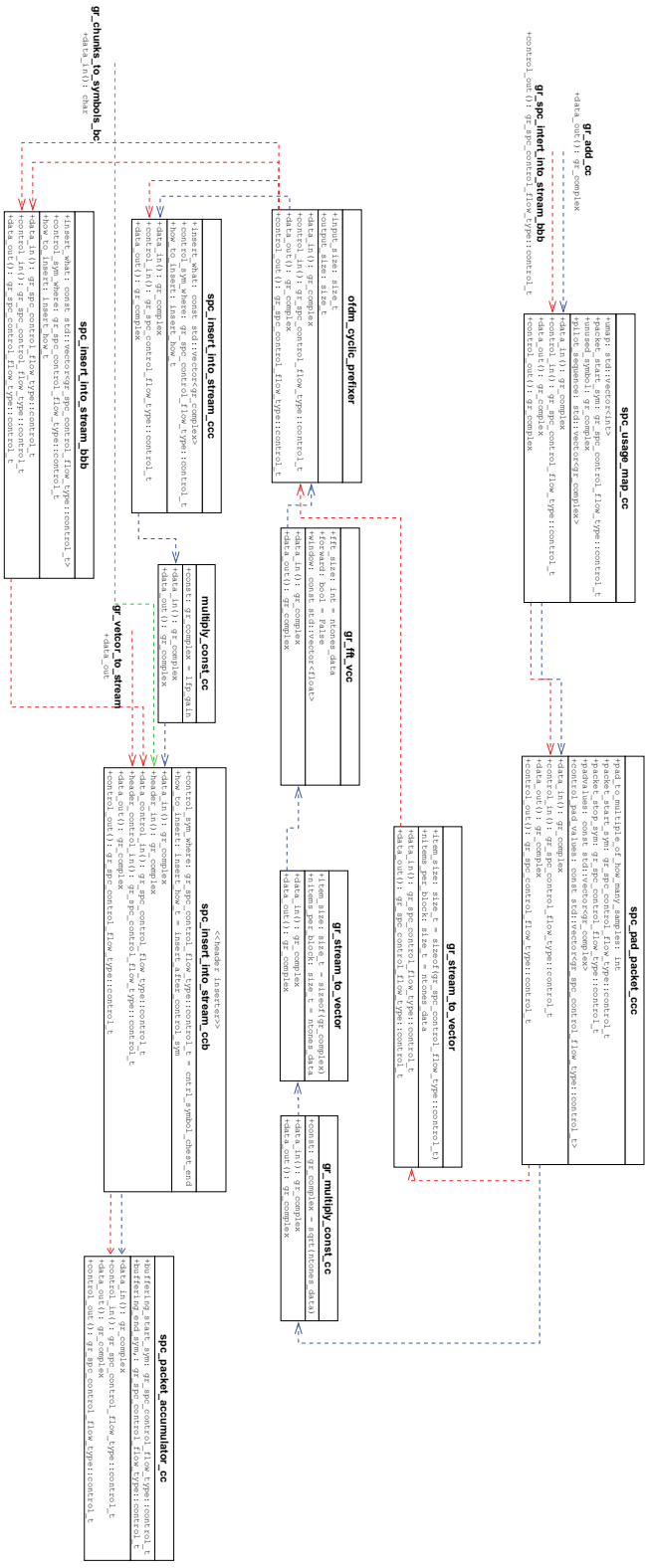

Figure 10. GNU Radio block diagrams of SC transmitter (Part II).

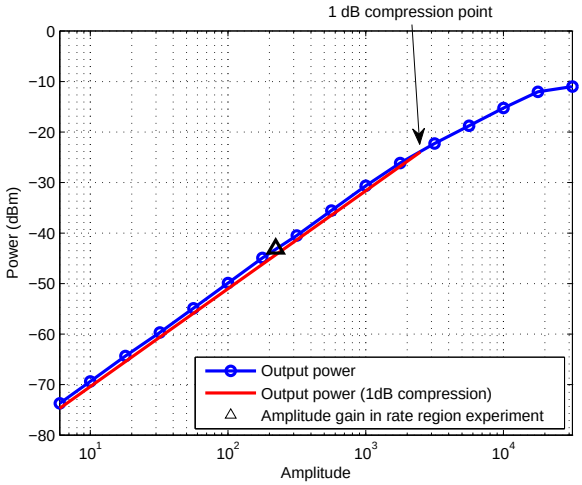

Figure 11. Transmit power per subcarrier measured in dBm versus amplitude multiplier for USRP1 with FLEX900 daughterboard. The far power and near power in transmitter are set at 1 and 0, respectively. The curve of circles is the measured transmit power per subcarrier. The solid-line curve is the 1 dB compression curve. The triangle is the amplitude gain that we use in rate region experiment (see SectionVIII).

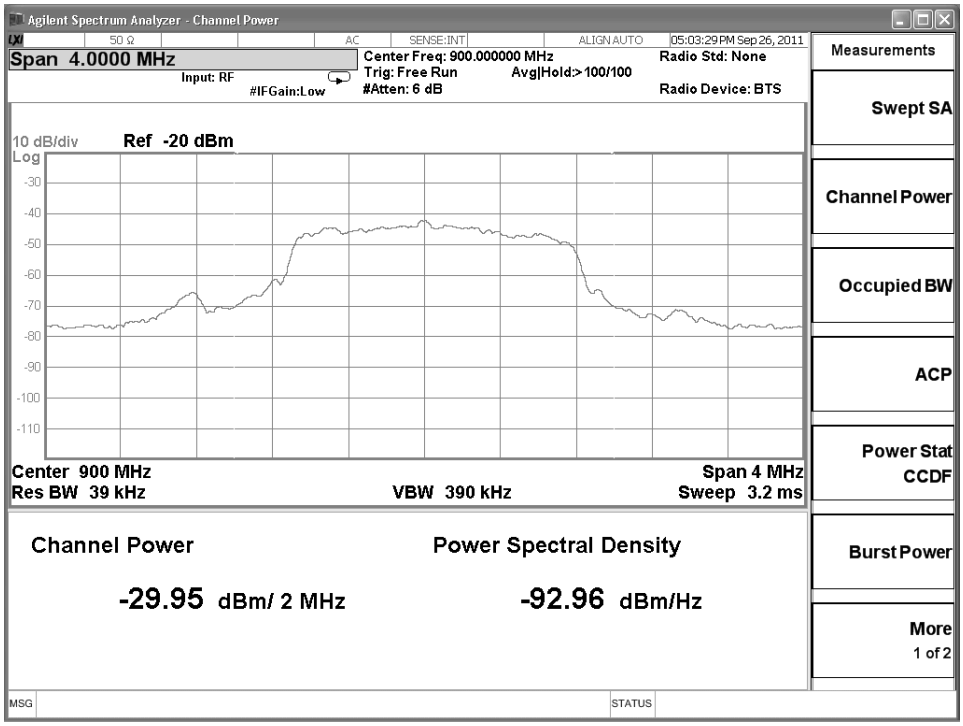

Figure 12. Spectrum of the USRP TX output. The amplitude multiplier is 221. The total transmit power is -29.95 dBm in a 2 MHz bandwidth range.

B. Noise Figure Measurement

Noise figure measures the degradation of SNR at receiver due to the noise from receiver amplifiers. For example, Figure 13 shows the spectrum of an FM signal from Agilent N9310A RF Signal Generator (N9310A). The snapshot is captured by Agilent N9010A EXA Signal Analyzer (N9010A). The power of the FM signal is -50 dBm, and the frequency deviation is 100 kHz. The thermal noise density at the output of N9310A is

$$N_0 = kT \approx -173.8\,\mathrm{dBm/Hz},$$

where $k = 1.374 \cdot 10^{-23}$ is the Boltzmann's constant; $T = 300$ K is room temperature. The resolution bandwidth is set at 18 kHz. From Figure 13, we obtain that the noise floor is at -112.3 dBm. Hence, the background noise density at N9010A is $N_o = -112.3 - 10\log_{10}(1.8 \cdot 10^4) = -154.9$ dBm/Hz, where 1.8×10^4 is the resolution bandwidth. The noise factor F is defined as the input to output SNRs:

$$F = \frac{S_i/N_i}{S_o/N_o},$$

where S_i and S_o represent input and output signal powers, respectively; N_i and N_o represent input and output noise powers, respectively. The noise figure is the logarithmic equivalent of the noise factor:

$$N_{\mathrm{fig}} = 10\log_{10} F = -10(\log_{10} kTB - \log_{10} N_o) + 10\log_{10} S_i - 10\log_{10} S_o,$$

where B is signal bandwidth. In this example, we have $S_i = S_o$. The noise figure of N9010A is then

$$N_{\mathrm{fig}} = -(-173.8 + 154.9) = 18.9\,\mathrm{dB}.$$

Similar steps can be taken to measure the noise figure of the USRP1. In USRP analyzer (**usrp_fft.py**), we only obtain relative power levels (shown in dB) in digital domain. However, we can still obtain the output SNR since the signal and noise powers are scaled by the same value. We keep the transmit signal power unchanged and measure the received signal power and noise floor with different RX gains. The RX gain is the amplifier gain setting in the receiver to magnify received signal. The range of RX gain differs with different daughterboards. For FLEX900, the RX gain ranges from 0 to 90. Figure 14 shows the signal power and noise floor as a function of the RX gain. The signal from N9310A is a FM signal with power -50 dBm. All the power values in Figure 14 are relative power (in dB) compared to the noise floor at $\mathrm{rx\_gain} = 0$, where the noise floor at $\mathrm{rx\_gain} = 0$ is equal to $-78.7\,\mathrm{dB}$. From the figure, we observe that when the RX gain ranges from 0 to 25, the signal power increases with the increase of the RX gain while the noise power remains stable. Hence, the noise figure is reduced, which is shown in Figure 15. The input noise power $N_i = -173.8 + 10\log_{10}(2 \times 10^3) = -140.8$ dBm, since the resolution bandwidth of USRP is 2×10^3 Hz. The rest of the calculation is then straightforward. If the RX gain drops in range from 25 to 55, both signal and noise power increase with RX gain. When the RX gain is larger than 55, the receiver reaches a saturation region. The signal is beyond the dynamic range of the receiver ADC. The noise floor cannot be further reduced since it is already at its minimum. Figure 15 shows the noise figure of USRP1 as a function of RX gain.

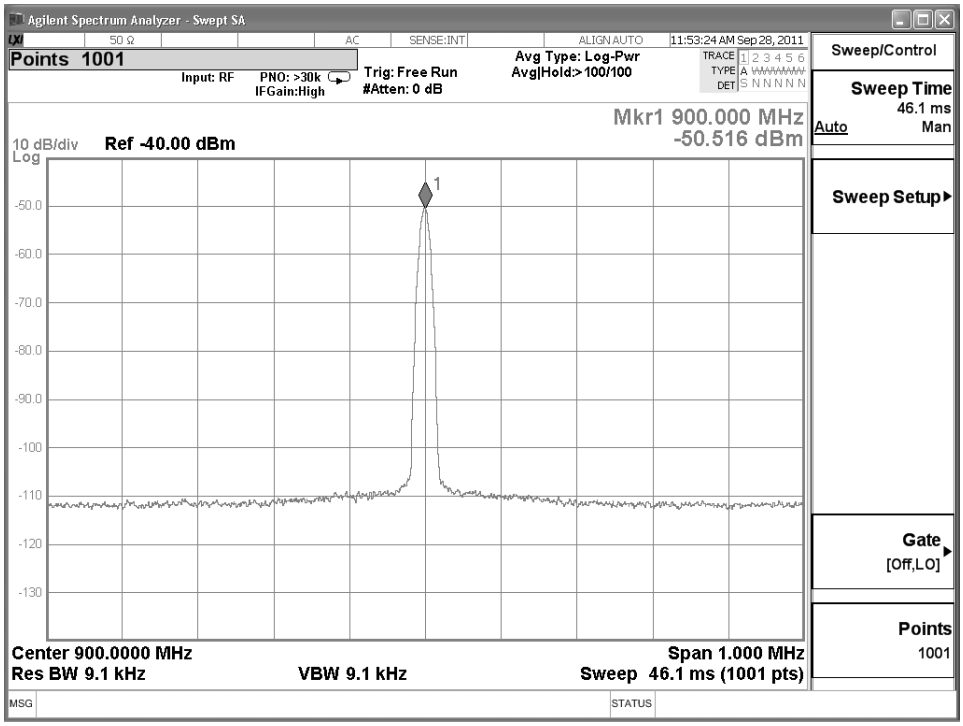

Figure 13. Spectrum of an FM signal with central frequency 900 MHz and frequency deviation 100 kHz.

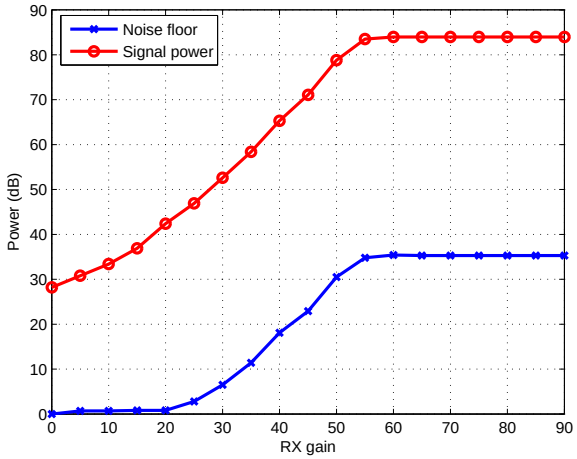

Figure 14. Signal power and noise floor (in dB) as a function of RX gain. Input signal is an FM signal. Data is collected from USRP spectrum analyzer (**usrp_fft.py**). All power values are compared to the noise floor at $\mathrm{rx\_gain} = 0$. The noise floor at $\mathrm{rx\_gain} = 0$ is -78.7 dB, which is the relatvie power value in digital domain.

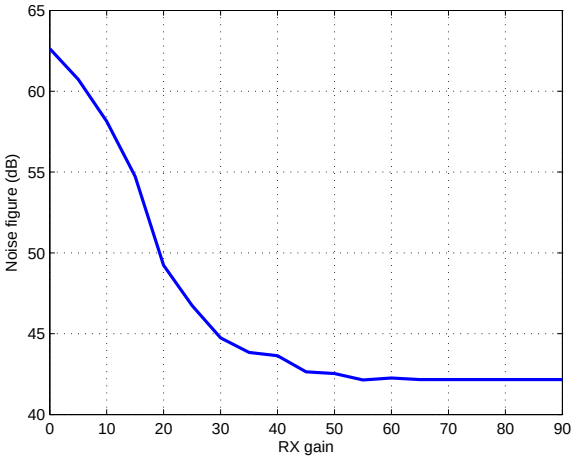

Figure 15. Noise figure versus the RX gain of USRP1 with FLEX900 daughterboard.

C. SNR Estimation

We use a cable to connect the USRP transmitter and receiver to estimate the SNR of a received packet. A typical SNR loss of 0.5 dB in cable is considered. The RX gain is set to 0. As indicated in Section V-B, the receiver first operates the timing and frequency recovery. It calculates the auto- and cross-correlation of a received signal chunk. When a packet is detected, we label the index of the first sample of the packet as **sync_index** (see Section V-B in details). It calculates the cross-correlation of the signal chunk starting from the **sync_index** as

$$R_c = \frac{1}{5}\sum_{k=0}^{4}\sum_{m=0}^{7} x^*(8k+m)x(8k+8+m),$$

where $x(i)$ is the signal sample at time index i. $x(0)$ is the sample at **sync_index**. The estimated SNR of a received packet is:

$$\hat{\gamma} = \frac{|R_c|}{P_N},$$

where

$$P_n = \frac{\sum_{i=1}^{n}|x(i)|^2}{n},$$

where n is the number of noise samples. The noise samples are collected at the end of each packet. In our experiment, the maximum transmission rate we need to support is 16QAM 5/6. (For a PER rate under than 10%, we say that the rate is supported.) From Figure 6, we obtain that the SNR is at least 18 dB in single-user system to cover the entire range of rates (from BPSK 1/2 to 16QAM 5/6).

Next, we determine the transmitter multiplier gain. From Section VII-B, we obtain the the noise power per subcarrier is

$$P_n = N_0 + 10\log\left(\frac{2\times 10^6}{16}\right) + 62.1 = -60.73\,\mathrm{dBm},$$

where $N_0 = -173.8$ is the thermal noise density; $2 \times 10^6/16$ is the bandwidth per subcarrier; 62.2 is the noise figure obtained from Figure 15 with RX gain at 0. The desired transmitter power is the noise power plus SNR (considering 0.5 SNR loss in cable).

$$P_t = -60.73 + 18 + 0.5 = -42.03\,\mathrm{dBm}, \tag{15}$$

which corresponds to a transmitter multiplier of 221 (see Figure 11). For our experiments, we only need to use multiplier values of 221 or lower, which means we keep the transmitter operating in the linear region. Software calculation verifies the result.

VIII. Experimental Results

A. Emulating a Gaussian BC

The time variation of the propagation loss is a bigger problem for the BC because of the presence of *two* Tx-Rx paths in the BC as opposed to just one in the point-to-point setting of Section IV-C. Moreover, checking the feasibility of candidate rate-pairs while solving (6) requires repetitive PER measurements. Thus we emulate a Gaussian BC using a combination of coaxial cables, a splitter, and an attenuator bank, as shown in Figure 5. The USRPs shown communicate with three Linux PCs via USB 2.0. The PCs are configured to run the appropriate Tx or Rx code in GNURadio.

The next step is to fix the single-user SNRs $\gamma_{\mathrm{N}}, \gamma_{\mathrm{F}}$. In our experiment we fix them implicitly by choosing a transmit power P_{N} and an attenuator setting a_{F}. In the single-user mode, P_{N} is chosen to be $P_{\min}(12)$ (plus some additional loss due to the splitter) using the single user results from Section IV-C. The measured γ_{N} was found to be ≈ 18 dB. With this value of P_{N}, we set F in the single-user mode to operate at a desired rate (e.g., QPSK, rate-5/6), and increase the attenuation in $\eta = 1$ dB steps until the far-link violates the PER constraint. γ_{F} is the receive SNR for the largest attenuation setting that supports a 10% (or lower) PER at F (for QPSK rate-5/6, this was found to be $\approx 10\,$dB). With this approximation to the Gaussian BC, we are now ready to obtain the rate region for this system by solving (6) experimentally.

B. The Runtime Requirement

The GNU Radio runs on a general purpose computer. In our experiment, we use the desktop with two quad-core Intel Xeon CPU E5520@2.27 GHz with 8 GB RAM. The transmitter code occupies 20% of one CPU core on average and RAM of 32 MB. It takes approximately 2.1 ms for the transmitter to generate a single packet. On the other hand, the receiver code occupies approximately 96% of one CPU core on average and RAM of 500 MB (No parallel computing technique is involved in the experiment). For the far user, the decoding procedure takes approximately 8.7 ms. For the near user, it needs approximately 19.7 ms to decode its packet. The time is meausred by recording the timestamps before and after encoding(decoding) a packet. The near user needs to do the decoding twice and encoding once so that the complexity is more than doubled compared to the far user. In the experiment, the transmitter has an idle gap of 100 ms between two packets. Hence the receiver has enough time to decode.

The source codes we have designed for SC is approximate 2 MB on disk with over 50 files. The total receiver implementation is within several thousand lines.

C. The Rate Region Experiment

Given $P_{\rm N}, a_{\rm F}$ and a code library $\mathcal{C}$ with rate-indices $k \in [M]$, we use the above experimental setup to solve (6) to find $(k, i_{\rm F}^*(k))$ using the following procedure:

1) Initialize:
   a) $\alpha(k) = \beta_{\rm N}(k)$ where $\beta_{\rm N}(k)$ is calculated from (14).
   b) $i_{\rm F}(k) = \arg\max_l \{P_{\min}(l) < (1-\beta_{\rm N}(k))a_{\rm F}P_{\rm N}\}$.
2) Calculate the PER for $(k, i_{\rm F}(k))$ for stream weights $\sqrt{\alpha(k)}$, $\sqrt{1-\alpha(k)}$.
3) If
   a) (feasible at N) AND (feasible at F): set $i_{\rm F}^*(k) = i_{\rm F}(k)$, $k \mapsto k+1$, go to Step 1.
   b) (feasible at N) AND (infeasible at F): $i_{\rm F}(k) \mapsto i_{\rm F}(k) - 1$, go to Step 2.
   c) (infeasible at N) AND (feasible at F): $\alpha(k) \mapsto \alpha(k) \times 10^{\frac{\eta}{10}}$, go to Step 2.
   d) (infeasible at N) AND (infeasible at F): $\alpha(k) \mapsto \alpha(k) \times 10^{\frac{\eta}{10}}$, $i_{\rm F}(k) \mapsto i_{\rm F}(k) - 1$, go to Step 2.

Here η is the step size for the near-fraction (in this report, we set $\eta = 1$ dB).

D. Results

We use the procedure in Section VIII-C to study three interesting problems: (a) How does the measured rate region change with $a_{\rm F}$ (i.e., the far-link is made stronger or weaker)? (b) how much does imperfect interference cancellation at N affect the rate region, and (c) what are good models to account for N's interference at F, and, in particular, how useful is the popular Interference-As-Gaussian-Noise (IAGN) model in predicting its impact. We discuss these problems in the following.

1) Changing the Strength of the Far-Link: To study this problem, we find the rate region for two possible far-link SNRs: $\gamma_{\rm F} = 5$ dB and $\gamma_{\rm F} = 10$ dB, which correspond to single user rate indices $K = 3$ and $K = 8$ respectively. The near user SNR for both cases is kept at $\gamma_{\rm N} = 18$ dB. These scenarios are emulated by using suitable attenuator values $a_{\rm F} = 9, 4$ respectively[19]. The results are shown in Figure 16 Here, we used a transmit power $P_{\rm N} = -43$ dBm and step size $\eta = 1$ dB.

Table IV shows the mapping of rate index k versus the transmission rate. We do not consider the BPSK $5/6$ during the experiment since we observe that it has worse performance (PER versus SNR) compared to the QPSK $1/2$, inspite of having a lower spectral efficiency (thus it is a poor code). We now describe a step-by-step procedure on how to obtain the set of achievable rate pairs (under the reliablility constraint) for the following two cases:

1) $\gamma_{\rm N} = 18$ dB and $\gamma_{\rm F} = 10$ dB (see Table V).
2) $\gamma_{\rm N} = 18$ dB and $\gamma_{\rm F} = 5$ dB (see Table VI).

[19]Note that $a_{\rm F}$ is not simply equal to $\gamma_{\rm N} - \gamma_{\rm F}$ owing to the two different boards and cables used, which had to be calibrated separately.

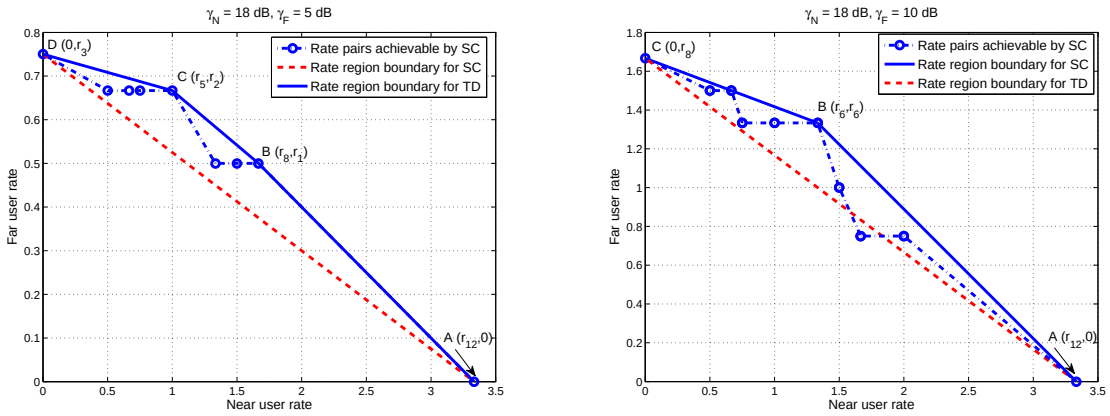

Figure 16. Experimentally obtained rate region for the library of $M = 12$ codes using the setup shown in Figure5 for two different choices of F. (Left) The values of α at corner points A-D are respectively $1, 0.126, 0.079, 0$. (Right) The values of α at corner points A-C are respectively $1, 0.126, 0$.

k	1	2	3	5	6	7
Rate	BPSK 1/2	BPSK 2/3	BPSK 3/4	QPSK 1/2	QPSK 2/3	QPSK 3/4
k	8	9	10	11	12	
Rate	QPSK 5/6	16QAM 1/2	16QAM 2/3	16 QAM 3/4	16 QAM 5/6	

Table IV
TABLE OF k VERSUS THE TRANSMISSION RATE.

We take Table V as an example. The far-user and near-user SNRs are 10 dB and 18 dB, respectively. From the single-user PER curves in Figure 6, the maximum supportable rates for far user and near user are QPSK $5/6$ ($k = 8$) and 16QAM $5/6$ ($k = 12$), respectively. The rate region experiment starts with QPSK 5/6 ($k = 8$) for far user and BPSK 1/2 ($k = 1$) for near user with $\alpha(1) = 0.0251$ (see Step 1 in Section VIII-C). The parameter α determines the SNRs of both users. If the near-user rate is not supported[20] (as the first three αs in the table), we follow Step 3.c in Section VIII-C. α is increased until the rate of both users are supported. For example, $\alpha(1) = 0.0251$ is not sufficient to keep the near-user's PER below 10% due to the imperfect interference cancellation. The increase step of α is chosen so that the SNR of near user is increased by 1 dB. In this case, $\alpha(1) = 0.0398$ is required. If the far-user's rate is not supported (as the case of QPSK $3/4$ ($k = 7$) for far user and BPSK 3/4 ($k = 3$) for near user), we jump to Step 3.b in Section VIII-C. We reduce the far user's rate to QPSK $2/3$ ($k = 6$) and the far-user's rate is supported. The experiment stops when BPSK $1/2$ ($k = 1$) is not supported for the far user. Similar steps can be taken to obtain Table VI.

We clearly see dependence on the choice of $\gamma_{\rm F}$. With $\gamma_{\rm F} = 5$ dB, there is not enough disparity between the near- and far-links to fully benefit from superposition (F is "too close" to BS). This is in fact predicted by theory [1]. On the other hand, we see the effect of a finite code library when $K = 3$ (F is "too far" from BS): since its single-user

[20]We say that the rate is supported if the PER is below 10%.

Far user rate	QPSK 5/6	QPSK 5/6	QPSK 5/6	**QPSK 3/4**	QPSK 3/4	**QPSK 3/4**
Near user rate	BPSK 1/2	BPSK 1/2	BPSK 1/2	**BPSK 1/2**	BPSK 2/3	**BPSK 2/3**
Far user PER			41.1	**1.7**		**3.6**
Near user PER	52.4	15.8	1	**1.4**	58	**9.9**
α	0.0251	0.0316	0.0398	**0.0398**	0.0398	**0.0501**
Under 10% PER?				**Yes**		**Yes**
Far user rate	QPSK 3/4	QPSK 3/4	QPSK 3/4	**QPSK 2/3**	QPSK 2/3	**QPSK 2/3**
Near user rate	BPSK 3/4	BPSK 3/4	BPSK 3/4	**BPSK 3/4**	QPSK 1/2	**QPSK 1/2**
Far user PER			12.1	**1.3**		**1**
Near user PER	67.6	19.1	1.5	**1.5**	25.1	**6.4**
α	0.0501	0.631	0.0794	**0.0794**	0.0631	**0.0794**
Under 10% PER?				**Yes**		**Yes**
Far user rate	QPSK 2/3	**QPSK 2/3**	QPSK 2/3	QPSK 2/3	**QPSK 1/2**	QPSK 1/2
Near user rate	QPSK 2/3	**QPSK 2/3**	QPSK 3/4	QPSK 3/4	**QPSK 3/4**	QPSK 5/6
Far user PER		**7.9**		33.5	**0.5**	
Near user PER	27.3	**2.4**	23	7.5	**3**	58.3
α	0.1	**0.1259**	0.1259	0.1585	**0.1585**	0.1585
Under 10% PER?		**Yes**			**Yes**	
Far user rate	QPSK 1/2	QPSK 1/2	**BPSK 3/4**	BPSK 3/4	**BPSK 1/2**	BPSK 1/2
Near user rate	QPSK 5/6	QPSK 5/6	**QPSK 5/6**	16QAM 1/2	**16QAM1/2**	16QAM 2/3
Far user PER		12.6	**1.1**	51.8	**4.1**	12.1
Near user PER	15.4	1.7	**2.1**		**9.8**	100
α	0.2	0.2512	**0.2512**	0.2	**0.2512**	0.3981
Under 10% PER?			**Yes**		**Yes**	

Table V
RATE REGION DATA SHEET. 18DB FOR NEAR-USER SNR AND 10DB FOR FAR-USER SNR. DATA OF FIGURE 16 (LEFT) .

Far user rate	BPSK 3/4	**BPSK 2/3**	BPSK 2/3	**BPSK 2/3**	BPSK 2/3	**BPSK 2/3**
Near user rate	BPSK 1/2	**BPSK 1/2**	BPSK 2/3	**BPSK 2/3**	BPSK 3/4	**BPSK 3/4**
Far user PER	17.9	**1.9**		**3.2**		**7.5**
Near user PER	5.9	**5.9**	48.3	**5.8**	58.4	**9.4**
α	0.0251	**0.0251**	0.0398	**0.0501**	0.0501	**0.0631**
Under 10% PER?		**Yes**		**Yes**		**Yes**
Far user rate	BPSK 2/3	**BPSK 2/3**	BPSK 2/3	BPSK 2/3	**BPSK 1/2**	BPSK 1/2
Near user rate	QPSK 1/2	**QPSK 1/2**	QPSK 2/3	QPSK 2/3	**QPSK 2/3**	QPSK 3/4
Far user PER		**4.2**		13.8	**0.4**	
Near user PER	25.6	**6.8**	22.8	1.8	**1.6**	24.2
α	0.0631	**0.0794**	0.1	0.1259	**0.1259**	0.1259
Under 10% PER?		**Yes**			**Yes**	
Far user rate	**BPSK 1/2**	BPSK 1/2	**BPSK 1/2**	BPSK 1/2	BPSK 1/2	BPSK 1/2
Near user rate	**QPSK 3/4**	QPSK 5/6	**QPSK 5/6**	16QAM 1/2	16QAM 1/2	16QAM 1/2
Far user PER	**0.6**		**0.7**			26.9
Near user PER	**2.3**	42.5	**8**	45.6	12.8	1.52
α	**0.1585**	0.1585	**0.2**	0.2	0.2512	0.3162
Under 10% PER?	**Yes**		**Yes**			

Table VI
RATE REGION DATA SHEET. 18DB FOR NEAR-USER SNR AND 5DB FOR FAR-USER SNR. DATA OF FIGURE 16 (RIGHT).

rate is too small to begin with, interference from N's symbols rapidly degrades its link quality so as to make any far-rate infeasible.

Therefore, the far user modulation and rate pair may be appropriately chosen based on the rate that the near user's traffic demands. For instance, when the near user's spectral efficieny is 1 (QPSK-$1/2$), choosing BPSK-$3/4$ for the far user provides a rate gain of about 28% over TD (as compared to a gain of about 21% over TD for QPSK-$5/6$), while when the near user's spectral efficiency is 3, it is preferable to choose QPSK-$5/6$ for the far user (over BPSK-$3/4$).

2) Impact of Imperfect Interference Cancellation at N: The deviation of $\alpha(k)$ from its ideal value $\beta_{\mathrm{N}}(k)$ is a measure of the residual far-user interference seen at N due to the imperfect cancellation of F's symbols (even when the far-packet is decoded correctly). The β_{N}'s are calculated from the single-user results using (14). The α's are determined from the experiment. For $K = 8$, we find that when $k = 6$, $\alpha(k) = 0.13 \geq \beta_{\mathrm{N}}(k) = 0.1$. At small near-rates the desired symbol stream (of near-symbols) has a much lower power than the interference symbol stream (of far-symbols). In this regime, even small regeneration errors manifest as large residual interference, necessitating an increase of $\alpha(k)$ beyond $\beta_{\mathrm{N}}(k)$. For example, $1 - \beta_{\mathrm{N}}(1) \approx 90\%$ of the transmit power is assigned to F. Even if only 10% of this power remains after cancellation, it is still about the same as the signal power. As $\beta_{\mathrm{N}}(k)$ increases, so does the near-rate, thereby making the near-link susceptable to even smaller levels of residual interference. as well.

The root cause of this problem lies in the small (but non-zero) estimation error in the channel frequency response and a small error in compensating the carrier frequency and phase offsets. Although this level of inaccuracy may result in relatively small losses in a single-user system (as shown in Figure 6), a multiuser system is much less tolerant to these errrors as our results show. Despite this inaccuracy, we find that SC can still provide rate gains using a reasonably well-designed single-user building blocks.

3) Modeling the Near-User Interference at the Far-User: We study the performance of F's (maximum-likelihood) demodulator for the three choices of the interfering (N's) signal's constellations—BPSK,QPSK,16QAM— and for two different interferer strengths $\alpha = 0.2$ and $\alpha = 0.8$. F's rate is maintained at BPSK-1/2. Indeed, as explained in Subsubsection III-B2, the performance of the demodulator is dictated by the superconstellation with $2^{b_{\mathrm{x_N}} + b_{\mathrm{x_F}}}$ points; its error probability critically depends on the effective minimum distance (11). Figure 18 depicts the far user PERs versus the SINR at F which we define as

$$\mathsf{SINR} \triangleq \frac{(1-\alpha)\gamma_{\mathrm{F}}}{1+\alpha\gamma_{\mathrm{F}}}. \tag{16}$$

We observe the following:

- In the weak interference regime (for $\alpha = 0.2$, i.e., SIR ≈ 6 dB), it is seen that BPSK is the *worst interferer*. For small α, the error probability arises primarily from the separation between the clusters (see Figure 17 (left)).
- In the strong interference regime (e.g., $\alpha = 0.8$, i.e., SIR ≈ -6 dB), it is seen that 16QAM is the worst interferer. When α is large, the clusters overlap and the error probability may be attributed to the high density

Far power	Near power	Far SNR	Near SNR	Far PER with BPSK interferer	Far PER with QPSK interferer	Far PER with 16QAM interferer
0.3607	0.0906	0	-6	100	91.6	92
0.4541	0.1141	1	-5	88	39.7	48.8
0.5717	0.1436	2	-4	41.1	10.2	11.9
0.7198	0.1808	3	-3	16	1.8	1.9
0.9061	0.2276	4	-2	3.2		

Table VII
FAR USER PER WITH DIFFERENT NEAR-USER MODULATION TYPES. $\alpha = 0.2$. DATA SOURCE OF FIGURE 8 (TOP) IN .

of the clusters (see Figure 17 (right)).

Validity of the IAGN model: The rate constraint on the far-user in (4) is of the form $R = W\log(1+\mathsf{SINR})$. However, *conditioned on N's codebook*, this interference is clearly not Gaussian. When the channel codes and the detection process at F are allowed to be arbitrarily complex (e.g., allowing Gaussian signaling, ML decoding), the *combined* effect of interference and noise can indeed be shown to be Gaussian for $L \to \infty$ when the interference (N's signal) cannot be jointly decoded with the desired signal (F's signal) [16], which is certainly the case for the Gaussian BC. It is not clear if this remains valid when the choices of channel codes is restricted (e.g., to those with finite L using finite constellations) and/or the receiver architecture (e.g., to demodulate-and-decode). Under such constraints, it is of fundamental interest to check the validity of the Gaussian approximation. We clarify that the interference considered here is from N's symbols only, and not from other transmitters in the network.

Our results show that for SC-BICM systems with a demodulator structure described in Section III-B2, treating the interference-plus-noise term as Gaussian perturbation can be quite inaccurate. For a small near-user power (i.e., small α), the cluster centers are well separated and each daughter point is likely to be close to its parent (see Figure 17 (left)). In this case, a BPSK interference distribution can place *every* daughter point at the farthest possible distance ($\sqrt{\alpha}$) from its parent. For Gaussian interference with the same power, 68% of all daughter points will lie *within* a ball of radius $\sqrt{\alpha}$ centered at the parent point[21]. Thus for a given noise level, Gaussian interference would result in fewer demodulation errors compared to BPSK interference: a Gaussian interference-plus-noise term *underestimates* the number of decoding errors.

For large near-user power, the cluster centers become closer and the clusters begin to overlap (see Figure 17 (Right)), making the precise form of the interference distribution important. For a given interferer power, higher-order interferer constellations (such as 16QAM) result in more densely packed clusters, which upon overlapping result in a smaller effective minimum distance $d_{\text{eff}}^{(\text{F},\text{N})}$ from (11). Viewed from this perspective, it is clear that a Gaussian interferer would be the worst than either BPSK, QPSK, or 16QAM in that it results in infinitely dense clusters. Thus, in the low SIR regime, a Gaussian interference-plus-noise term *overestimates* the number of decoding errors. These trends are apparent from Figure 18. Measurement data is tabulated in Table VII and VIII.

[21]About 68% of values drawn from a normal distribution are within one standard deviation away from the mean.

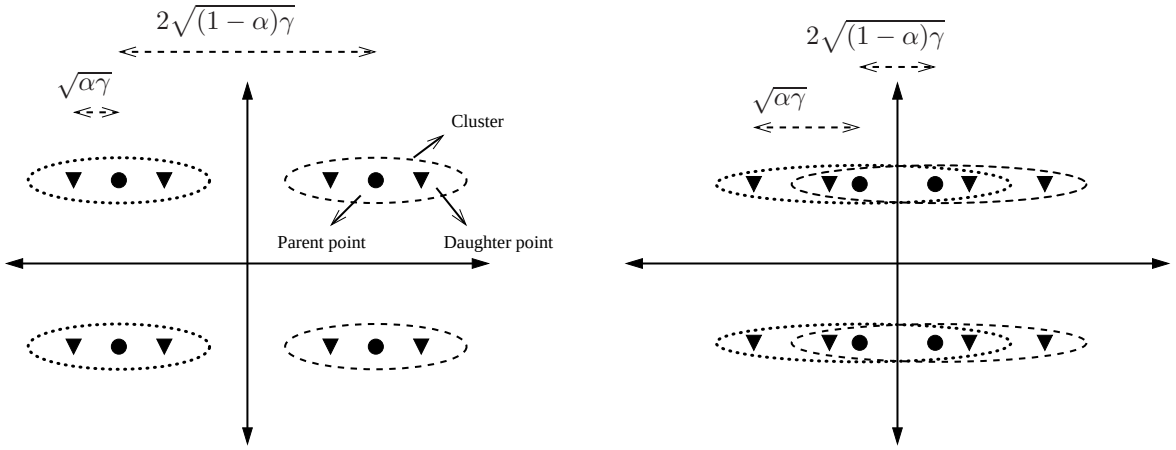

Figure 17. Depiction of the superconstellation points in the weak (left) and strong (right) interference cases. Evidently, when α is small, the PER depends on the cluster separation, while when α is large, the PER is determined by the cluster density.

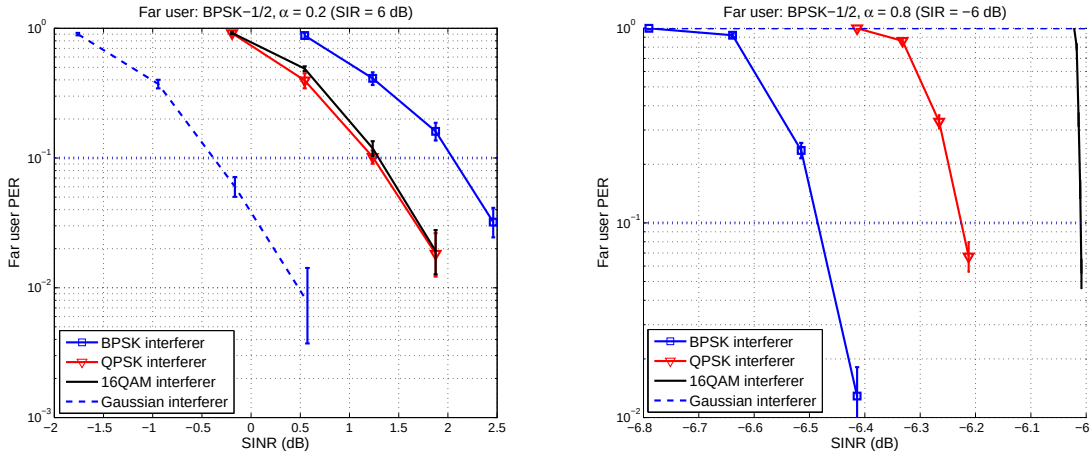

Figure 18. Far user PER versus SINR for different near-user constellations.

IX. Concluding Remarks

A. Summary

We have presented a software-radio implementation of Superposition Coding using off-the-shelf single-user coding and decoding blocks. We experimentally determine the set of achievable rate-pairs for this system under a packet error constraint. Our results suggest that SC can provide substantial gains in spectral efficiencies over those achieved by orthogonal schemes such as Time Division Multiplexing. Our findings also question the validity of treating inter-user interference as Gaussian noise to measure system performance in practical systems, and thereby the validity of the IAGN model for these systems. To the best of our knowledge, this is the first such attempt.

Far power	Near power	Far SNR	Near SNR	Far PER with BPSK interferer	Far PER with QPSK interferer	Far PER with 16QAM interferer
0.3607	0.0906	0	6	100		
0.4541	0.1141	1	7	92.1		
0.5717	0.1436	2	8	23.6		
0.7198	0.1808	3	9	1.3	100	
0.9061	0.2276	4	10		86.2	
1.1408	4.5415	5	11		33.1	
1.4361	5.7174	6	12		6.7	
3.6074	14.3615	16	10			100
4.5415	18.08	17	11			80.7
5.7174	22.7614	18	12			34
7.1978	28.6549	19	13			14.9
9.0615	36.0743	20	14			5.5

Table VIII
FAR USER PER WITH DIFFERENT NEAR-USER MODULATION TYPES. $\alpha = 0.8$. DATA SOURCE OF FIGURE (BOTTOM) IN .

B. Discussion

In our investigation, we made some assumptions to simplify the design process and performance characterization. We now discuss how our framework can be modified to investigate the value of SC even when these assumptions are relaxed.

1) The channel ordering exploited for two-user SC in this report can be extended to three or more users, although finding the rate-region experimentally can be more cumbersome due to the presence of two power variables α_1 and α_2.
2) The code library that was held fixed in our design can in general be a design parameter by itself. This becomes relevant when an SC system is to be built from scratch rather than on top of an existing single-user system as in this report. Nonetheless, the current framework can be adopted to evaluate the performance of a given library.
3) The BS determines α from its knowledge of the individual link SNRs. In practice, these can be obtained via a feedback link. For every such SNR pair, the BS can choose an SC scheme by adopting the procedure described in this report.
4) The "rate-centric" view in this report focuses on the spectral efficiency gains from SC (for a given reliability), and has its roots in information theory. Viewing SC as a multiuser coding scheme, it is also possible to adopt a "reliability-centric" view wherein one measures the reliability gains (due to SC's coding gains) for a given pair of spectral efficiencies. Although both viewpoints are equivalent, they differ greatly in their experimental complexity. We study the latter problem in our recent submission [17] via on-air experiments.

C. Future Work

By experimentally demonstrating the benefits of SC using off-the-shelf single-user techniques, we have shown that SC is a potentially valuable transmission scheme. Our work can be extended in several directions. For example, the code library selection can leverage both advanced coding techniques such as turbo or LDPC codes and advanced receiver architectures based on iterative interference cancellation(see, e.g., [18], [19] and the references therein). Also, the principle of superposition can applied to other cases where signal superposition is known to be theoretically optimal, e.g., in multiple access channels (models for cellular uplinks) and for certain classes of relay channels. In these cases, the superposition process occurs at the *receiver* rather than at the transmitter in the BC; thus signals that interfere have different propagation paths. This difference opens up a new set of problems in system design, such as node synchronization, channel code selection and receiver design. Allowing multiple antennas at the TX and/or RX adds another dimension to the design space. Another possible line of investigation is to analyze the implications of SC for higher layers in the network stack. For example, the problem of scheduling multiple users with an SC-enabled physical layer involves many interesting tradeoffs [20].

X. Acknowledgment

The support of NSF (grants CNS-1016742 and 0830651) and DARPA/IPTO IT-MANET program (grant W911NF-07-1-0028) is gratefully acknowledged.

References

[1] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed., John Wiley & Sons, Inc., 2006.

[3] J. Mitola III, "Software radios: Survey, critical evaluation and future directions," *IEEE Aerospace and Electronic Systems Magazine*, vol. 8, no. 4, pp. 25-36, 1993.

[4] *GNU Radio*, http://www.gnu.org/software/gnuradio/.

[5] X. Li, W. Hu, H. Yousefizadeh, and A. Qureshi, "A case study of a MIMO SDR implementation," in *Proceedings of IEEE Military Communications Conference (MILCOM)*, pp. 1-7, Nov. 2008.

[6] E. Bayraktaroglu, C. King, X. Liu, G.Noubir, R. Rajaraman, and B. Thapa, "On the performance of IEEE 802.11 under jamming," in *Proceedings of The 27th IEEE Conference on Computer Communications (INFOCOM)*, pp. 1265-1273, Apr. 2008.

[7] K. Mandke, S.-H. Choi, G. Kim, R. Grant, R. C. Daniels, W. Kim, R. W. Heath, and S. M. Nettles, "Early results on Hydra: a flexible MAC/PHY multihop testbed," in *Proceedings of IEEE 65th Vehicular Technology Conference (VTC2007-Spring)*, pp. 1896-1900, Apr. 2007.

[8] R. Alimi, L. Li, R. Ramjee, H. Viswanathan, and Y. R. Yang, "iPack: in-network packet mixing for high throughput wireless mesh networks," in *Proceedings of The 27th IEEE Conference on Computer Communications (INFOCOM)*, pp. 66-70, Apr. 2008.

[9] R. K. Ganti, Z. Gong, M. Haenggi, S. Srinivasa, D. Tisza, S. Vanka and P. Vizi, "Implementation and experimental results of superposition coding on software radio", *2010 IEEE Conference on Communication (ICC 2010)*, May 2010.

[10] J. G. Proakis, *Digital Communications*, 4th ed., McGraw-Hill, 2000.

[11] A. G. Fabregas, A. Martinez and G. Caire, *Bit-Interleaved Coded Modulation*, Foundations and Trends in Communications and Information Theory, Vol. 5:No 1-2 pp. 1-153, 2008.

[12] G. Caire, G. Taricco and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Transactions on Information Theory*, Vol. 44, pp. 927-946, May 1998.

[13] G. Caire and E. Viterbo, “Upper bound on the packet error probability of terminated trellis codes,” *IEEE Communication Letters*, Vol. 2, pp. 2-4. Jan. 1998.

[14] J. Heiskala and J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*, Sams, 2001.

[15] T. M. Schmidl and D. C. Cox, “Robust frequency and timing synchronization for OFDM,” *IEEE Transactions on Communications*, vol. 45, pp. 1613-1621, 1997.

[16] F. Baccelli, A. El Gamal, and D. Tse, “Interference Networks with point-to-point codes”, *IEEE Trans. Info. Theory*, vol. 57, no. 5, pp. 2582-2596, May 2011.

[17] S. Vanka, S. Srinivasa and M. Haenggi, “Improving Broadcast Coverage via Superposition Coding: An Experimental Study”, manuscript in preparation.

[18] R. Zhang and L. Hanzo, “A Unified Treatment of Superposition Coding Aided Communications: Theory and Practice,” *IEEE Communications Surveys and Tutorials*, No. 99, pp. 1-18, July 2010.

[19] J. Andrews, “Interference Cancellation for Cellular Systems: A Contemporary Overview”, *IEEE Wireless Communications Magazine*, pp. 19-29, Apr. 2005.

[20] P. Vizi, S. Vanka, S. Srinivasa, M. Haenggi, and Z. Gong, “Scheduling using Superposition Coding: Design and Software Radio Implementation,” *IEEE Radio and Wireless Week 2011*.

[21] F. Abbas, *Simple User Manual for Gnuradio 3.1.1*, GNU Radio, 2007.