An Open, Scalable Massive MIMO Testbed Operating at E-Band Frequencies

James Dunn
Borivoje Nikolic, Ed.
Ali Niknejad, Ed.
Elad Alon, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2020-194
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-194.html

December 1, 2020

An Open, Scalable Massive MIMO Testbed Operating at E-Band Frequencies

by

James Dunn

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Engineering — Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Borivoje Nikolić, Chair
Professor Ali Niknejad
Professor Elad Alon

Fall 2019

The thesis of James Dunn, titled An Open, Scalable Massive MIMO Testbed Operating at E-Band Frequencies, is approved:

Chair ____ Date ________________

____ Date ________________

____ Date ________________

University of California, Berkeley

An Open, Scalable Massive MIMO Testbed Operating at E-Band Frequencies

Abstract

An Open, Scalable Massive MIMO Testbed Operating at E-Band Frequencies

by

James Dunn

Master of Science in Engineering — Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Borivoje Nikolić, Chair

Multi-user massive MIMO enables orders of magnitude higher spectral efficiency than non-spatial techniques, increasing channel capacity and enabling the next generation of wireless communications. This thesis presents hardware design, assembly, and testing of Hydra, a real-time, multi-user massive MIMO system. Hydra consists of a base station and user equipment constructed with commercial, off-the-shelf components, and addresses implementation and scalability issues with current massive MIMO systems by demonstrating hardware-efficient distributed beam-forming at mm-wave carrier frequencies.

Contents

List of Figures

List of Tables

List of Abbreviations

ADC Analog-to-digital converter

ASIC Application-specific integrated circuit

COTS Commercial, off-the-shelf

DAC Digital-to-analog converter

DSP Digital signal processing

GPSDO GPS-disciplined oscillator

GUI Graphical user interface

IC Integrated circuit

LNA Low-noise amplifier

LO Local oscillator

MIMO Multiple input, multiple output

MMIC Monolithic, millimeter-wave integrated circuit

PA Power amplifier

PCB Printed circuit board

PRBS Pseudo-random bit sequence

RF Radio frequency

SINR Signal to interference and noise ratio

SNR Signal to noise ratio

UE User equipment

VGA Variable gain amplifier

Acknowledgments

I first thank my advisor, Professor Borivoje Nikolić. His knowledge at every level of the hardware/software stack has been an inspiration to me, and he has been supportive at all steps in this degree to make me a well-rounded engineer. I additionally thank my committee Professors Ali Niknejad and Elad Alon for their mentorship and guidance on this project, and graduate advisors Shirley Salanio and Michael Sun for helping me navigate the degree program. There are many students whose prior work is the basis for this system: Antonio Puglielli, Emily Naviasky, Lucas Calderin, Lorenzo Iotti, Sameet Ramakrishnan, and especially Greg Lacaille. I greatly appreciate Anita Flynn's work on improvements for the next revision of Hydra, and on 3D CAD drawings used in this thesis. Her attention to practical details is unmatched. I thank Zhenghan Lin and Yue Dai for assistance with system measurements. I thank the JUMP ComSenTer program, which funded much of this work and has allowed me to learn from other universities pursing device and application-level wireless research. I am grateful for managers Ajith Amerasekara and Ken Lutz, whose support allowed me to pursue a Master's degree while employed as a staff member at the Berkeley Wireless Research Center. Fellow BWRC staff members Candy, Mikaela, Yessica, Brian, Fred, Amber, Greg and Jeff kept everything running smoothly and are a joy to work with. Finally, I thank my family and partner Sylvia for love, encouragement, and curiosity toward my work.

Chapter 1

Introduction

1.1 Motivation

As mobile handset data consumption continues to increase, and with new wireless applications such as industrial control, autonomous vehicles, and virtual reality on the horizon, the demand to extend wireless channel capacity by orders of magnitude shows no sign of ceasing. With spectrally-efficient multiplexing techniques in time, frequency, and code already driving channel capacity to within a fraction of a dB of the Shannon limit, the next promising domain to extend channel capacity is space. The massive MIMO paradigm, in which a large number of base station antenna elements serve a much smaller number of users has emerged as one of the most promising means of increasing channel capacity to projected demand. As [4] demonstrated, the number of users served as well as interference reduction scales with the number of antenna elements in an array. It is therefore desirable to maximize this number of antenna elements to exploit the spatial domain as much as possible.

A recent direction of research in massive MIMO systems is toward implementation at mm-wave carrier frequencies. The reduced size of mm-wave antennas allows necessarily large arrays to be implemented in more compact, lightweight, inexpensive form factors. The antenna gain of these large arrays compensates for increased path loss associated with higher frequencies. Available bandwidth in the E-Band when sub-6GHz spectrum is nearly fully allocated makes mm-wave implementation more desirable to network operators.

An additional consideration for network operators is flexibility. To realize the ubiquitous connectivity envisioned in 5G and beyond, network operators will deploy base stations in a wide variety of settings. For example, base stations targeting a streetlamp post versus a telecommunications tower will need to serve different numbers of users, with different degrees of inter and intra-cell interference.

Given that mm-wave antenna sizes allow arrays to consist of potentially hundreds of elements, and a wide range of array size variants will need to be implemented in different settings, many system-level challenges arise. How must the architecture be designed to allow for hundreds of antenna elements in a mm-wave base station, and how must it be designed to work well for large ranges of array sizes from as few as ten, to as many as hundreds?

This thesis presents hardware design, assembly, and testing for a Hydra, a massive MIMO system that addresses these issues using modular hardware for a distributed method of linear beam-forming that scales with large numbers of array elements. Ultimately, the goal is to provide this scalable system to the broader academic and industrial research community as an open test bed, fostering development and testing of new massive MIMO technologies. With this goal in mind, this work targets the following features of a useful test bed:

- Accessible: The test bed IP should be open-source and well-documented, such that it can be obtained, understood, and applied by all who seek to use it. Hardware should be commercially available.

- Flexible: The test bed should be valuable to a variety of different massive MIMO research projects at different carrier frequencies, bandwidths, size constraints, power constraints, and other variations. Commercially-available hardware components should be interchangeable with custom hardware.

- Verified: The hardware should be functionally verified so that higher-layer research at the MAC or application level may abstract the physical layer.

1.2 Prior and Related Work

1.2.1 Prior Work on the Hydra System

This thesis details work on hardware design, assembly, and testing for the full-scale Hydra system, and the associated challenges and measurements. However, a great deal of work on Hydra has preceded this thesis. The architectural and DSP ideas for implementation of a distributed massive MIMO system were conceived of and detailed in [6], as well as the implementation of hardware for a subset of the Hydra system at 4 base station channels and 2 UEs. This work reuses certain hardware from [6], and contributes new revisions of other hardware components to allow for scaling to larger numbers of base station channels and UEs. Additionally, all DSP, the software control suite, and much of the hardware planning for Hydra was done in [1].

1.2.2 Related Work on Massive MIMO Hardware Testbeds

As a result of massive MIMO's promising capabilities, it has become the topic of much academic and industrial research. [9] demonstrated COTS hardware design and linear precoding schemes for a 2.4 GHz, 64-antenna massive MIMO system. [8] improved upon this work with a more modular hardware design, allowing for implementation of larger numbers of antennas from a mechanical standpoint. These works implement a form of decentralized conjugate beam-forming, but do not implement de-correlation, instead assuming independent, uncorrelated channels and scaling the transmission power at each base station antenna. However, this distributed method still uses a "fat tree" structure in which there are semi-centralized processing nodes. Such an architecture will be difficult to scale to wider channel bandwidths at mm-wave. [10] also demonstrates a COTS, 2.6 GHz massive MIMO system, evaluated with different methods of spatial processing. This system performs all baseband processing on a centralized unit, requiring hundreds of gigabits of interconnect bandwidth to this processor for 20 MHz bandwidth and 10 users. As we target 250 MHz bandwidth with greater than 16 users, a different architecture is clearly needed. More recently, [11] demonstrated a 64-antenna system operating at 28 GHz carrier frequency with 500 MHz bandwidth. This system implements similar modular hardware and distributed conjugate beam-forming, but does not perform any de-correlation using a central element with full channel state information.

1.3 Thesis Scope and Organization

The scope of this thesis consists of system design, hardware implementation, and measurement and validation for the Hydra system. Architectural considerations for the implementation of a Massive MIMO system are discussed in Chapter 2. This architecture informs the high-level system design, which is described in Chapter 3. Chapter 4 details hardware design for system components. Chapter 5 presents measurement results that verify functionality and measure system performance. Finally, Chapter 6 describes conclusions, potential applications, and future improvements to this system.

Chapter 2

Massive MIMO Architectural Considerations

While the reduced size of mm-wave antennas physically allow for denser arrays numbering hundreds of elements, current back-end processing architectures for RF MIMO arrays do not easily scale to these numbers of channels. In this chapter, we analyze architectural considerations for a mm-wave Massive MIMO system.

2.1 Methods of Spatial Processing

We first consider which form of spatial processing is most appropriate for large arrays. Examples of spatial processing techniques that consist of linear matrix operations include conjugate beam-forming and zero forcing; non-linear methods include maximum likelihood detection and sphere decoding [5]. [3] demonstrated that as the number of base station antennas M grows asymptotically large, linear spatial processing eliminates uncorrelated noise and fast fading entirely. [6], [7] demonstrated that, as user streams become asymptotically de-correlated through physical separation in a Rayleigh channel, simple conjugate beam-forming becomes optimal. For practical situations with finite M and ratios of M to K less than 256, interference must be considered with a processing method that performs channel de-correlation. [6], [7] additionally showed that for ratios of M to K greater than 4, linear de-correlation processing methods such as zero-forcing reasonably approach the performance of non-linear de-correlation processing methods such as maximum-likelihood. Importantly, zero-forcing computational complexity scales as the product of M and K, while maximum likelihood detection scales as the exponent of K. We conclude from these results that for M to K ratios between 4 and 256, zero-forcing is the most performant and efficient method of spatial processing. We begin with this linear method as the basis for our scalable system.

2.2 Distribution of Hybrid Spatial Processing

Given that linear zero-forcing performs and scales well for large arrays from a computational complexity standpoint, we next investigate how to distribute the processing across computational elements most effectively. Considerations for processing distribution include the maximum computational load placed on any single element, as well as the interconnect bandwidth required between elements. As the amount of channels in a massive MIMO system may number in the hundreds, each operating at potential bandwidths on the order of GHz, these system characteristics have important implications for cost, power, and deployment flexibility at the physical layer.

Linear, spatial processing methods such as zero-forcing generally consist of two phases: beam-forming and de-correlation. [2] presented an analysis of separate beam-forming and de-correlation stages, known as hybrid pre-coding. First stage beam-forming uses complex conjugate operations to maximize SNR and form a beam to each user across the antenna array, reducing the channel matrix rank from M antennas to K users. Importantly, this operation may be performed without global channel state information. That is, it may be performed across C modules, with each operation reducing rank from M/C to K. While this stage maximizes SNR, it does not maximize SINR, as it does not remove inter-user interference. To reduce inter-user interference, a second stage de-correlation operation must be performed. This operation requires global channel state information of order K. We next explore two architectures in which these phases of spatial processing may be distributed, and the physical layer implications for each.

We investigate a scenario in which C modules, each consisting of P antennas, analog radio front-ends, and data converters, are connected to a base-band processing element. In Figure 2.1, we show a centralized architecture in which C modules each pass samples directly to a central element. This central element performs signal aggregation and beam-forming before sending data for de-correlation and other baseband DSP (shown as *baseband post processing*). In this case, interconnect bandwidth and dimension of matrix multiplication operations scale with $M = C * P$, the total number of antenna elements. This is because data is passed before being reduced from antenna space to user space by the beam-forming operation.

In contrast, a distributed architecture is shown in Figure 2.2. In this case, C modules additionally perform first stage beam-forming across their respective P antennas, reducing data order to K. Samples are accumulated with neighboring modules in sequence until the final module sends data for de-correlation and post-processing. Data interconnect scales with K, and dimension of matrix multiplication operations scales with the greater of K and P. As K is necessarily much less than M in a massive MIMO system, and P may be made arbitrarily smaller than M based on the number of modules, interconnect and computational requirements are significantly relaxed. Uniform modules may be added to increase array size and therefore channel capacity, allowing for more flexible implementations across the design space of system power and cost.

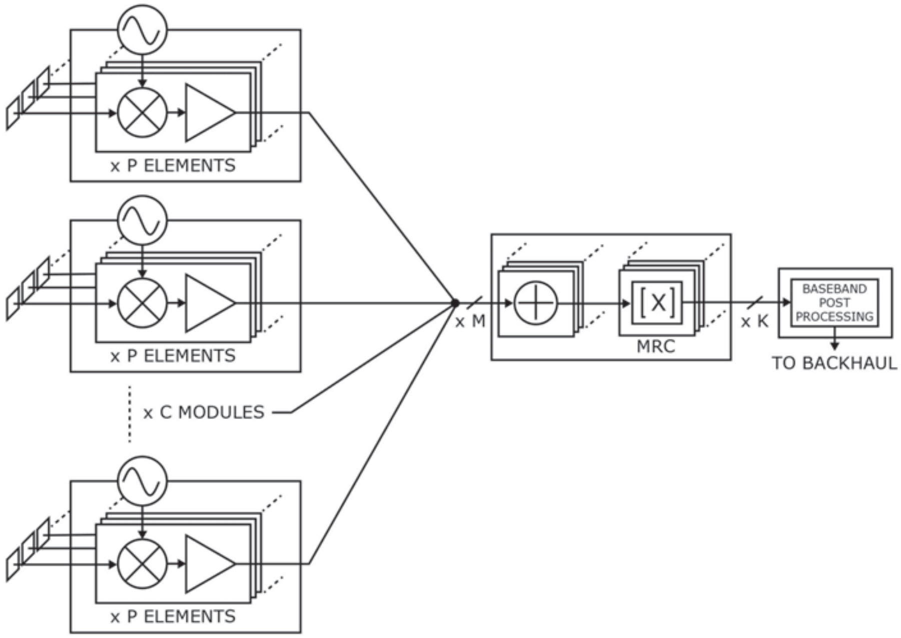


Figure 2.1: Centralized beam-forming architecture. Interconnect bandwidth scales with M.

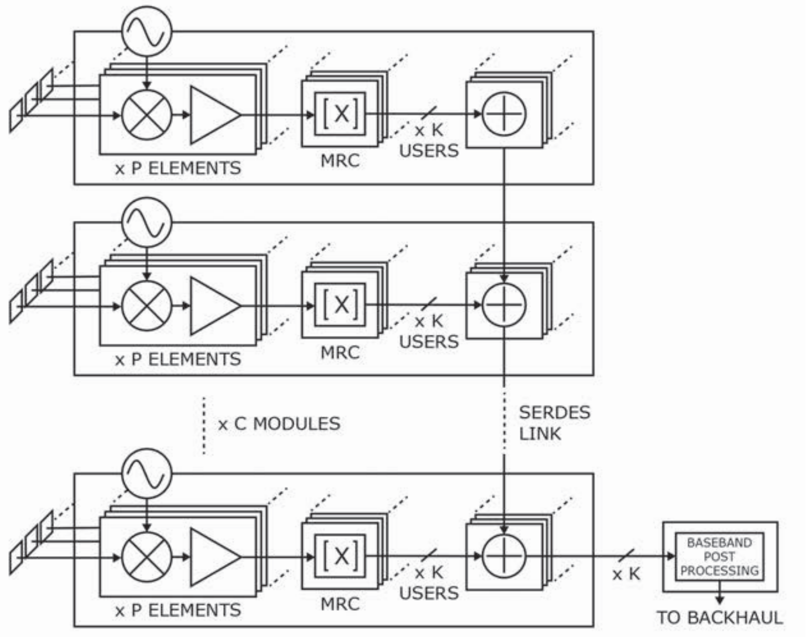


Figure 2.2: Distributed beam-forming architecture. Interconnect bandwidth scales with K.

Chapter 3

System Design

Based on an analysis of the distributed architecture in Chapter 2, this chapter presents high-level system design decisions. Figure 3.1 presents a block diagram of this system design. This work is an up-link system, which means that user equipment transmits data, and the base station receives it. The base station consists of antenna arrays, data converters, and beam-forming processing, arranged in modules. The beam-forming processing unit is chosen to be an FPGA to allow for real-time processing and high-bandwidth data transfer between modules. These beam forming FPGAs are connected with a SERDES link to allow data aggregation shown in Figure 2.2. For zero-forcing de-correlation and further DSP post-processing, we use a PC. Though this software processing cannot match the performance of hardware, it provides flexibility for rapid prototyping and is fast enough to process data at rates of roughly 100 Kbps. An FPGA may be used in place of the PC for faster real-time post-processing. Payload data and channel state information for processing is sent to the host PC over Ethernet. User equipment does not perform spatial processing, and must only transmit data synchronously. This can be achieved with a low-performance FPGA that is programmed to replay a set of pilots and data payloads. In addition to de-correlation, the host PC may be used for system-wide control. This includes JTAG programming of both base station and user equipment FPGAs, and UART to transfer data payloads. Finally, we consider a method for synchronizing data frames, which consist of a pilot sequence and data payload, between the transmitting user equipment and receiving base station. To achieve this, we wire a low frequency frame synchronization clock shared between all UEs and the base station.

High-level system parameters for Hydra are given in Table 3.1.

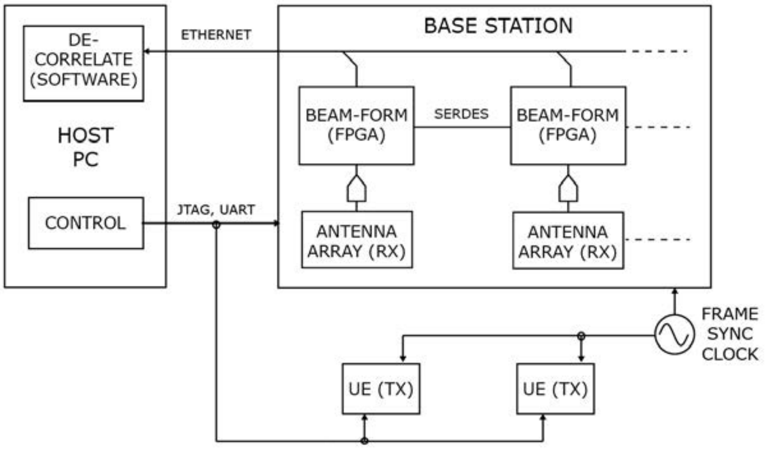


Figure 3.1: Block diagram of Hydra system.

Parameter	**Variable**	**Value**
Bandwidth	W	250 MHz
Carrier frequency	f_c	75 GHz
Sampling rate	F_S	500 MHz
No. subcarriers	$N_{subcarriers}$	256
No. UEs assembled (tested)	K	12 (3)
No. BS antennas assembled (tested)	M	96 (20)

Table 3.1: High-level system parameters.

Chapter 4

System Hardware

4.1 Base Station

This section details hardware design for the distributed base station. The first consideration is how to partition functionality of the distributed system into hardware elements. We consider the following functional components of the base station:

- Receive antenna array.
- Analog mm-wave radio (amplifiers, local oscillator, mixer).
- Data conversion (analog-to-digital in case of receive base station).
- Power sequencing and distribution.
- Digital status and control.
- Conjugate beam-forming computation.
- Zero-forcing de-correlation computation.
- Interconnect within modules, between modules, and to back-haul.

We organize this functionality into the following hardware modules:

- *Head*: A PCB consisting of the antenna array and analog mm-wave radio components.
- *Spine*: A set of PCBs consisting of data conversion, digital status and control, conjugate beam-forming, and data aggregation between modules.
- *Tail*: Zero-forcing de-correlation, back-end DSP.

A block diagram of the modules and connections comprising the base station is shown in Figure 4.1. Detailed descriptions of the hardware module implementation are provided in the following sections.

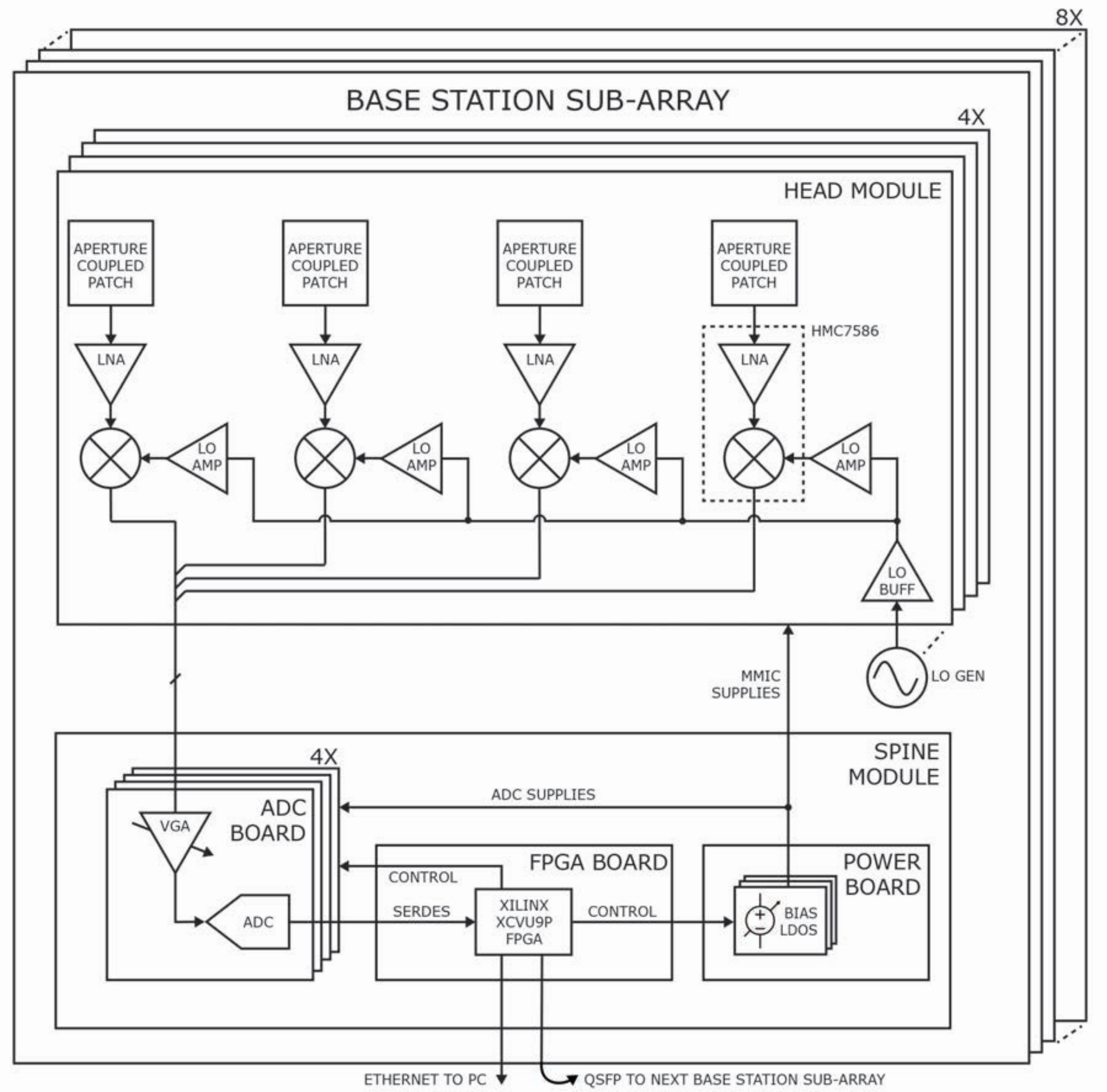
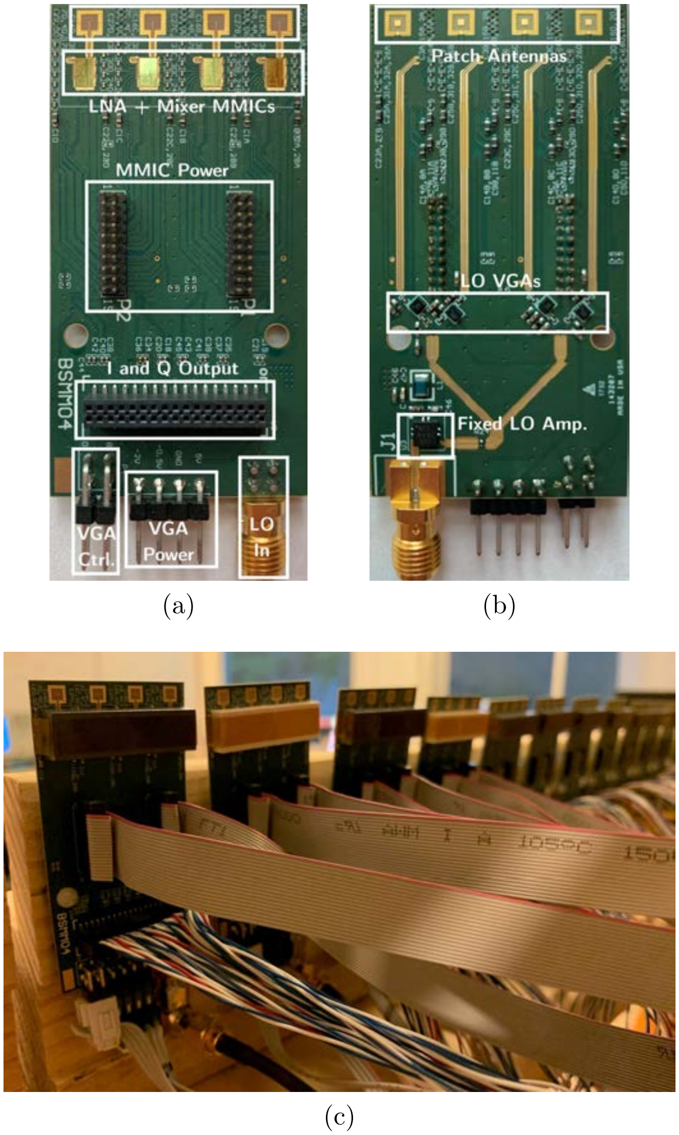


Figure 4.1: Block diagram of base station.

4.1.1 Head

The head module, designed in [6], consists of a single PCB housing antenna and radio hardware. There are 4 head modules per base station slice, and each head module PCB consists of 4 receive channels. Each receive channel consists of a custom micro-strip aperture-coupled patch antenna, E-band LNA and down-convert mixer receive MMIC, and VGA for the LO signal to each mixer IC. The I and Q outputs of the receive MMIC are routed via controlled-impedance traces to a twisted-pair connector header. The LO signal is inputted



(a)

(b)

(c)

Figure 4.2: Head PCB module. a): Front of fabricated PCB. b): Rear of fabricated PCB. c): Installed in array.

to the PCB via SMA header, buffered with a fixed RF amplifier, split to four channels via three Wilkinson dividers, and amplified with four variable gain RF amplifiers. The PCB stack-up is controlled to provide desired antenna characteristics and trace impedance. All components are standard surface mount except the MMIC, which is a wire-bonded chip-on-board. Fabricated PCB photos and placement in array are shown in Figure 4.2.

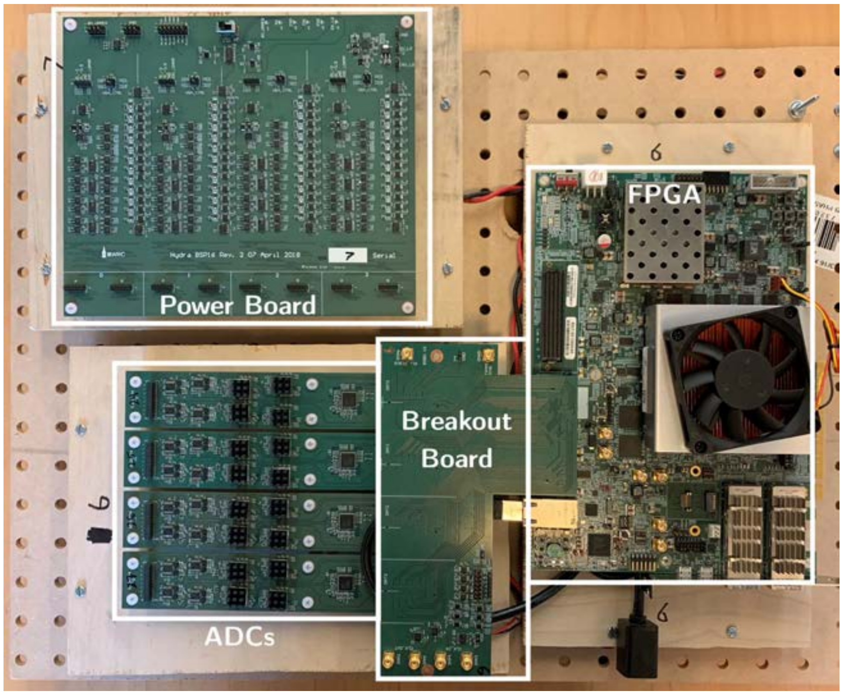


Figure 4.3: Components of one spine slice.

4.1.2 Spine

The spine is a module consisting of several PCBs arranged together in a *slice*, which may then be tiled as depicted in Figure 2.2. Figure 4.3 shows the components of a single slice, and Figure 4.4 shows slices tiled together in a full base station array. The spine performs data conversion, conjugate beam-forming, inter-module aggregation, and auxiliary functionality such as control and power. The following sections describe the design of each PCB in the spine.

Analog-to-Digital Conversion Board

The analog-to-digital conversion board is a custom PCB with a modular form factor, designed in [6]. There are four ADC boards per base station slice, and each board contains four commercial ADC ICs, for a total of 16 I and 16 Q channels per cluster. Each four-channel ADC board corresponds to one head module. These boards take analog baseband I and Q data from the head module and digitize into samples. Each ADC IC quantizes to 5 bits at 500 Msps. Also present is a clock distribution IC for delivering sample clocks to the ADCs. The output of the ADC ICs conforms to the JESD204B serial interface standard. The outputs of each ADC IC are are routed to a mezzanine connector, which attaches to the breakout board.

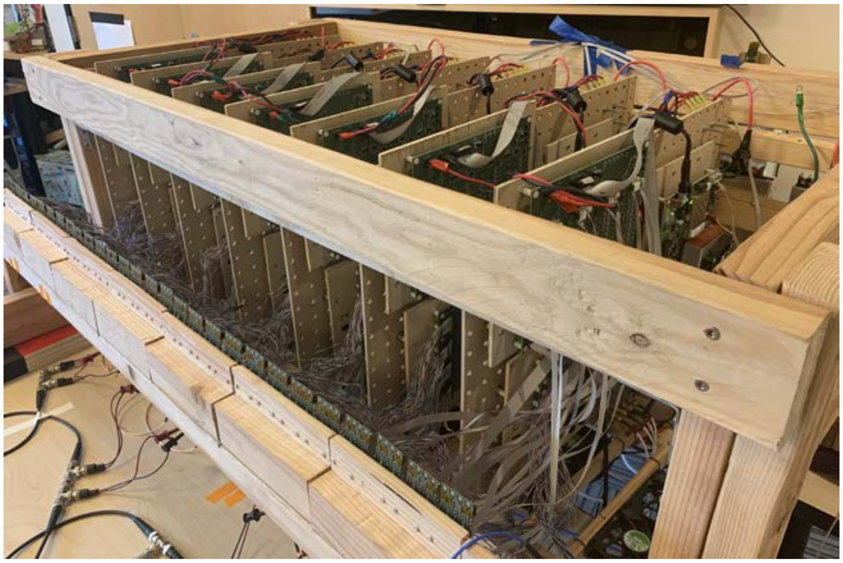


Figure 4.4: Spine slices assembled into full base station

Breakout Board

The breakout board performs two functions. First, it aggregates the SERDES data lanes from the four ADC PCBs into a single mezzanine connector to the FPGA board. Second, it distributes the baseband clock to the slice, as well as buffering the clock and outputting to the next slice in the distributed array.

FPGA

The FPGA performs conjugate beam-forming on samples and aggregates beam-formed data from neighboring slices. Additionally, the FPGA handles SPI and I2C status and control for all components of the slice. We use a commercially available development board from Xilinx, chosen primarily for its lateral SERDES connectivity via input and output QSFP connectors. An Ethernet connection to the host PC provides channel state information for de-correlation. Digital I and Q samples are fed to the FPGA from the ADCs via the breakout board. Conjugate beam-forming is performed for the slice, data is aggregated from a previous slice via an external QSFP connector, and sent to the next slice via a separate QSFP connector. Channel state information for this slice is sent over Ethernet to the host PC.

Power Board

The power board is a custom PCB that biases ICs of the head PCB. This includes supplies for the LNA/mixer MMIC, LO VGA, and fixed LO buffer. There is one power board per base station slice, with independent supplies for 16 receive channels. The LNA and mixer MMICs of the head PCBs have no internal biasing circuitry and require a particular power sequence at startup. This board contains sets of variable LDOs and digital current sensors that implement this biasing sequence for each receive chain channel in the slice. The variable LDOs and digital current sensors are controlled via I2C and SPI from the FPGA.

4.1.3 Tail

The tail performs zero-forcing de-correlation, as well as additional baseband post-processing, such as timing and carrier recovery and demodulation. The base station spines connect to the tail via Ethernet. Each spine slice provides its channel matrix to the tail over this connection, as de-correlation requires full channel state information. Additionally, beamformed and aggregated data over all the spine slices is transmitted from the final slice in the chain over Ethernet to the tail. Tail spatial processing and subsequent DSP is performed in software with Python.

4.1.4 Chassis

Figure 4.4 depicts the base station system chassis housing multiple spine slices and head arrays. The chassis frame and slice platforms are constructed from wood for rapid prototyping, light weight, and minimal attenuation or reflection. Spine slices are mounted vertically and connected to a horizontal antenna array.

4.2 User Equipment

This section details hardware design for the user equipment. As with the base station, we consider how to partition functionality of the user equipment into hardware elements. We consider the following functional components of the user equipment:

- Transmit antenna element.
- Analog mm-wave radio (amplifiers, local oscillator, mixer).
- Data conversion (digital-to-analog in case of up-link base station).
- Power sequencing and distribution.
- Digital status and control.
- Baseband: playback of pilot sequence and data payload.

A block diagram of the modules and connections comprising one user equipment is shown in Figure 4.5. Detailed descriptions of the hardware module implementation are provided in the following sections.

Radio Front-End

This custom PCB is similar to the base station head, but contains a transmit rather than receive chain, and one channel per board. The transmit chain consists of commercial MMIC PA and up-converter, and an LO buffer. Analog baseband is input as differential I and Q from micro coaxial connectors.

Digital-to-Analog Converter

This single channel DAC is a commercially available evaluation board. It receives digital samples from the FPGA via the breakout board and sends analog samples to the radio front-end.

Breakout Board

This custom PCB routes SERDES output from the FPGA to the DAC board. Additionally, it contains clocking, power, and JTAG modules for the FPGA.

FPGA

A Xilinx Artix 7 FPGA implements transmit baseband functions. A pilot sequence and data sequence are uploaded to the FPGA's block RAM, modulated, and sent to the DAC over the FPGA's SERDES interface. Like the base station, this FPGA also generates I2C and SPI control for auxiliary UE components.

Power Board

The power board performs bias sequencing for the single-channel transmit MMIC using variable LDOs and digital current sensors.

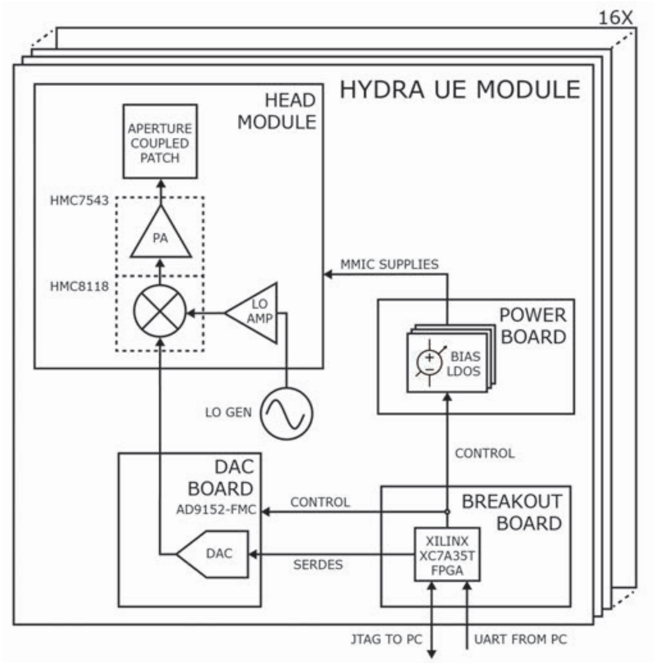
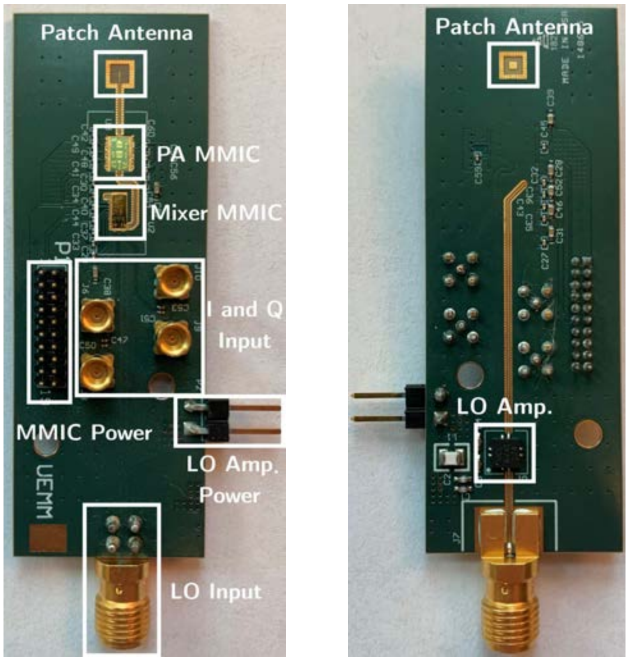


Figure 4.5: Block diagram of user equipment.



(a) (b)

Figure 4.6: User equipment front-end radio PCB. a): Front of fabricated PCB. b): Rear of fabricated PCB.

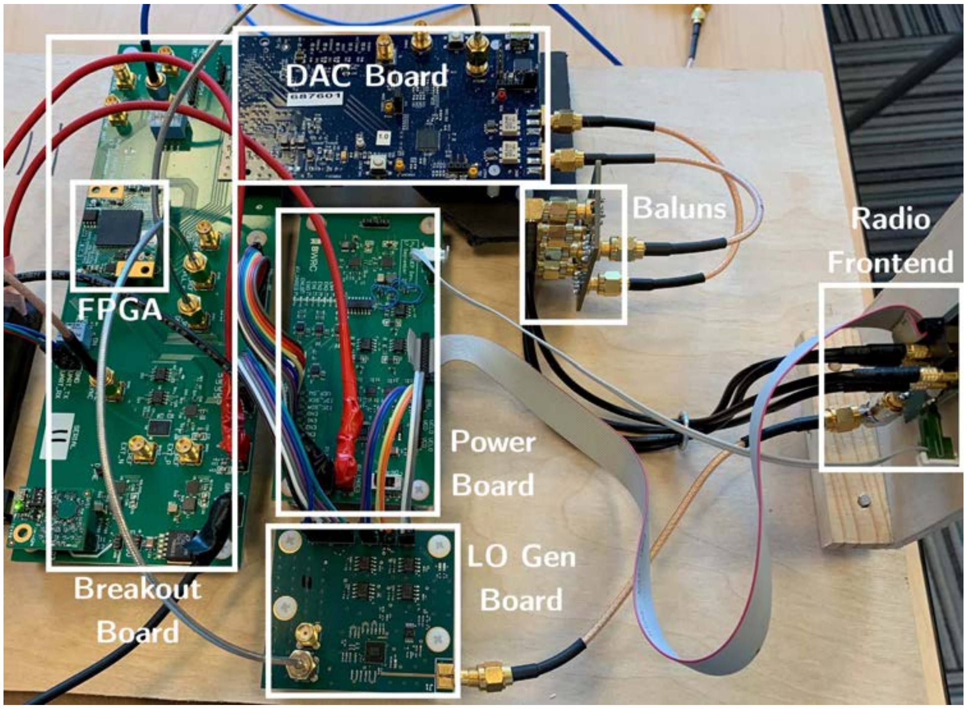


Figure 4.7: Additional user equipment hardware, assembled into a single module.

Chapter 5

Validation and Measurement

This chapter describes the experimental setup and procedures to verify functionality and measure performance of individual system components and the up-link system as a whole.

5.1 Software Control

The software used for controlling measurement hardware and displaying results is a custom GUI implemented in PyQT for [1]. This GUI allows the user to set parameters associated with hardware and DSP, and provides visualization of results. This includes time domain symbol data, frequency domain channel response, and demodulated constellation plots. An example of the GUI environment displaying results is shown in Figure 5.6.

5.2 Hardware Verification

5.2.1 Analog Radio Functionality

We first test functionality of the analog radios by ensuring that they bias properly, and transmit (UE) or receive (base station) with sufficient SNR.

Bias Testing and Assembly Yield Issues

We automate the biasing of analog radio MMICs with Python scripts that call .tcl scripts, which run on the FPGA to generate SPI and I2C commands that control biasing ICs on the power board. The biasing scripts apply a drain voltage and ramp a gate voltage for each supply on the radio MMICs until a target drain current is reached. The biasing scripts report an error if a drain current goes out of acceptable range, or if a gate voltage must go out of range to achieve the proper drain current. A drain current immediately going out of range indicates a short, likely in a wire-bond of one of the MMICs. A gate voltage going out of range to produce the desired drain current indicates other damage to the MMIC, such as a poor wire-bond connection.

LO Gain Yield Issues

Once radio channels with biasing errors are eliminated, the next source of yield loss is LO gain. Of base station channels that bias correctly, a large number have no signal above the noise floor in a time domain view of a received packet. We notice a strong positive correlation between channels with high SNR and the mixer MMIC supply bias point on the front-ends, and that the bias points fall into two bins (Figure 5.1). Channels with mixer bias points above -0.5V generally correspond to usable SNR, and for below -0.5V to no SNR. The mixer bias point additionally has a positive correlation with LO signal gain, which we believe is the underlying issue. The LO signal travels along an impedance-controlled trace from the input SMA to the mixer MMIC. Along this trace, there is a via from the top to bottom PCB layer to accommodate layout. This trace and via are easily warped by a combination of the thin front-end radio PCBs (designed as such to achieve micro-strip antenna EM characteristics), thermal cycles caused by the large amounts of heat radiated by the MMICs, and mechanical stress placed on the PCBs by mounting and connectors.

Table 5.1 summarizes yield loss on base station and UE front-ends. We ultimately identify 20 functional base station and 3 functional UE radios.

Radios	**Base Station**	**UE**
Assembled	96	12
With bias errors	18	0
With insufficient LO gain	42	7
With unknown SNR issues	16	2
Functioning	**20**	**3**

Table 5.1: Yield issues and number of channels affected.

5.2.2 UE Transmit Power

Transmit power of the three functional UE radios was determined with an E-Band power meter and sweep of distance. Measurements were de-embedded from path loss and antenna gain of the power meter. Plots of these measurements for three UEs are shown in Figure 5.2. Two UEs have transmit power of roughly 5dBm, while one UE is 12dBm, demonstrating the wide variation in analog radio characteristics seen across functioning UEs.

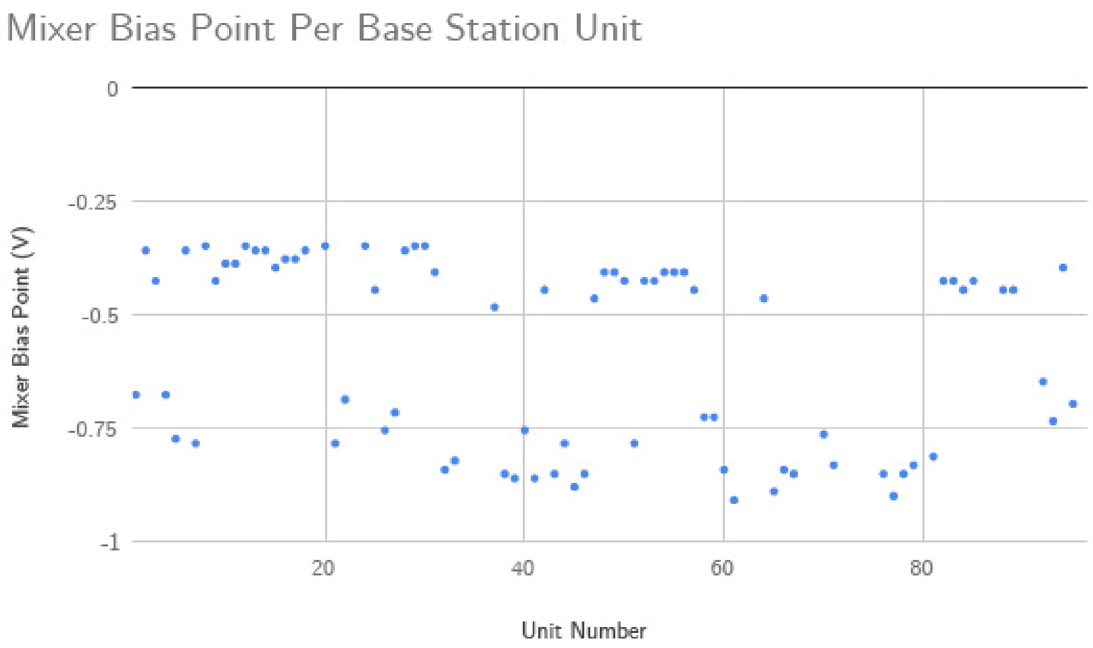


Figure 5.1: Plot of mixer MMIC gate bias points for base station radio modules. Bias points below -0.5V indicate low LO gain and non-functionality.

5.2.3 Baseband Hardware Functionality

Baseband hardware, including DACs, ADCs, breakout PCBs, and FPGAs, were verified by connecting a cabled channel between the UE DAC and base station ADC, bypassing radios. The bit sequence transmitted by the UE DAC was successfully demodulated with high SNR for each base station slice, indicating full yield on baseband hardware. The procedure for this baseband test is described in Appendix B.

5.3 System

After verifying individual hardware components, we test and measure the full system as configured in Figure 3.1. As explained in Section 5.2.1, we experienced significant yield loss among the analog radio front-end PCBs, reducing our number of base station channels to 20 and number of UEs to 3. However, there is meaningful experimental data that we may collect from this configuration. This section describes tests to verify and measure the full up-link system and spatial processing in single and dual-UE configurations for 20 base station channels. The experimental configuration is shown in Figure 5.3, where a single UE is attached to a mobile cart for distance and angular measurements (marked on floor), and a second stationary UE is added for dual-UE experiments.

5.3.1 Single-UE Measurements

We first measure a scenario with 1 UE and 20 base station radios. UE 2 from Figure 5.2 is used in this experiment for its strong transmit power. To first verify functionality, we transmit a frame containing pilots and data payload with a single UE at a distance of 50 inches and angle of 0 degrees (broadside). The frequency domain channel estimation across 256 sub-carriers for a single run is shown in Figure 5.4. A demodulated QPSK constellation plot for received data is shown in Figure 5.5. The channel estimate is as expected, with a relatively flat response for all sub-carriers. The constellation plot shows a clear pattern of QPSK demodulation. Once functionality has been verified, we perform sweeps of distance and angle to measure system performance.

SINR with Distance

This experiment fixes a single UE at a broadside angle and moves the UE away from the base station array to determine SINR as a function of UE distance. Results are shown in Figure 5.6. Multiple measurements are taken, corresponding to multiple data points at each distance point. We observe a reduction of roughly 6db SINR at a distance of 120 inches, and a higher deviation in results, likely due to placement-sensitive multi-path effects at larger distances.

SINR with Azimuth

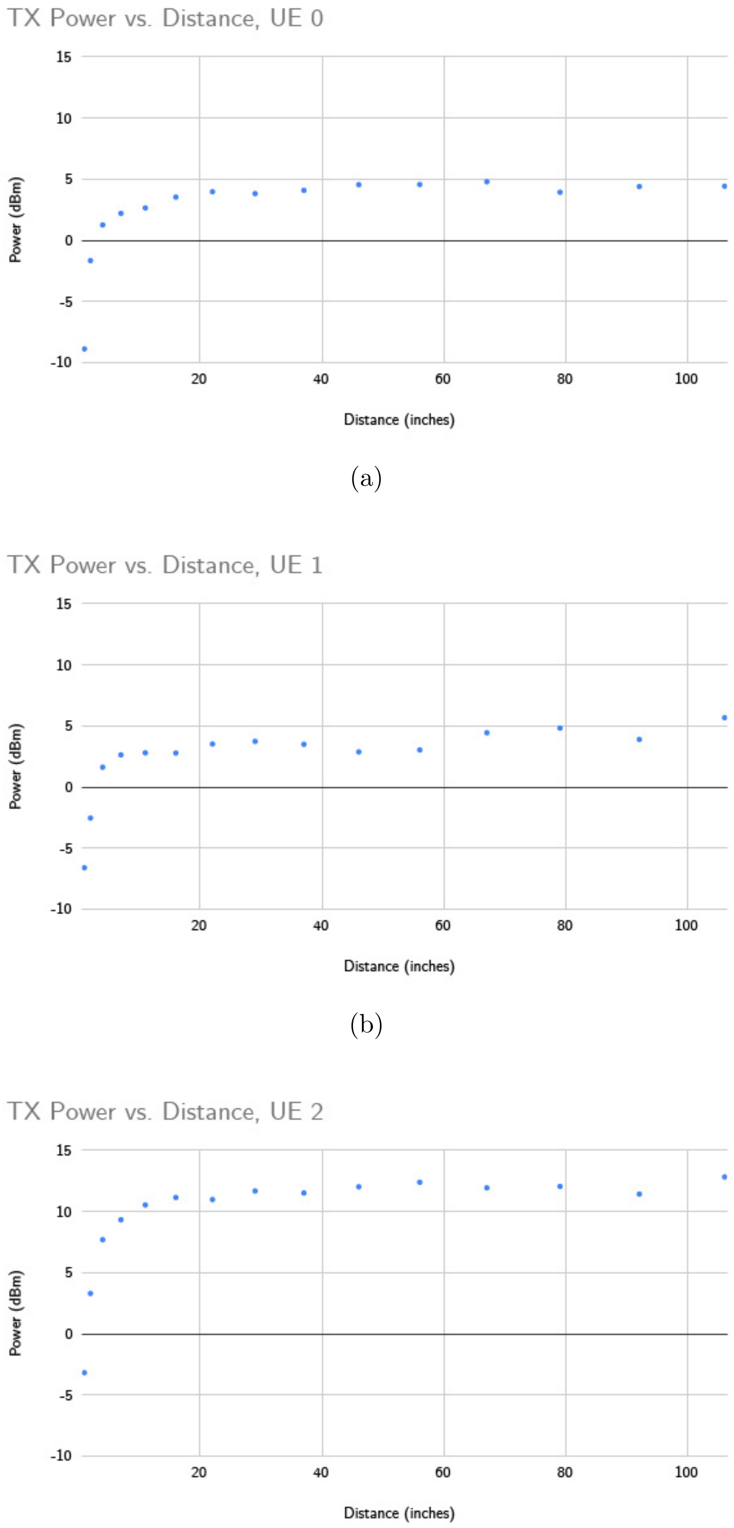
This experiment fixes a single UE at a distance of 50 inches and sweeps the UE azimuth angle with respect to the base station array center. Results are shown in Figure 5.7. Multiple measurements are taken, corresponding to multiple data points at each distance point. We observe a reduction of roughly 3db SINR at azimuth angles of +/- 60 degrees.

5.3.2 Dual-UE Measurements

We next measure a scenario with 2 UEs to demonstrate inter-user interference cancellation. UEs 0 and 1 from Figure 5.2 are used due to their similar transmit power. As with the single-UE scenario, we first measure channel estimation and constellations. Figure 5.8 shows channel estimation curves corresponding to the beam formed to each UE. As expected, we see a frequency-flat channel for the UE to which the beam is formed, and an attenuated, highly frequency-dependent curve for the opposite UE. Constellation plots are shown in Figure 5.9, with successful demodulation for a dual-UE system.

SINR with Angular Separation of UEs

This experiment fixes UE 2 at a distance of 90 inches, a height of 60 inches, and an angle of 0 degrees (broadside). UE 0 has its azimuth angle swept at a distance of 30 inches and height of 50 inches. The purpose of this experiment is to measure SINR as a function of angular separation between UEs, demonstrating the ability to de-correlate and remove inter-user interference. The results for this experiment are shown in Figure 5.10.



(a)

(b)

(c)

Figure 5.2: Plot of transmit power for three UEs, de-embedded from path loss.

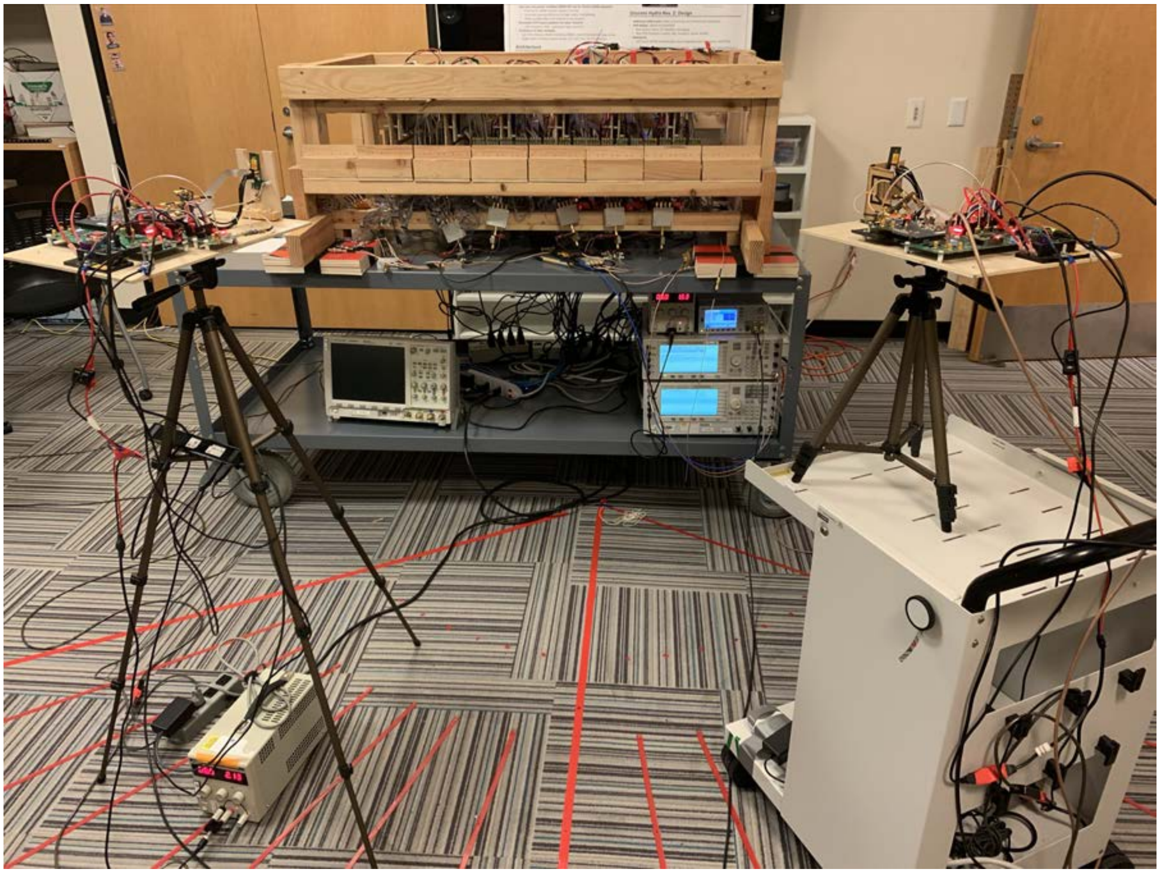


Figure 5.3: Configuration of base station and user equipment for measurement.

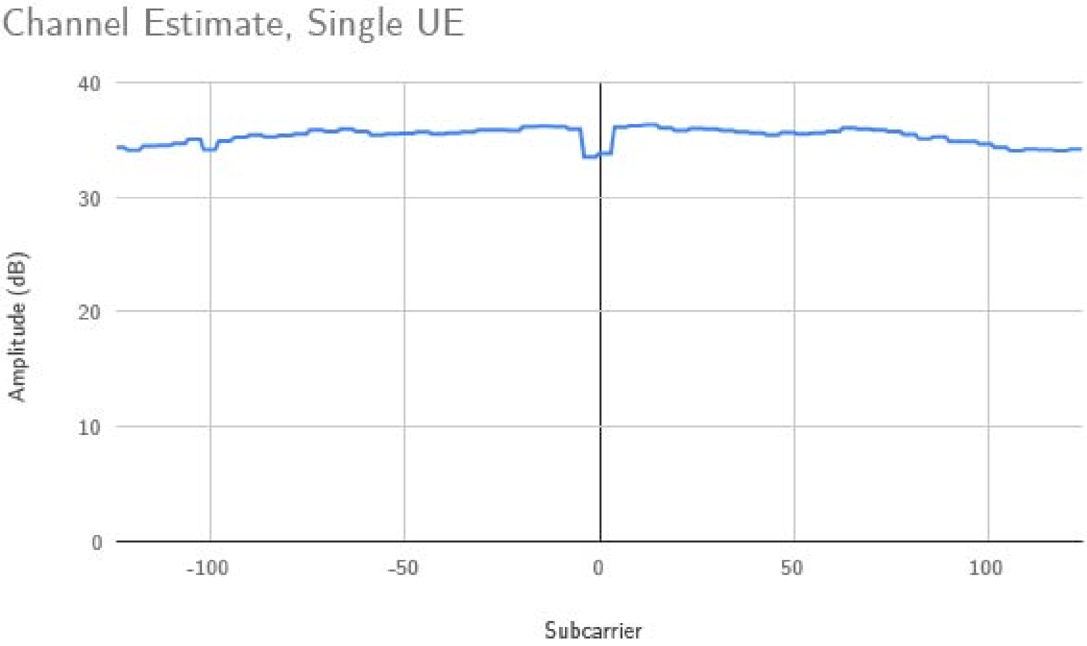
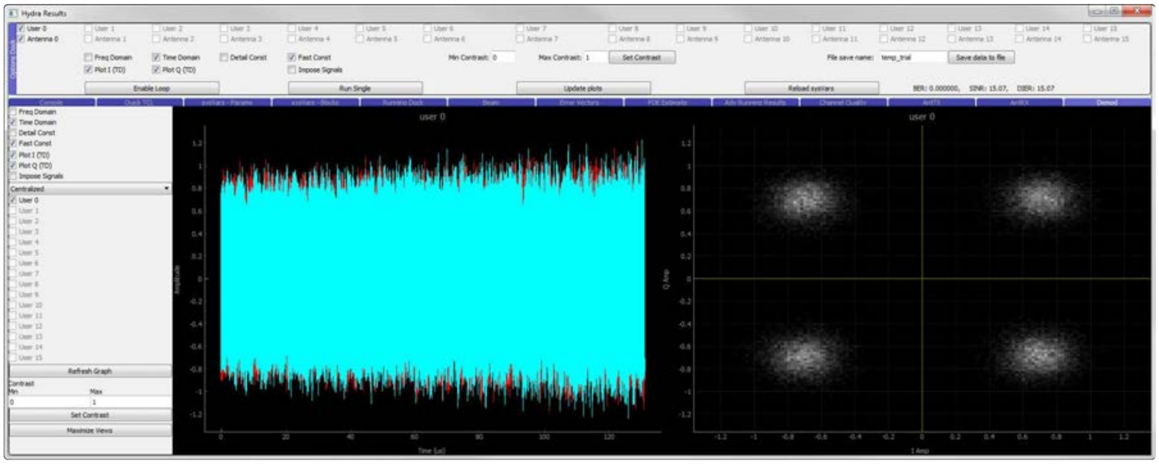


Figure 5.4: Channel response across 256 sub-carriers for single UE.


Hydra Results

☑ User 0 ☑ Antenna 0

□ Freq Domain ☑ Time Domain □ Detail Const ☑ Fast Const

☑ Plot I (TD) ☑ Plot Q (TD) □ Impose Signals

Min Contrast: 0 Max Contrast: 1

File save name: temp_trial

BER: 0.000000, SINR: 15.07, DIER: 15.07

Figure 5.5: Measured QPSK constellation plot of demodulated data, single UE.

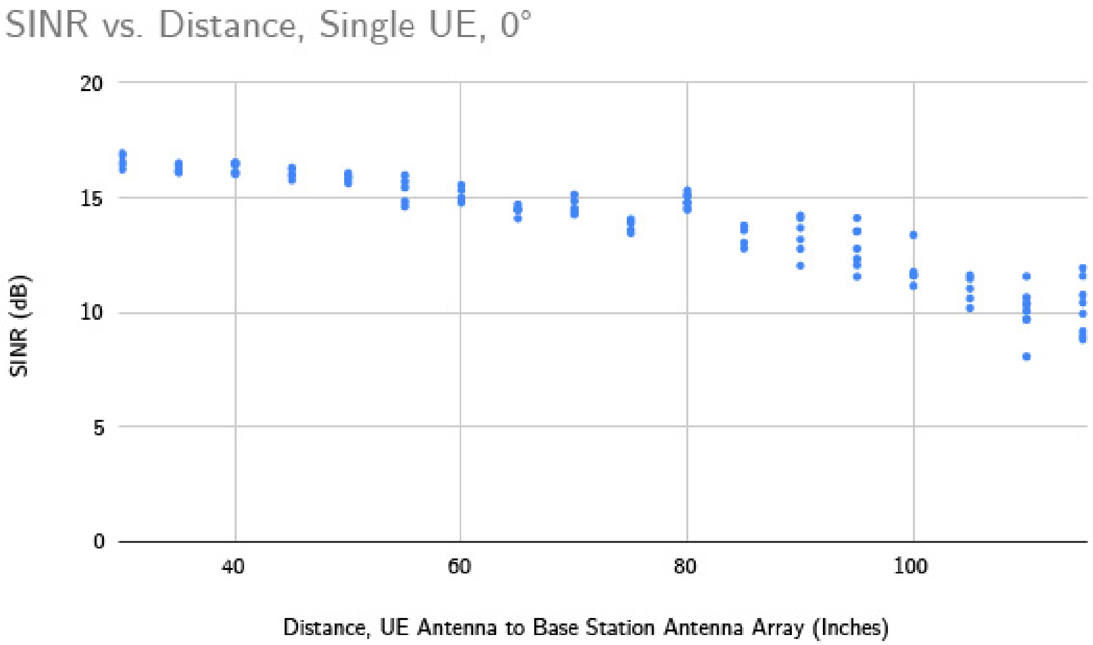


Figure 5.6: SINR vs. distance, single UE antenna to base station antenna array. Angle is 0 degrees (broadside).

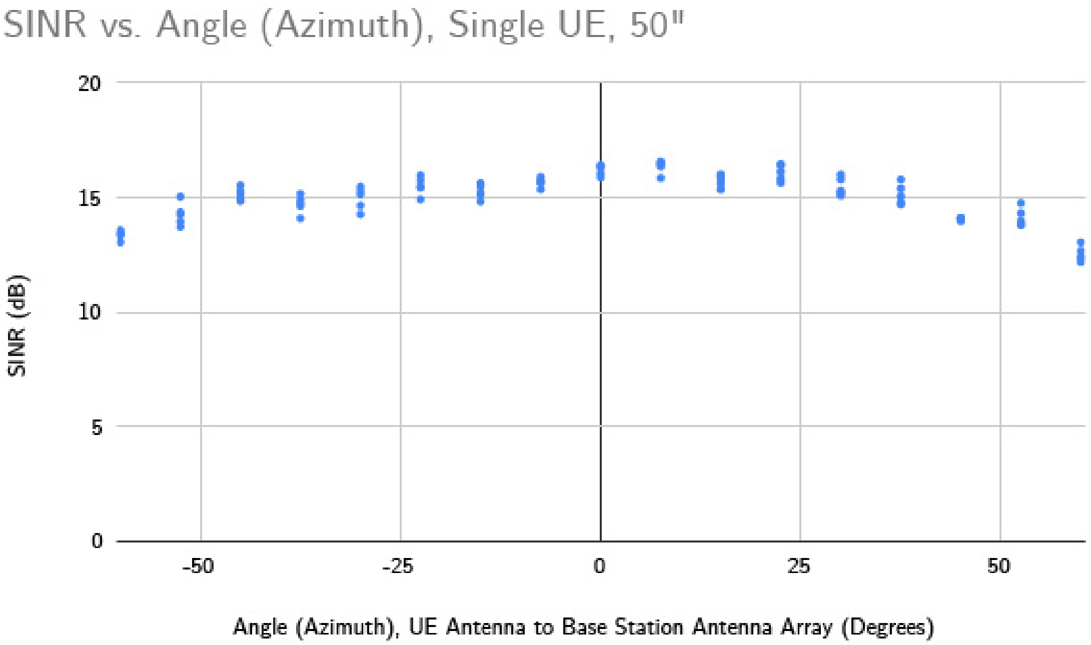
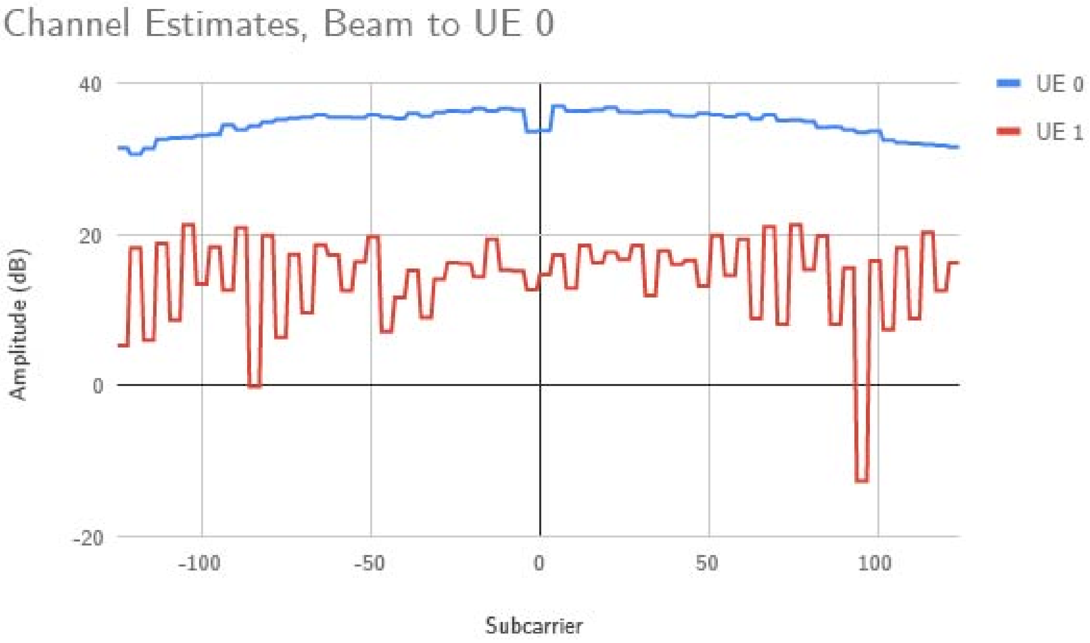
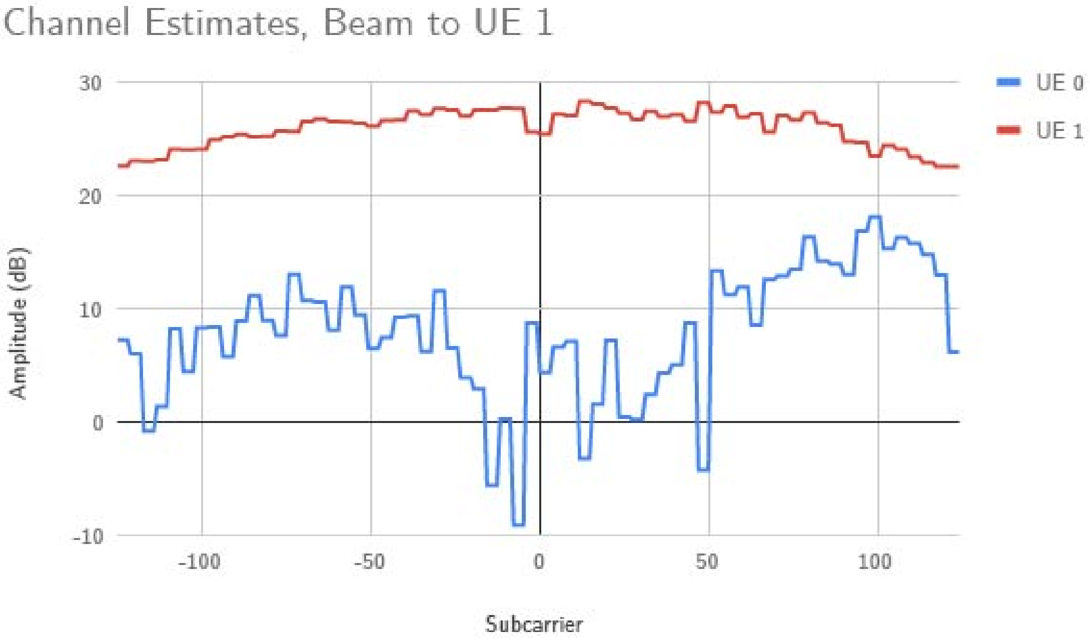


Figure 5.7: SINR vs. angle (azimuth), single UE antenna to base station antenna array. UE is fixed at distance of 50 inches from panel center.

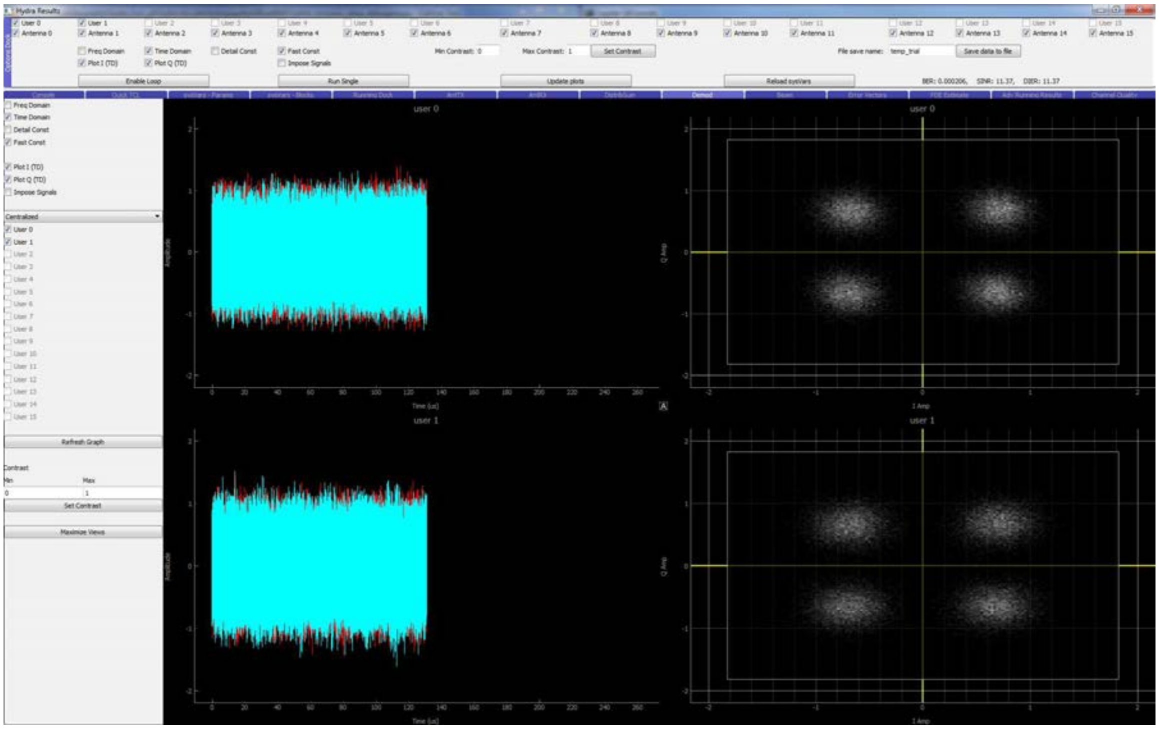


(a)



(b)

Figure 5.8: Channel response across 256 sub-carriers for dual UEs. a): For beam to UE 0. b): For beam to UE 1.



Hydra Results

☑ User 0 ☑ Antenna 0
☑ User 1 ☑ Antenna 1
□ User 2 ☑ Antenna 2
□ User 3 ☑ Antenna 3
□ User 4 ☑ Antenna 4
□ User 5 ☑ Antenna 5
□ User 6 ☑ Antenna 6
□ User 7 ☑ Antenna 7
□ User 8 ☑ Antenna 8
□ User 9 ☑ Antenna 9
□ User 10 ☑ Antenna 10
□ User 11 ☑ Antenna 11
□ User 12 ☑ Antenna 12
□ User 13 ☑ Antenna 13
□ User 14 ☑ Antenna 14
□ User 15 ☑ Antenna 15

□ Freq Domain ☑ Time Domain □ Detail Const ☑ Fast Const
☑ Plot I (TD) ☑ Plot Q (TD) □ Impose Signals

Min Contrast: 0 Max Contrast: 1

File save name: temp_trial

BER: 0.000206, SINR: 11.37, DBER: 11.37

Centralized
☑ User 0
☑ User 1

Contrast
Min 0 Max 1

Figure 5.9: Measured QPSK constellation plots of demodulated data, dual UEs.

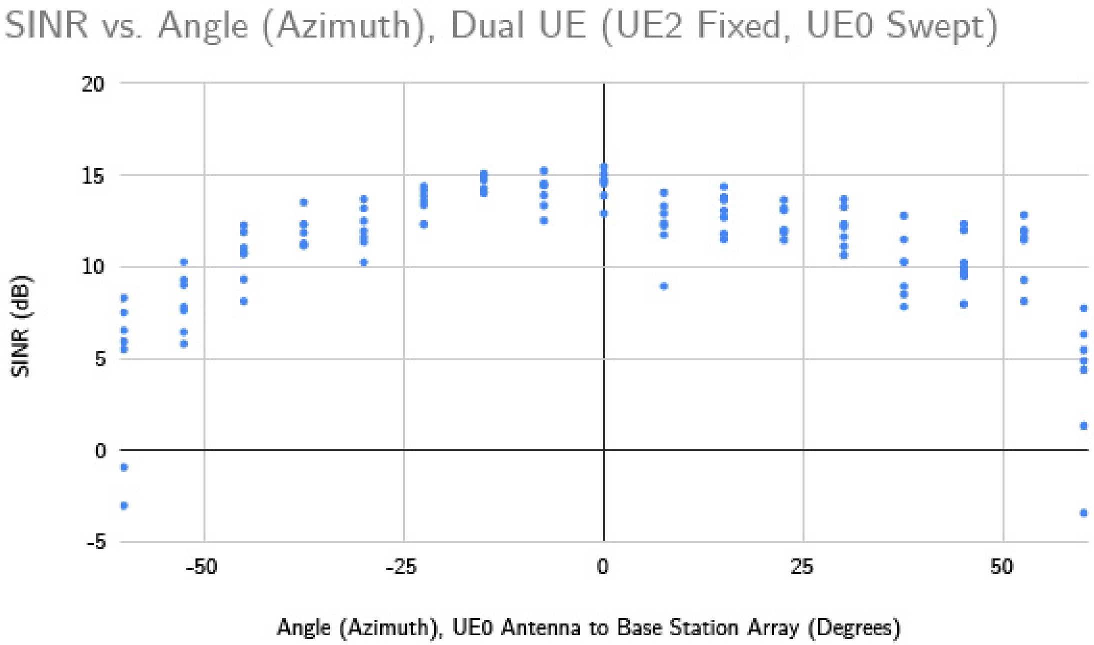


Figure 5.10: SINR vs. angular separation (azimuth) between two UEs.

Chapter 6

Conclusion

6.1 Thesis Contributions

This thesis has presented system design considerations and hardware implementation details for a scalable massive MIMO system operating at E-band frequencies, and demonstration of a base station consisting of sub-modules that perform fully distributed beam-forming. The modular design of the base station allows for flexible implementation with number of antenna arrays, which may be increased with addition of sub-modules to serve additional users. In an effort to make this system available as a prototype for further massive MIMO research in industry and academia, all components are commercially available and documented, and custom PCB design files are provided. Yield challenges associated with assembly are detailed, and remedies are discussed in the following section as guidance for those who attempt to construct a similar system. Functionality of the system is tested with 20 base station channels. SINR is measured with distance and angle for single-UE and angular separation for dual-UE configurations.

6.2 Future Work

We identify several areas in which this system may be improved in future revisions, as well as additional measurements that may be taken once these design improvements are made.

6.2.1 Reducing Cabled Connections to UE Modules

A current limitation for multi-UE setups is the amount of cabled connections that are needed between the UE and base station. Currently, a data frame synchronization connection needs to be made from the base station to each UE module. Additionally, separate JTAG and UART connections need to be made from the host PC to each UE module. This can be seen in Figure 3.1. Three cables per UE module running through the room proves difficult to manage even for dual-UE setups, and will cause cable management issues for future

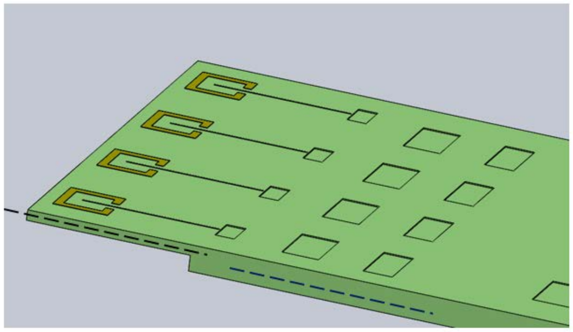


Figure 6.1: Render of future radio front-end PCB, showing increased laminate thickness. 3D design credit Anita Flynn.

experiments with 10 or more UEs. We first plan to implement a GPS-disciplined oscillator (GPSDO) module on each UE and the base station to perform wireless frame synchronization across the system. Additionally, we plan to implement UART and JTAG USB connections over Bluetooth.

6.2.2 System Robustness

As detailed in Section 5.2.1, significant yield issues arose in our radio front-end modules. Of modules that were successfully assembled, commercially available GaAs MMICs have delicate biasing requirements and significant thermal output, making these chips and thus the system fragile. We attribute this lack of yield and robustness to mechanical, electrical, and thermal design flaws, which will be addressed as follows.

Mechanical

The most significant yield loss among front-end radios was low LO gain. We traced this issue to a flexing of the front-end PCBs, which may change the controlled impedance of the PCB trace carrying the LO signal, or damage vias connecting the traces of different sides of the board. Additionally, physical protection of the MMICs from damage after wire bonding is limited. To address warping of the PCBs in a future revision, we intend to use a PCB with multiple thicknesses, in which most areas of the PCB laminate are thicker to prevent warping, and areas around the antenna are thin to achieve the desired EM characteristics for the micro-strip PCB antenna. Figure 6.1 shows a 3D render of the future PCB. To address protection of the wire-bonded chips-on-board, we plan to implement a 3D-printed cap and support structure on the PCB to fix the protection structure in place, shown in Figure 6.2.

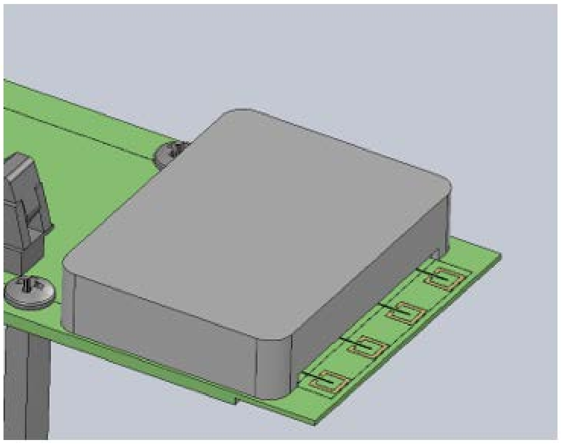


Figure 6.2: Render of wire-bond chip-on-board cover for future radio front-end PCB. 3D design credit Anita Flynn.

Thermal

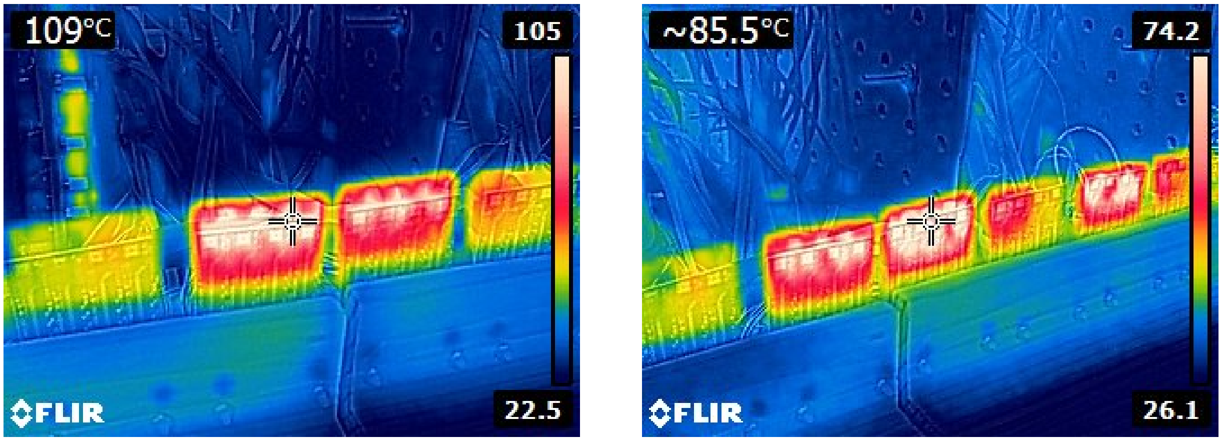
Heat dissipation for radio MMICs was not considered for this revision of the system. Heat has the potential to reduce longevity of the radio MMICs and cause warping of the PCBs due to thermal cycles on the laminate. As these are wire-bonded chips-on-board, heat sink application directly to the MMICs is infeasible. For the next revision of our radio front-end PCBs, we will design ground planes for heat dissipation to off-chip heat sinks elsewhere on the PCB. Additionally, we will construct a more robust air cooling solution. With a simple fan and shroud directing air onto bsae station radio MMICs, we were able to reduce IC temperatures by nearly 25 degrees Celcius (Figure 6.3).

6.2.3 Half-Duplex

The system presented in this work is up-link; the user equipment transmits and the base station receives data. Work is currently being done to add down-link functionality, so that user equipment may both transmit and receive in half-duplex operation for more practical demonstrations and investigations into channel reciprocity. At the hardware level, modifications include a redesigned radio front-end board with transmit and receive chains, and a transmit/receive switch. Additionally, a custom DAC PCB will be designed to accompany the ADC.

6.2.4 Use with Custom ASIC Modules

This work is constructed using COTS circuit components and custom PCBs, but its modular construction allows substitution of custom ASICs for commercial ICs. Once system-level



(a)

(b)

Figure 6.3: Temperature readings for radio MMICs. a): Before adding fan and shroud. b): After adding fan and shroud.

functionality is verified for the massive MIMO architecture presented in this work, it may be used as a test bed for custom components. We intend to substitute analog radio and digital baseband components with custom ASICs developed at Berkeley, which will reduce the system's power consumption and physical footprint significantly. A second revision of this work will include a more modular chassis for mounting custom radio front-ends, shown in Figure 6.4.

6.3 Design File and Code Availability

A repository containing PCB designs, bill of materials, code, and documentation is located at http://github.com/UCB_hydra/hydra_discrete.

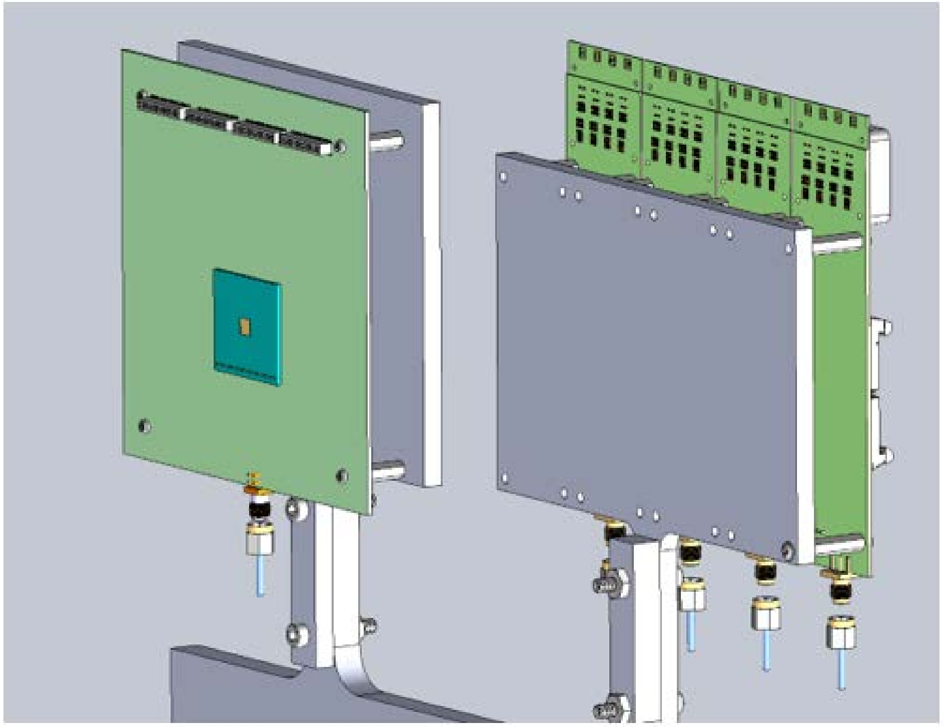


Figure 6.4: Modular chassis design for future revision of this system. Left: PCB housing custom radio ASIC. Right: Next revision of COTS radio PCB. 3D design credit Anita Flynn.

Bibliography

[1] Greg LaCaille. “TBD”. PhD thesis. UC Berkeley, 2017.

[2] L. Liang, W. Xu, and X. Dong. “Low-Complexity Hybrid Precoding in Massive Multiuser MIMO Systems”. In: *IEEE Wireless Communications Letters* 3.6 (Dec. 2014), pp. 653–656. ISSN: 2162-2345. DOI: 10.1109/LWC.2014.2363831.

[3] T. L. Marzetta. “How Much Training is Required for Multiuser Mimo?” In: *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*. Oct. 2006, pp. 359–363. DOI: 10.1109/ACSSC.2006.354768.

[4] T. L. Marzetta. “Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas”. In: *IEEE Transactions on Wireless Communications* 9.11 (Nov. 2010), pp. 3590–3600. ISSN: 1558-2248. DOI: 10.1109/TWC.2010.092810.091092.

[5] Thomas L. Marzetta et al. *Fundamentals of Massive MIMO*. Cambridge University Press, 2016. DOI: 10.1017/CBO9781316799895.

[6] Antonio Puglielli. “System Architecture and Signal Processing Techniques for Massive Multi-user Antenna Arrays”. PhD thesis. UC Berkeley, 2017.

[7] F. Rusek et al. “Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays”. In: *IEEE Signal Processing Magazine* 30.1 (Jan. 2013), pp. 40–60. ISSN: 1558-0792. DOI: 10.1109/MSP.2011.2178495.

[8] Clayton Shepard, Hang Yu, and Lin Zhong. “ArgosV2: a flexible many-antenna research platform”. In: Sept. 2013, pp. 163–166. DOI: 10.1145/2500423.2505302.

[9] Clayton Shepard et al. “Argos: Practical Many-Antenna Base Stations”. In: *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*. Mobicom ’12. Istanbul, Turkey: Association for Computing Machinery, 2012, pp. 53–64. ISBN: 9781450311595. DOI: 10.1145/2348543.2348553. URL: https://doi.org/10.1145/2348543.2348553.

[10] J. Vieira et al. “A flexible 100-antenna testbed for Massive MIMO”. In: *2014 IEEE Globecom Workshops (GC Wkshps)*. Dec. 2014, pp. 287–293. DOI: 10.1109/GLOCOMW.2014.7063446.

[11] B. Yang et al. “Digital Beamforming-Based Massive MIMO Transceiver for 5G Millimeter-Wave Communications”. In: *IEEE Transactions on Microwave Theory and Techniques* 66.7 (July 2018), pp. 3403–3418. ISSN: 1557-9670. DOI: 10.1109/TMTT.2018.2829702.

Appendix A

Cables and Connections

Cable Number	Cable End A Connector Type	Cable End B Connector Type	Cable Length	Cable End A Board, Location	Cable End B Board, Location	Purpose
1	SMA	SMA	12"	DAC Board, DACCLK	Breakout Board, UE_REF_DAC	DAC ref. clock
2	SMA	SMA	6"	DAC Board, IDAC	Balun Board, SMA5	In-phase DAC output
3	SMA	SMA	6"	DAC Board, QDAC	Balun Board, SMA1	In-quadrature DAC output
4	SMA, Right Angle	MMCX	12"	Balun Board, SMA2	mmWave Board, J10	I_N (double-check polarity)
5	SMA, Right Angle	MMCX	12"	Balun Board, SMA3	mmWave Board, J9	I_P (double-check polarity)
6	SMA, Right Angle	MMCX	12"	Balun Board, SMA6	mmWave Board, J6	Q_N (double-check polarity)
7	SMA, Right Angle	MMCX	12"	Balun Board, SMA7	mmWave Board, J8	Q_P (double-check polarity)
8	SMA, Right Angle	SMA	6"	LO Board, J1	mmWave Board, J7	mmWave LO in
9	SMA	SMA	12"	LO Board, SMA1	Breakout Board, UE_REF_PLL_N	LO PLL ref. N
10	SMA	SMA	12"	LO Board, SMA2	Breakout Board, UE_REF_PLL_P	LO PLL ref. P
11	2x1, 0.1"-pitch female header	2x1, 0.1"-pitch female header	12"	Power Board, UEMMW_5V_LO_BUFF	mmWave Board, P2	LO amplifier power
12	10x2, 0.05"-pitch female header	10x2, 0.05"-pitch female header	12"	Power Board, UEMMW_POWER	mmWave Board, P1	mMWave RF IC power
13	2x1, 0.1"-pitch female header	2x1, 0.1"-pitch female header	5"	Power Board, GND	LO Board, P4	LO GND
14	2x1, 0.1"-pitch female header	2x1, 0.1"-pitch female header	5"	Power Board, UELO_VDD	LO Board, P1	LO VDD
15	2x1, 0.1"-pitch female header	2x1, 0.1"-pitch female header	5"	Power Board, VCO	LO Board, P3	LO VCO variable voltage
16	10x1, 0.1"-pitch female header	10x1, 0.1"-pitch female header	5"	Power Board, P2	Breakout Board, P1	Power board enables (align signal names)
17	6x1, 0.1"-pitch female header	6x1, 0.1"-pitch female header	5"	LO Board, P2	Breakout Board, P7	LO I2C control (align signal names)
18	4x2, 0.1"-pitch female header	Terminal block spade crimp	10"	Breakout Board, 6V_UNREG	Terminal block connection	6V fused external power
19	4x2, 0.1"-pitch female header	Terminal block spade crimp	10"	Power Board, 6V_UNREG	Terminal block connection	6V fused external power
20	4x2, 0.1"-pitch female header	Washer crimp	10"	Breakout Board, GND	Common Ground Post	GND
21	4x2, 0.1"-pitch female header	Washer crimp	10"	Power Board, GND	Common Ground Post	GND
22	USB type A	Mini USB	6' (feet)	JTAG Module Board	USB Hub	JTAG programming
23	USB type A	3-pin UART	6' (feet)	Breakout Board, P2	USB Hub	UART progamming

Table A.1: List of cables and connections in one Hydra UE module.

Cable Number	Quantity	Cable End A Connector Type	Cable End B Connector Type	Cable Length	Cable End A Board, Location	Cable End B Board, Location	Purpose
1	4	SMA, Right Angle	SMA	8"	LO Distribution Splitter	mmWave Board, J1	LO in
2	4	2x2, 0.1"-pitch female header	2x2, 0.1"-pitch female header	22, 25, 28, 31"	Power Board, VGA_CTRL[1-4] (check orientation)	mmWave Board, P4 (check orientation)	mmWave LO in VGA control
3	4	4x1, 0.1"-pitch female header	4x1, 0.1"-pitch female header	22, 25, 28, 31"	Power Board, P13[1-4] (check orientation)	mmWave Board, P3 (check orientation)	mmWave VGA power
4	8	20x2 0.05"-pitch female header	20x2 0.05"-pitch female header	19, 22, 25, 28"	Power Board, [0-3][A-B] (check orientation)	mmWave Board, P1 and P2	mmWave IC power
5	4	17x2 0.05"-pitch male header	17x2 0.05"-pitch male header	18"	ADC Boards, U25 (check orientation)	mmWave Board, U2 (check orientation)	I/Q data
6	1	7x2 0.1"-pitch female header	7x2 0.1"-pitch female header	12"	Power Board, P1 (check orientation)	Breakout Board, P2 (check orientation)	Power Board IC I2C and enable
7	1	4x1, 0.1"-pitch female header	4x1, 0.1"-pitch female header	28"	FPGA Board, J53 (check orientation)	Frame Sync Board, any GPIO	Frame synchronization
8	4	SMA	SMA	18"	Breakout Board, CLK_IN/OUT	Daisy-chained Spine Slice's Breakout Board CLK_IN/OUT	Baseband clock distribution
9	4	4x2, 0.1"-pitch female header	Terminal block spade crimp	28"	ADC Boards, U6	6V fused external power	ADC external power
10	4	4x2, 0.1"-pitch female header	Washer crimp	28"	ADC Boards, U6	Common ground post	ADC external GND
11	1	4x2, 0.1"-pitch female header	Terminal block spade crimp	28"	Power Board, 6V_UNREG	6V fused external power	Power Board external power
12	1	4x2, 0.1"-pitch female header	Washer crimp	28"	Power Board, 6V_UNREG	Common ground post	Power Board external GND
13	1	2x1, 0.1"-pitch female header	Terminal block spade crimp	28"	Breakout Board, P1	6V fused external power	Breakout Board external power
14	1	2x1, 0.1"-pitch female header	Washer crimp	28"	Breakout Board, P5	Common ground post	Breakout Board external GND
15	1	USB type A	Micro USB	6' (feet)	FPGA Board, JTAG port	USB Hub	JTAG programming
16	1	QSFP	QSFP	3' (feet)	FPGA Board, QSFP port	Daisy-chained FPGA Board, QSFP port	Spine data transfer

Table A.2: List of cables and connections in one Hydra base station spine sub-module.

Appendix B

System Bringup Procedures

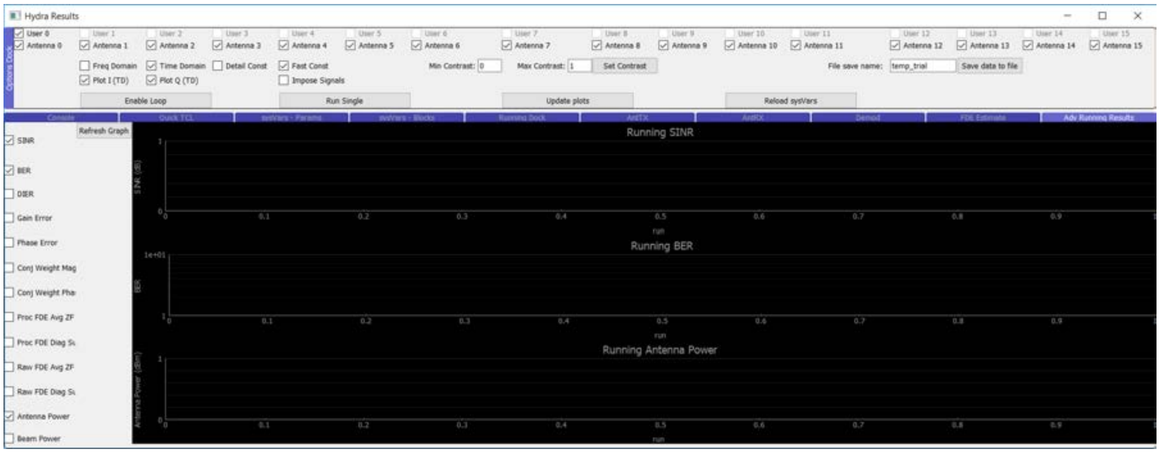
B.1 Python DSP Test

B.1.1 Software Configuration

- On the Python host PC, clone the repository given in Section 6.3
- Navigate to this cloned repository, and use text editor to edit the following file: `/testing/python/Python_DSP_Test.py`
- Modify the variable `repopathstring` to point to your repository location.
- Open an Anaconda prompt and execute `jupyter qtconsole`
- In this new console, execute `run [path to repo]/Python_DSP_Test.py`
- The window in Figure B.1 should appear. This is the Python DSP GUI environment. Parameters for the DSP can be found under the *sysVars - Params* tab.
- Once this window has loaded, click the *Run Single* button. After this, click the *Demod* tab. You should see an I/Q diagram as in Figure B.2.

B.1.2 Analyzing Results

The Python file that was run started the DSP GUI and generated a flat-fading Gaussian virtual channel with a preset channel noise. Clicking *Run Single* generated a single test vector, applied the Gaussian fading and preset noise of the virtual channel, "received" the samples after channel effects were applied, then applied Spine and Tail DSP to demodulate. Seeing clear constellation points indicates that the Python environment is functioning, and the user can move on to testing hardware.

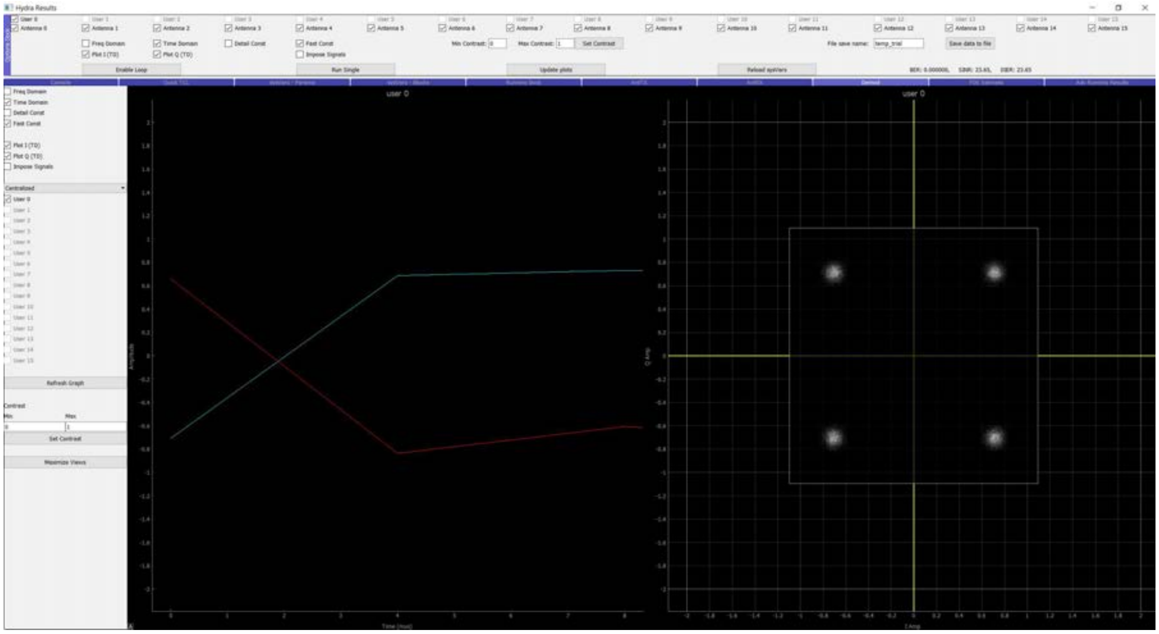


Min Contrast: 0

Max Contrast: 1

File save name: temp_trial

Figure B.1: Python DSP GUI environment.



Min Contrast: 0

Max Contrast: 1

File save name: temp_trial

BER: 0.000000, SINR: 23.65, DIER: 23.65

Figure B.2: Demodulated data over virtual link.

B.2 Up-link Test at Baseband

This test will exercise up-link of the PRBS from a single UE to a single BS panel at baseband, as well as transmission of the samples over Ethernet to the host and demodulation of the samples.

B.2.1 Hardware Configuration

Boards and Cables

A single UE hardware unit should be configured as shown in Figure 4.7 and wired as listed in Table A.1. A single BS spine slice (no head module is needed) should be configured as shown Figure 4.4 and wired as listed in Table A.2. Connect the Ethernet port of the Spine FPGA to either a network switch, or directly to the host PC. The JTAG modules of the BS Spine FPGA and UE FPGA should be connected to the host PC. The UART-to-USB adapter should be connected from the UE FPGA to the host. Additional components needed include:

- SMA-to-Tigereye PCB.
- 2x30 dB SMA attenuators, rated to at least 1GHz.
- 2 x SMA baluns, rated to at least 1GHz.

Connect the UE DAC's I and Q to attenuators, baluns, and the SMA-to-Tigereye PCB as shown in Figure B.3. Do not worry about accidentally reversing I and Q for now; this can be fixed in software in a future section. The Tigereye cable will be connected to the *Channel 0* ADC on the base station as shown in Figure B.4. Preserve orientation on the Tigereye cable by referencing to the marked *O* on the Tigereye connector footprint on both breakout board and ADC.

Power on the boards in the following order:

- UE (6V DC power supply to fuse box).
- Basestation FPGA (switch on Xilinx VCU118).
- Basestation slice (6v DC power supply to fuse box).

Instruments

A 500MHz clock source and 2-channel, 4KHz clock source are needed for this test. For the 4KHz clock source, connect channel 1 to the *FRAME SYNC* port of the UE Breakout Board, and channel 2 to the base station frame sync. Do not worry about manually configuring the waveform of the clock source; this will be done in software. On the rear of the instrument, ensure that GPIB is connected to the USB-to-GPIB adapter, and connect the rear port labeled *10MHz IN* to the UE Breakout Board's *10MHz_REF_OUT*. For the 500MHz clock source: connect output *Q* (non-inverting) to a balun, and connect the two outputs of the balun to the Spine breakout board's differential clock in. Important: place a 50-ohm terminator on the Q-bar (inverting) output of the 500MHz clock source. On the rear of the instrument, ensure that GPIB is connected to the USB-to-GPIB adapter. Connect the instrument's *10MHz INPUT* port to the 4KHz clock source's *10MHz OUT* port via a

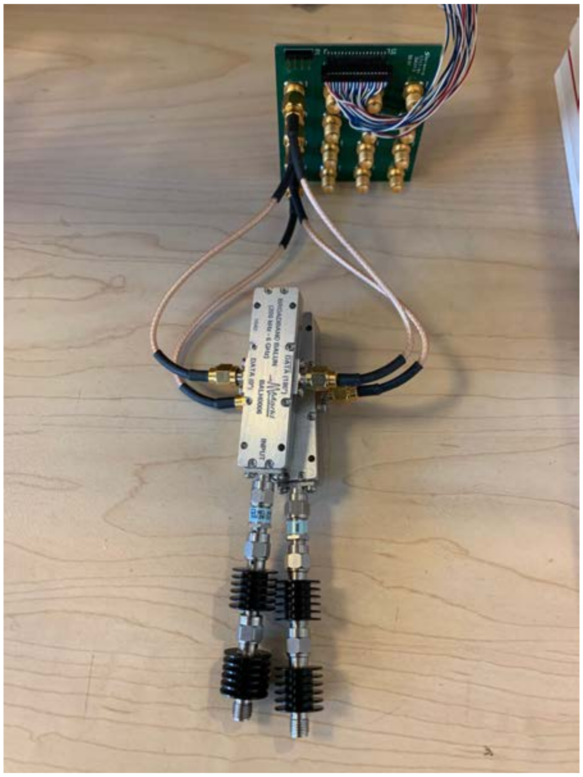


Figure B.3: Connecting UE DAC to Tigereye breakout.

BNC cable. Set the clock frequency to 500MHz, and the output level to +7dBm. Connect the GPIB-to-USB adapter that daisy-chains the two instruments to a host PC USB port.

B.2.2 Software Configuration

- Navigate to the repository, and use a text editor to edit the following file: `/testing/python/Uplink_Baseband_Test.py`
- Update the variable `repopathstring` to point to the repository location.
- Update the `sysVarsObj UEFPGAIDAliasString` with the UE number (0-15).
- Update the `sysVarsObj UARTIDString` with the COM port of the USB UART adapter.
- Update the `sysVarsObj BSFPGAIDAliasString` with the base station panel number (0-7).
- Update the argument `awgAddressList` to function `startExtFrameSync` with the GPIB address of the 4KHz clock source.

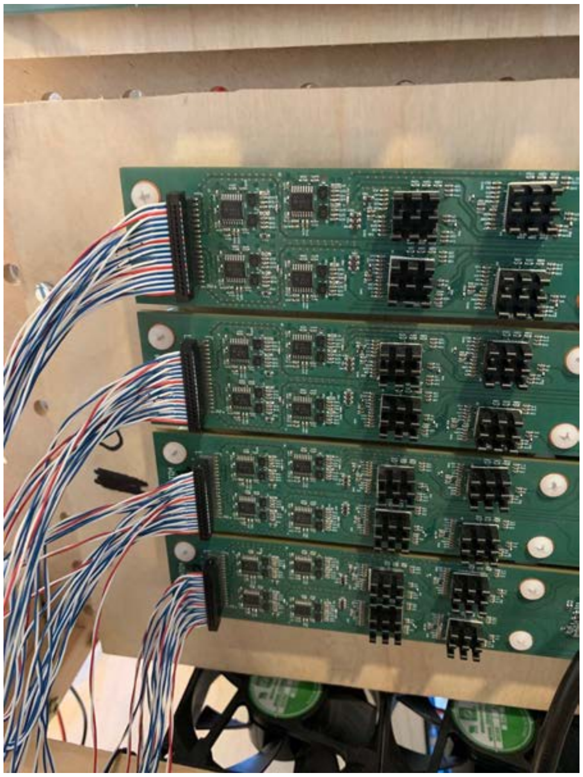
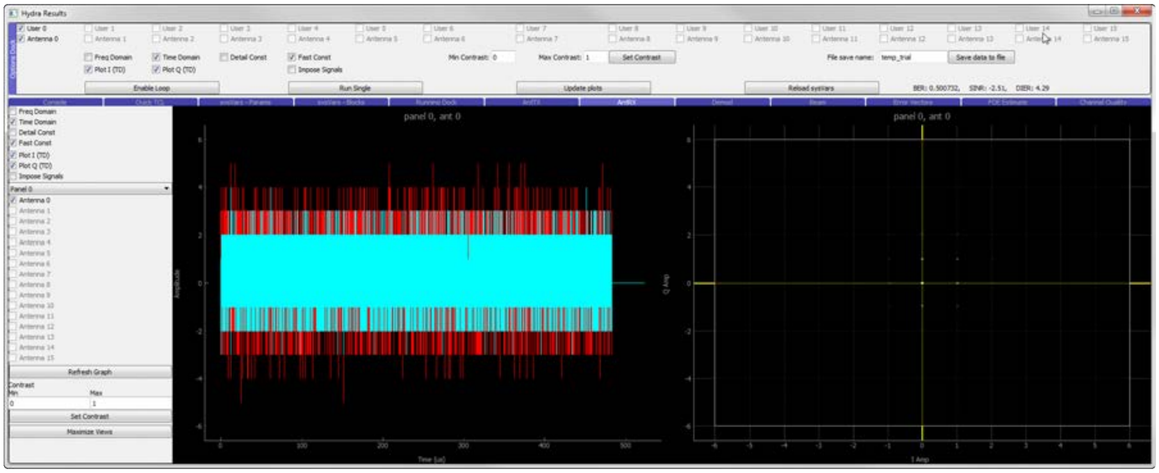


Figure B.4: Tigereye cable, ADC end.

- Open an Anaconda prompt and execute `jupyter qtconsole`. In this new console, execute `run Uplink_Baseband_Test.py`
- Press enter at the prompts until the first GUI window appears.
- Click the *Run Single* button once, wait for the GUI to process, then click on the *AntRX* tab. Click the *Maximize Views* button to scale the plot. There should be a noisy signal on Antenna 0, such as in Figure B.5. It will not contain signal because numerous parameters have not been set in the startup script. This simply verifies that the DAC and ADC are working and that some signal makes it to the host. Close this window and return to the iPython console.
- Continue pressing enter at the console to proceed with the startup script, until a second GUI window appears. Again, click the *Run Single* button, wait for the GUI to process, and click the *AntRX* tab. Click the *Maximize Views* button to scale the plot. There should be a signal similar to Figure B.6. If the plot looks similar but with a phase shift as in Figure B.7, this means that the frame synchronization phase offset between UE and base station must be adjusted. To adjust this offset, click on the *Quick TCL* tab. At the bottom right in the drop-down titled *Channel*, select AWG Ch1. Enter



☑ User 0 ☑ Antenna 0
□ User 1 … □ User 15; □ Antenna 1 … □ Antenna 15

□ Freq Domain ☑ Time Domain □ Detail Const ☑ Fast Const
☑ Plot I (TD) ☑ Plot Q (TD) □ Impose Signals

Min Contrast: 0 Max Contrast: 1

File save name: temp_trial

BER: 0.500732, SINR: -2.51, DIER: 4.29

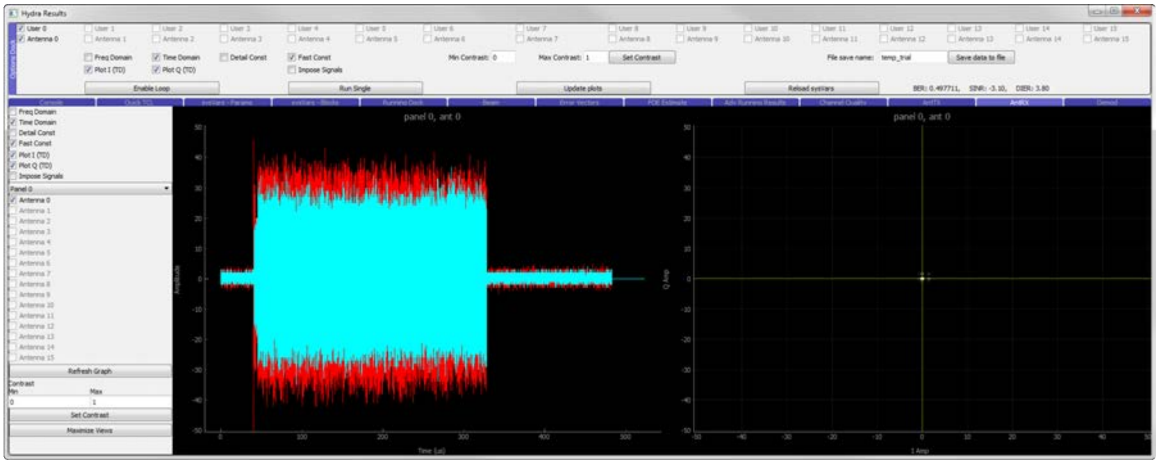
Figure B.5: Noisy signal prior to setting DSP parameters in bringup script.

the desired phase offset (180 degrees in this case) in the box labeled *Phase* and click *Frame Sync Phase*. Re-run *Run Single* and return to the *AntRX* tab. There should now be a signal that contains a centered data frame.

- Now that the proper signal has been received at the antenna, we may check the demodulation. Click on the GUI tab *Demod* and inspect the constellation plot. If there is a clear QPSK constellation as in Figure B.8, it means that baseband processing is correct. If, however, there is noise as in Figure B.9, the most likely cause is a swapped I and Q cable. Since Golay pilots are only transmitted on the I channel, if I is swapped with Q, the channel will not estimate properly. To fix this, go to the *Quick TCL* tab, tick the *Swap* option, and click *Swap IQ*. Re-run *Run Single* and inspect the *Demod* tab.

B.2.3 Analyzing Results

This test has demonstrated up-link at baseband with one UE and one base station channel. The constellation points correspond to the demodulated PRBS payload loaded into the UE FPGA. Golay and pilots were sent, which were used to estimate the channel. As a wired channel, we expect it to have high SNR and low BER. The GUI window shows a current reading of these metrics, shown toward the top-right of the GUI window in Figure B.8. If the user would like to save per-sample data to a file, there is an option to do so at the top-right of the GUI window.

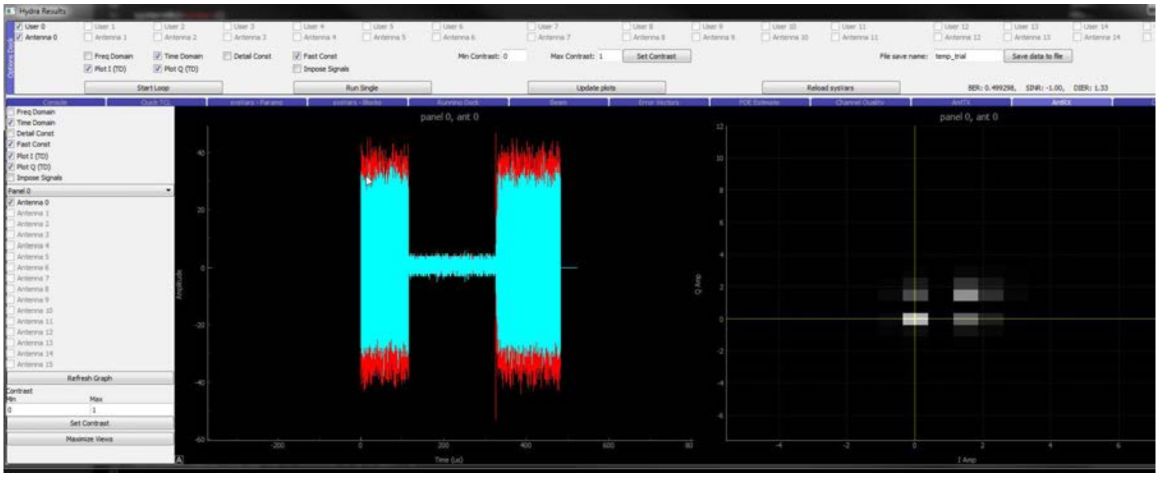


Min Contrast: 0 Max Contrast: 1

File save name: temp_trial

BER: 0.497711, SINR: -3.10, DIER: 3.80

Figure B.6: Centered data frame in time domain.



Min Contrast: 0 Max Contrast: 1

File save name: temp_trial

BER: 0.499298, SINR: -1.00, DIER: 1.33

Figure B.7: Phase offset data frame in time domain.

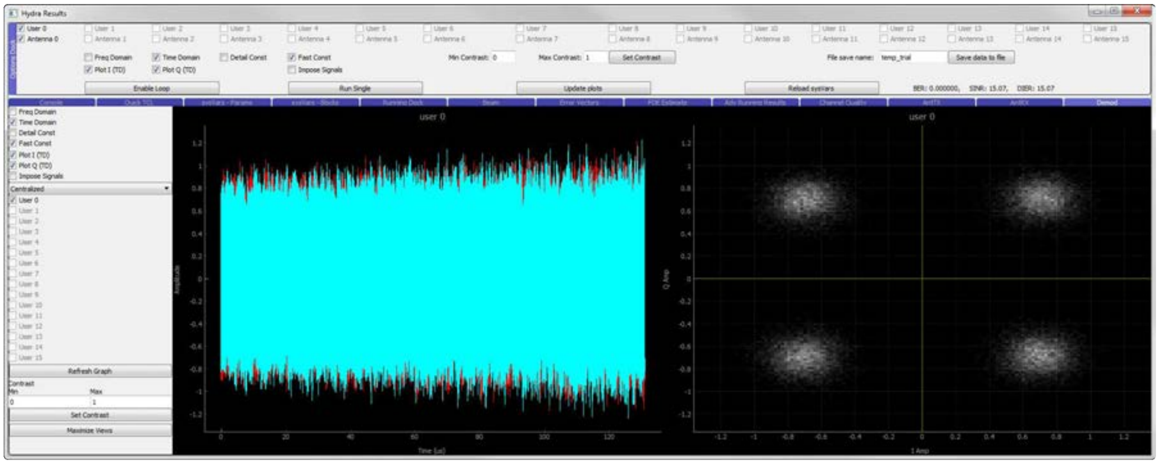
B.3 Up-link Test at Carrier

This test will exercise up-link of the PRBS from a single UE to multiple base station antennas at carrier frequency, utilizing the radio front-ends.

B.3.1 Hardware Configuration

Boards and Cables

The initial hardware configuration follows that of the baseband up-link test in Section B.2. However, more than one base station slice and UE may be used. Base station slices are con-



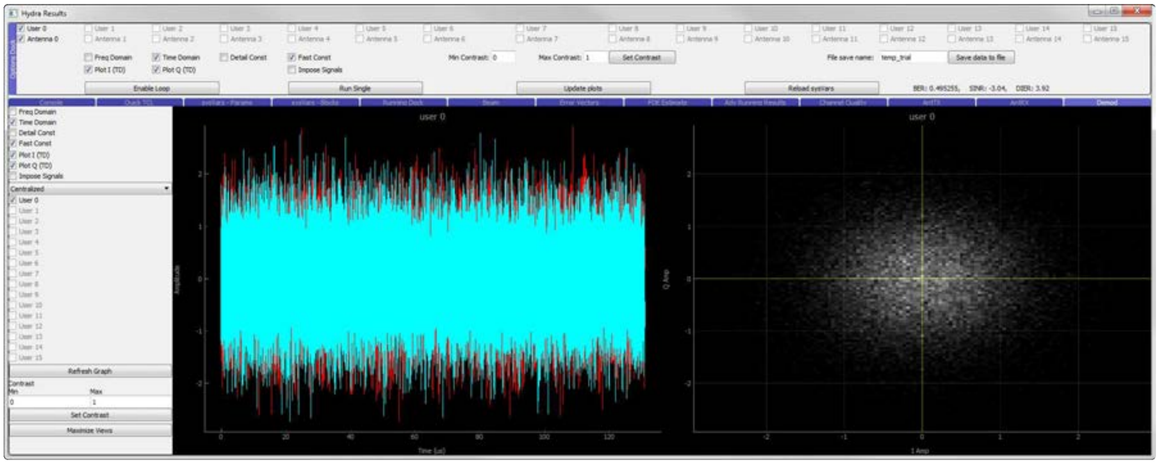
☑ Time Domain ☑ Fast Const ☑ Plot I (TD) ☑ Plot Q (TD)

Min Contrast: 0 Max Contrast: 1

File save name: temp_trial

BER: 0.000000, SINR: 15.07, DIER: 15.07

Figure B.8: Clear QPSK constellation.



☑ Time Domain ☑ Fast Const ☑ Plot I (TD) ☑ Plot Q (TD)

Min Contrast: 0 Max Contrast: 1

File save name: temp_trial

BER: 0.495255, SINR: -3.04, DIER: 3.92

Figure B.9: Issue with demodulation, such as a swapped I and Q cable.

figured in the chassis as shown in Figure 4.4 and wired per Table A.2. Instead of connecting the UE DAC coaxial cables and base station ADC Tigereye cables to the breakout board, they will be connected to the radio front-ends. The radio front-end connections for a single UE are given in Table A.1, columns 2-8, and are pictured in Figure 4.7. For a base station radio, connections are given in Table A.2, columns 1-5, and pictured in Figure 4.2 c).

Instruments

The instruments in Section B.2 are re-used in this test. Additionally, a 12.5GHz signal generator is needed to provide an LO input to the base station radio front-ends (heads). The LO signal may be routed to each head with a power splitter and amplifier. Additionally, the frame synchronization signal must be routed to each UE via an SMA splitter.

B.3.2 Software Configuration

- The file `/testing/python/Uplink_Baseband_Test.py` should be edited with the same configuration steps in Section B.2.2.
- More than one `UEFPGAIDAliasString` and `BSFPGAIDAliasString` value may be included, separated by commas.
- Update `sysVarsObj NumPanels` with the number of base station panels (slices) used in the test.
- Update `sysVarsObj NumUsers` with the number of UEs used in the test.
- Comment out any unused channels in the dictionary object `bringupDict`. This disables any radio channels that may fail during biasing.
- Open an Anaconda prompt and execute `jupyter qtconsole`. In this new console, execute `run Uplink_Carrier_Test.py`
- Follow the commands in this script to program FPGAs, bias radios, and start the baseband.
- When the first GUI window appears, base station channels may be disabled under the *Channel Quality* tab.
- After clicking *Run Single*, the *Demod* tab should show a constellation for each user in the system. An example of constellations for a 2 UE system are shown in Figure 5.9.

B.3.3 Analyzing Results

This test has demonstrated up-link over-the-air at carrier frequency. Like the baseband test, constellations show demodulated PRBS data from the UEs. Multiple base station antennas and UEs may be used to conduct full system tests.