Project Report
EE 493 Senior Design

The Seer

A system designed to estimate the direction-of-arrival of low-band 5G signals

By:

Evan Peelen
Tate Harsch-Hudspeth
Victor Madrid

May 2021

Faculty Advisor: Dr. Mohamed Salem, Sonoma State University
Industry Advisor: Joe Reid
Client: Cellular Providers, Cellular Application Developers

Project Website:
https://harschht.github.io/The-Seer/

Acknowledgments:

We would like to thank the entire Department of Engineering Science at Sonoma State University for their support over the years. A special thanks to our advisors Dr. Mohamed Salem and Joe Reid for their endless help and support throughout the senior design process. We would also like to thank Dr. Donald Estreich, Andrew Choi, Roger Nichols, Stan Bischof, and Enrique Zeiger for their time and guidance. Lastly, we would like to thank the Office of Research & Sponsored Programs and the Koret Foundation for funding our project.

Abstract

The Seer focuses on streamlining the process of location estimation to benefit telecommunications, assisted GPS (AGPS), and radio enthusiasts. By implementing a neural network that utilizes the signal data from a receiver (Rx) composed of an array of antennas, our system creates an accurate and adaptably complex model of the environment it is trained in. The system focuses on Sub-6 5G NR within the 600 MHz (n5) to 850 MHz (n71) band. Positioning the antennas a half-wavelength apart ensures that the main variation between antennas is due to the antennas' orientation relative to the transmitter (Tx). Analyzing the amplitude and phase of these received signals provides useful data for the creation of our deep learning model. Use of a neural network allowed a model to be created that matched the complexity of the urban indoor environment our team targeted with our prototype. Other methods of pinpointing the location of an incoming signal use models that assume an isotropic environment, while true indoor urban environments are by no means isotropic. The existence of other EM waves, as well as multipath, and constructive and destructive interference add to the complexity of solving the inverse function of finding the direction using the received signal parameters. Our system has the added benefit of learning and can be implemented in any environment through training. Our system prototype has the potential of improving the current methods for determining the direction-of-arrival (DOA) of low-band 5G signals, as well as finding the range of transmitting devices, bolstering communication between a base station and Tx, while lowering the economic impact of these large scale 5G systems.

Table of Contents

List of Figures:

List of Tables:

Problem Statement

Large cellular companies spend up to 70% of their capital on power bills, that is more than they spend on employee salaries. Dynamic and static beam forming attempts to reduce the unnecessary use of power in areas covered by antennas that may not always need coverage. One way to tune these antennas is to keep them pointed on areas that gather the most users, but it is not always so cut and dry. Some places have large crowds in one area during the day, but the crowds gather in different areas at night. Dealing with this can be cumbersome, costing money in the form of labor and power use. This could be simplified if there was a way to determine Tx location based on a single incoming pilot signal. This technology could also benefit smart device applications that rely on precise location for their services (AGPS). There are methods currently available that can model RF environments, but require the environment to be isotropic, which is usually not the case when dealing with urban areas.

Current methods of transmission are centralized around massive multiple-input multiple-output (MIMO) technology [4] that implements beamforming to send data into an area, rather than radiating omnidirectionally with umbrella coverage. Beamforming is imperative when it comes to the new 5G system because the wavelengths are short, impairing their ability to penetrate surfaces. For beamforming networks to work, the system must know where the recipient of the data is, so it can appropriately adjust the direction and intensity of the radiation. If the system were to incorrectly estimate the location of the recipient, the result would be increased power loss along with a reduction in customer satisfaction. The current methods of triangulation and periodic pilot signals used to determine the best Tx/Rx path for beamforming use considerable amounts of power and do not account for non isotropic environments. Complex environments are difficult to mathematically model due to the non-proportionality of power relative to distance, leading to our complex inverse problem.

Introduction

To model a cell tower and this transmission process, software defined radio technologies including the HackRF One, RTL-SDR, and GNU Radio will be interfaced together [11]. The HackRF One will act as the transmitting cell phone, while an array of RTL-SDRs connected to 5G dipole antennas will act as the base station receiving antenna array. Additionally, a programmed neural network [5] will process the received data from a database of collected measurements, creating a model of the environment it is trained in. After training, the system is tested on new data, never before seen by the neural network, to test its ability to make predictions using unseen data. Once the accuracy is verified, a graphical user interface (GUI) effectively outputs the direction-of-arrival of the transmitted signal. Tying both the software defined radio (SDR) hardware and deep learning software together, our team has developed a system that benefits cellular providers who want more efficient telecommunication by reducing power loss as well as capital loss, increasing overall transmission efficiency and service satisfaction.

Literature Review & Previous Works

In [1], George Godby from Michigan State University presented a comprehensive report for his ECE 480 Senior Design project: Using GNU Radio for Signal Phase Measurements. His report lays out his process and the interpretations of the data resulting from using GNU Radio to process an incoming signal on two separate antennas, and then finding the relative phase difference between them. He states in his introduction that the motivation behind this topic is based on the fact that signal phase information may be used for finding signal direction and angle of arrival. This is a key step in our project process for The Seer, making Godby's report an important resource for the team. Using two antennas spaced sufficiently apart for the target wavelength, SDR to receive the signal, and a software flowgraph developed to extract the phase information from the two antennas; Godby was able to show a direct correlation between the antennas position and the recorded phase difference as it relates to time. The report is well-structured and presents a step by step account of the project, first discussing the project flowgraph with a high level overview followed by a more detailed breakdown of what each component in the flowgraph does along with its specified parameters. The process is straightforward: break the signal up into bins using an FFT, capture the bin of interest where the signal peaks, and then subtract this value from the reference antenna's value to obtain the relative phase difference between the received signals. Before the received signal enters the FFT it is broken up into vectors, and after the FFT it is pieced together back into a stream. Godby presents some helpful tips on how to calculate bin size for various sampling rates, as well as his thought process in choosing an FFT size. Our team was able to choose parameters that fit our project by adjusting these calculations to fit our specified frequency and sampling rate. Once this relative phase difference was calculated, Godby used Python, specifically the NumPy library from SciPy, to extract the data from the binary data file produced by GNU Radio's File Sink block.

This process is a key part to our project, The Seer, and we found Godby's report to be a helpful starting point, and proof of concept, for finding the relative phase difference between two receiving antennas. His work also helped to confirm that this relative phase difference changes according to a change in position, an important factor in the success of our neural network. Using his work, our team was able to adapt his flowgraph to fit our purpose, adding true RF power in addition to the phase extraction for each antenna, and updating the FFT size, bin of interest, and sampling rate to work with our hardware and target signal frequency. We also switched out his UHD: USRP Source block for the OsmoSDR: Osmocom Source block, as well as introducing a Moving Average block and a Head block between the final bin selection and the File Sink. The Moving Average was added to compute a moving average over 1000 samples as a way to reduce the variance between measurements, while the head block leading into the moving average would limit the number of overall samples that enter the Moving Average block, this allowed us to specify the amount of time that our receiver flowgraph collected samples to be passed to the File Sink. By implementing a function in our data extraction Python code that pulled out the last moving average value collected by the File Sink, we were able to produce a consistent measurement that would only vary according to distance, position, and other variables making up the propagation environment.

In [9], Jason Brownlee PhD gave a thorough introduction into deep learning with Keras, a "powerful and easy-to-use free open source Python library for developing and evaluating deep learning models". Covering topics from loading data all the way to making predictions, Brownlee laid a solid and complete foundation for implementing your first neural network in Python. Included in this introduction to machine learning, he touched on helpful tools like SciPy and sci-kit learn which can be used to boost the efficiency and simplicity of your deep learning project. Keras is a deep learning library that can run on backends like Theano and TensorFlow, and allows for a complex deep learning model to be established with relatively short code. This is possible thanks to the large and powerful library of Keras functions. These functions make nontrivial tasks such as preprocessing data, compiling a model, developing layers, fitting the model, training the model, and using the model to make predictions more manageable so you can focus on the efficiency and accuracy of your model without getting lost in the network of powerful software that's running underneath the hood. Keras is not only helpful to beginners, as each aforementioned step can be broken down into multiple lines of code, allowing for flexibility and customization of your deep learning project. This makes Keras a powerful tool both for beginners and advanced machine learning specialists. Your code can be as complex as you want it to be while still offering Kera’s organizational benefits so that training, validation testing, loss/accuracy analysis, and predictions can be processed quickly and efficiently.

Aside from the helpful tips on how to parse CSV files for Keras inputs, normalizing data, the difference between a batch and an epoch, and defining then compiling your Keras model, our team found the most helpful information in his blog to be the metrics to use when choosing important functions like the loss (or cost) function, the optimizer, and the activation function. The choice of these functions is vitally important to a deep learning model and varies based on the type of problem the model is expected to solve, the number of outputs, and the complexity of the model. Brownlee had an example for a single-output binary classification problem for readers to follow along and try out on their own. Our team's project, The Seer, was implementing a multi-output regression problem which required different metrics. Brownlee had links to other articles from his blog like "How to Choose Loss Functions When Training Deep Learning Neural Networks" and "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning" that had helpful tips on how to choose metrics for different types of problems. There was no example of a multi-output regression problem but using the tips and tricks provided by Brownlee and other online sources, the team was able to piece together a model to fit our needs. Refer to the System Results (pg.53) for an overview of our Keras model, and the NN flowgraph at the end of System Architecture (pg.19) to see the metrics we used to satisfy our multi-output regression problem.

Methodology

To try and solve this problem, our team planned to use an array of five antennas spaced out a half wavelength from one another. Since our target frequencies are in the low-band 5G range (600-850MHz), after analyzing our antenna’s S11 parameters using a vector network analyzer (VNA), we chose 750MHz resulting in a spacing of 0.2 meters. Using five antennas in a linear horizontal array, we planned to use the varying amplitude and phase shift of the signal received by each antenna to determine the location of the Tx. In order for the antennas to provide coherent signal data, the team took steps to synchronize the RTL-SDRs by daisy chaining the clocks together via solder bridging and removing the bypass resistors on the physical components. To make sure that the clock signals were not disrupted by this process, we used an oscilloscope [14] to check the period and amplitude of the common clock after every addition to the array. The final product would include one master clock chained to four puppets with negligible change in clock period. With the clocks synchronized, the received power and individual phase shifts could be processed by GNU Radio [11], a software capable of processing the data received by the RTL-SDRs. To ensure that the data being collected would be useful to the neural network, we planned to use the middle antenna as a reference, dividing the phasors from each of the other four antennas by the reference antenna's phase to obtain the relative phase difference between them. Using GNU Radio, we would extract the magnitude and relative phase difference from each of the outer four antennas using FFT signal processing techniques, taking a moving average of the samples before saving the binary data to be processed by a data extraction code written in Python. Using this data extraction code, we would obtain the last moving average value from each of the four antenna's magnitude and relative phase difference binary data files, which were arranged into a ten-column tensor for the neural network to use for training. The last two columns of this tensor were the (r,theta) values that related to the distance between the Tx and the reference antenna on the Rx. To sample the testing environment, we would use a Latin Hypercube Sampling (LHS) [18] library in Python to produce an unbiased distribution of possible Tx locations. After compiling a sufficient number of data measurements, the neural network would be trained, and the loss and accuracy would be analyzed. Prior to training, the team would simulate data using two different radio wave propagation models: the Friis transmission equation, and the JTC urban indoor environment model. This simulated data allowed the team to dial in the number of hidden layers and weights per hidden layer in preparation for the live data. After the live data was collected and used for training, and an acceptable accuracy was found, the model and corresponding weights would be saved, then used for predictions on live measurements that were outside of the training dataset. The team developed a graphical user interface (GUI) to display the predicted value to the user, along with a polar plot representing the DOA of the received signal.

For the hardware, we used TG.35.8113 Apex II Wideband 5G/4G Dipole Terminal Antennas connected to RTL-SDRs making up the Rx, with a HackRF One attached to the Port-A-Pack used as the Tx. The antenna array would be put together via PVC pipe to mitigate any possible reflection or attenuation from the stand itself, since PVC is practically transparent at our target frequencies. We used SMA low-loss coax cables as extensions from the SDRs to the antennas in order to account for the half wavelength spacing. The SDRs were then connected to a USB hub powered with an external power supply connected to a standard 120V outlet, which sent data to an RPi 4 System on Chip (SoC). The RPi ran the team's GNU Radio flowgraph which sent the collected data to a database on the cloud. Once in the cloud, the data could be accessed by the data extraction code, which prepared the data for the neural network. The neural network was implemented in Keras [9], a deep learning API that uses TensorFlow2 as a backend. We used Tkinter and Matplotlib Python libraries [15] for the GUI.

Due to complications with the hardware and a lack of testing equipment availability resulting from the COVID-19 pandemic, the final system consisted of three SDRs connected to three antennas, with only power measurements used for data. We used one of the RTL-SDRs as a master clock to drive the clocks on the other four puppets. RTL admits to potential phase drifting occurring in their devices when operated over a few tens of MHz, this drifting disrupted the phase data the team was hoping to use for the DOA. Please see Test 5 (pg.45) for an in-depth look at the reason the team stepped down from five antennas to three, Test 6B (pg.49) for the reasoning behind dropping the relative phase difference from the input data, and Future of The Seer (pg.54) for a design that would compensate for these encountered issues.

Anticipated Risks

The following is a compilation of risks that the team anticipated early in the design stage of our project. From a hardware perspective, the uncontrolled RF testing environment will make testing hardware difficult. Our computers have limited USB ports, so a USB hub will be necessary. The low-band 5G dipole antennas do not come with SMA extensions for antenna spacing, so low loss coax cables will be required. Signal degradation due to multiple connections and transmission lines, or noise generated by loose connections on the SDR clocks could lead to offsets in our collected phase data, which could lead to uncertainties in the inputs to the neural network. From a software perspective, there is a chance we will overfit our data, resulting in less accurate predictions. We may overlook environmental factors and fail to create a complex enough model to make accurate predictions. If our data extraction flowgraph is not representative of the true RF power and relative phase difference of the received signal, our neural network will not be able to make accurate predictions.

Challenges

Following are both hardware and software challenges proven to be obstacles requiring contingency and mitigation plans to successfully complete our proposed senior design project. Beginning with hardware challenges, the team encountered limited lab equipment access and delayed funding availability due to the impact of COVID-19. CPUs built before 2011 do not support Advanced Vector Extensions (AVX), and consequently cannot support TensorFlow2 and Keras. The SDRs are power hungry devices, and our CPU hardware was unable to drive all 5 SDRs simultaneously, because of this, our antenna array design needed to be adjusted. Phase drift (time delay) at the Rx caused by synchronizing the SDR clocks and overheating components disrupted the sampling rate, affecting the credibility of our phase data. Our third party HackRF One had a faulty mixer which caused the center transmitting frequency to be offset by 40 kHz at 750 MHz, with the offset increasing at higher frequencies and decreasing at lower frequencies. The RTL-SDR used for initial testing was damaged due to prolonged exposure to high input power resulting in unwanted frequency shift keying, introducing uncertainties into our data analysis. Onto the major software challenges the team faced, most of the documentation regarding Keras and GNU Radio is geared towards specific applications, requiring us to build our own models. GNU Radio does not contain all libraries in the default download, so additional libraries needed to be found and installed. Open-source software documentation was limited, requiring additional testing and research.

Further elaborating on the hardware challenges, the third party HackRF One's faulty mixer amplified the effect of interfering frequencies that were trailing and leading the signal by 400 KHz, as seen below.

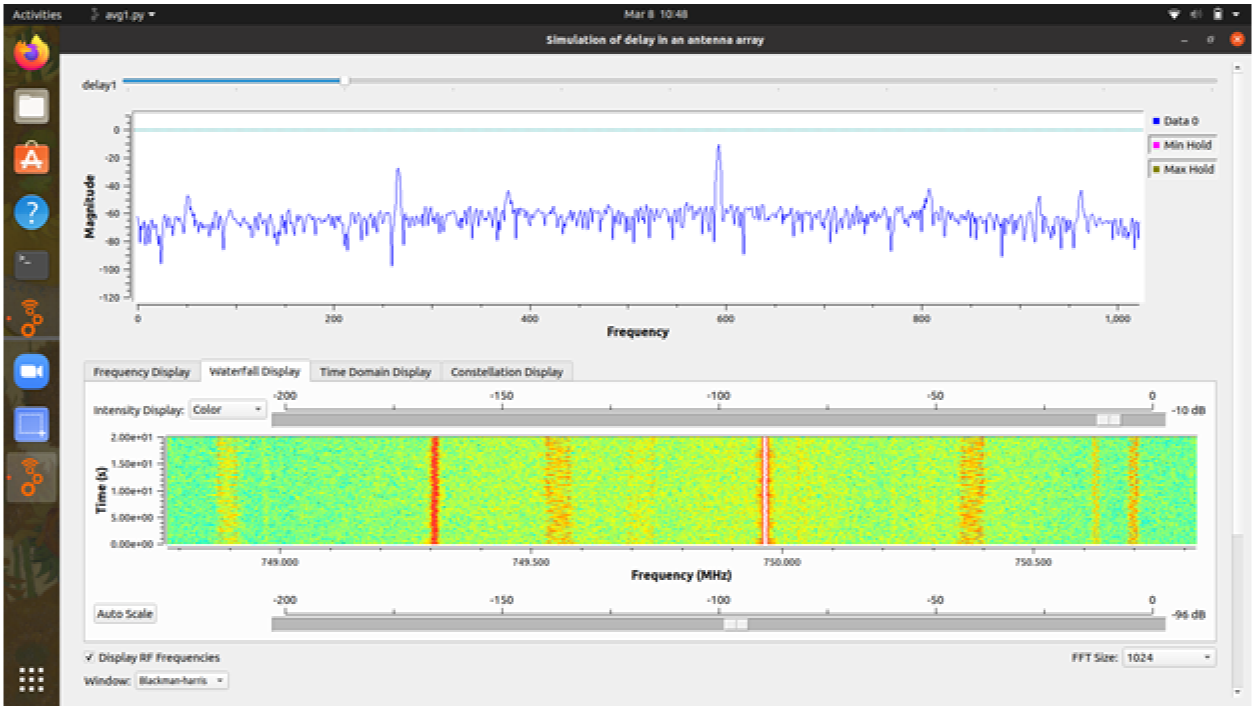

Fig. 1 - Interfering Frequencies

The cause of this issue was damage the HackRF One caused to the RTL-SDR used for initial testing from having the gain set too high during transmission tests. This high gain did not only affect the HackRF One, disrupting its already faulty mixer, it also degraded the accuracy and capability of our SDR.

Our team encountered a few components that did not live up to the product descriptions. We made sure to test each component before adding it to our system. A faulty, damaged, or low-quality product would introduce unknowns that could potentially compromise the quality of our system. After purchasing some SMA Coax Cables from eBay, the team conducted tests using a VNA to determine the S11 parameters of the component. We learned that the return loss was not only higher than expected, but there were oscillations present indicative of a low-quality cable. The cable was reflecting enough of the signal to cause a standing wave to appear along the transmission line, this was a problem for The Seer, as a standing wave within our SMA cable could disrupt the phase data being sent from the antennas to the RTL-SDRs, compromising data that is key to our neural networks ability to make accurate predictions. The plots associated with these cables can be found in the Test 3 section starting on page 32. After purchasing a new set of cables from a more reputable seller we found more reasonable figures after testing. The return loss was slightly below 0 dB at an open circuit and dipped below -15 dB from about 725 MHz to 775 MHz when connected to our TG.35.8113 5G antenna.

We also experienced the downside of supply and demand when our 5G antenna (TG.45.8113) was bought out and put on a backorder list that was over 900 in length. This occurred around the time that our team completed testing the two TG.45.8113s we had already purchased. Thankfully, our team had developed a design matrix for our 5G antenna and were able to purchase the next-best part for the job (TG.35.8113). Along with re-testing of our antennas, we adjusted our carrier frequency from 600 MHz to 750 MHz based on the S11 parameters of the new antenna.

Project Requirements

The following requirements are mandatory and must be met in order to satisfy the needs of our customer base.

Marketing Requirements (MR)

MR-1. The system must streamline the process of the base station determining the direction-of-arrival of the incoming signal (less time and less energy than triangulation).
MR-2. The system must be able to determine direction-of-arrival within an acceptable range.
MR-3. The system can be modified for other environments through training of the neural network.
MR-4. The system must be able to handle noise up to a certain threshold.
MR-5. The system must be able to understand and work with low-band 5G signals.
MR-6. The project should have an interface where the user can see data clearly.
MR-7. The system must be in-expensive enough for mass production.

Engineering Requirements (ER)

ER-1. The system must be able to estimate the direction-of-arrival of the transmitted signal with an accuracy level of 90% or greater.
ER-2. The system must work for 5G signals transmitted within a radial distance of at least 6 meters.
ER-3. The system must be able to come up with a valid model for any environment it is trained in.
ER-4. The system must be accurate in the presence of <= -40 dBW of noise.
ER-5. The system must be able to work with frequencies in the 600-850 MHz band.
ER-6. There must exist a GUI that displays real and accurate data within 60 seconds.
ER-7. Prototype must cost less than $600.

Implementation

System Architecture

The Seer is a system of three main stages. The first stage is data collection, where our Rx composed of our antenna array is used along with a Tx sending a low-band 5G signal in the form of a constant sinusoid. Using Latin Hypercube Sampling (LHS) the testing environment is unbiasedly sampled, and a database of measurements is collected. Once a signal is sent and received by our antenna array it enters stage two. Stage two is the data processing stage, where all the signal processing is accomplished. Using GNU Radio, we run FFT signal processing on the received signals from each antenna, extracting the target magnitude and phase data. The data is saved as matrices of the extracted data in binary 32 format before entering stage three. Stage three is the machine learning stage, the binary data files are preprocessed using Python and the last value of each matrix is loaded into tensors along with the (r,theta) coordinate corresponding to the distance between the Rx and Tx for each measurement. These tensors are loaded into the neural network for training, after which the network is ready to make predictions on validation data. The validation data is collected in the same way, and after the predictions are made the results are displayed on the team's GUI.

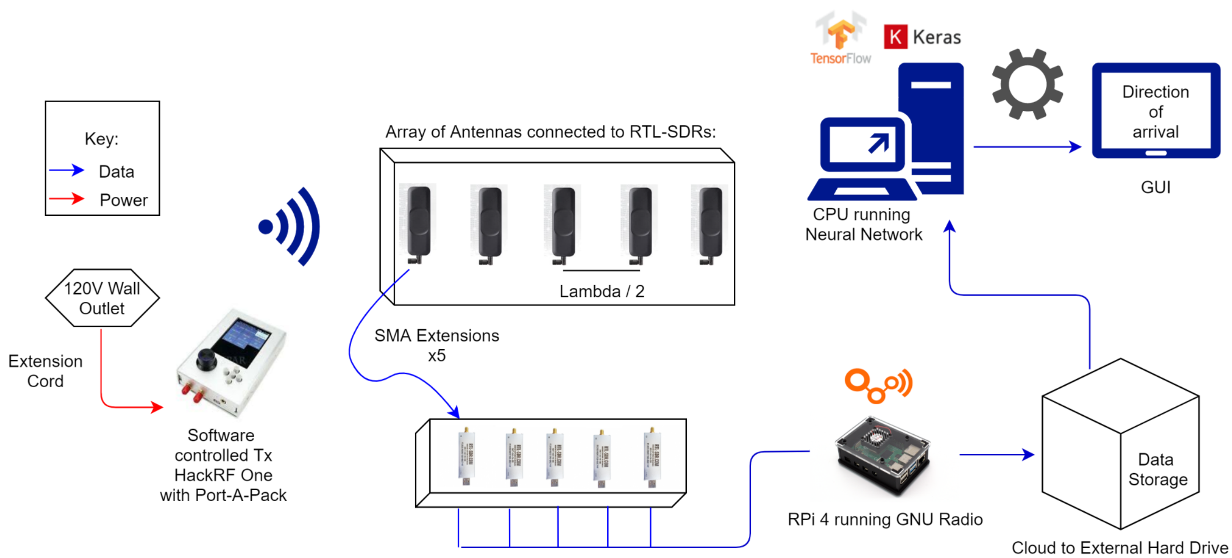

Fig. 2 - Original System Diagram

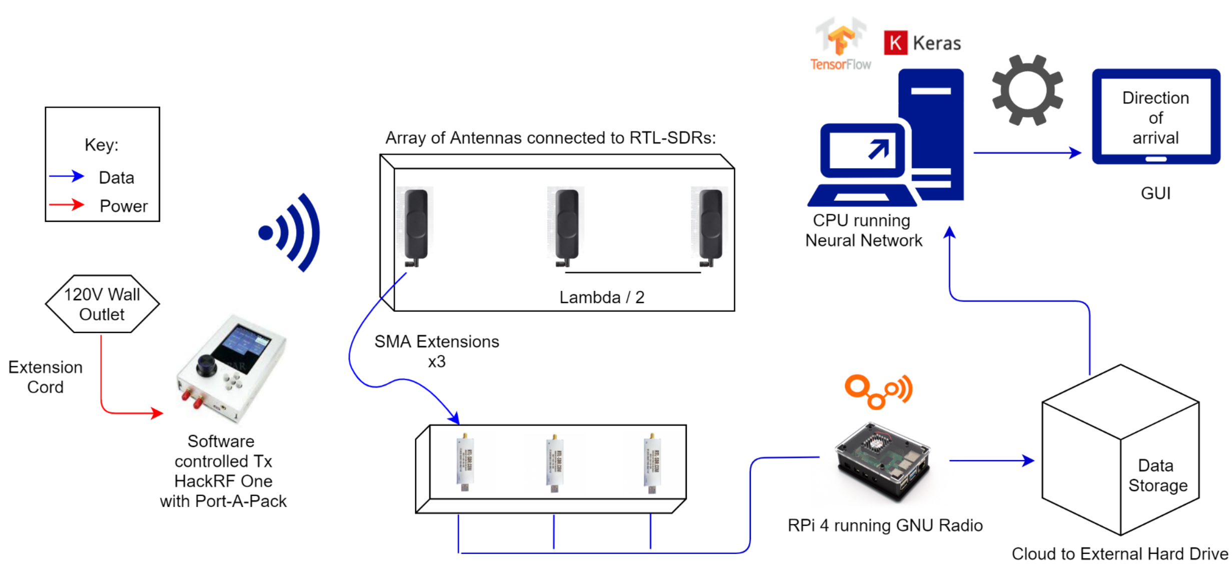

Fig. 3 - Final System Diagram

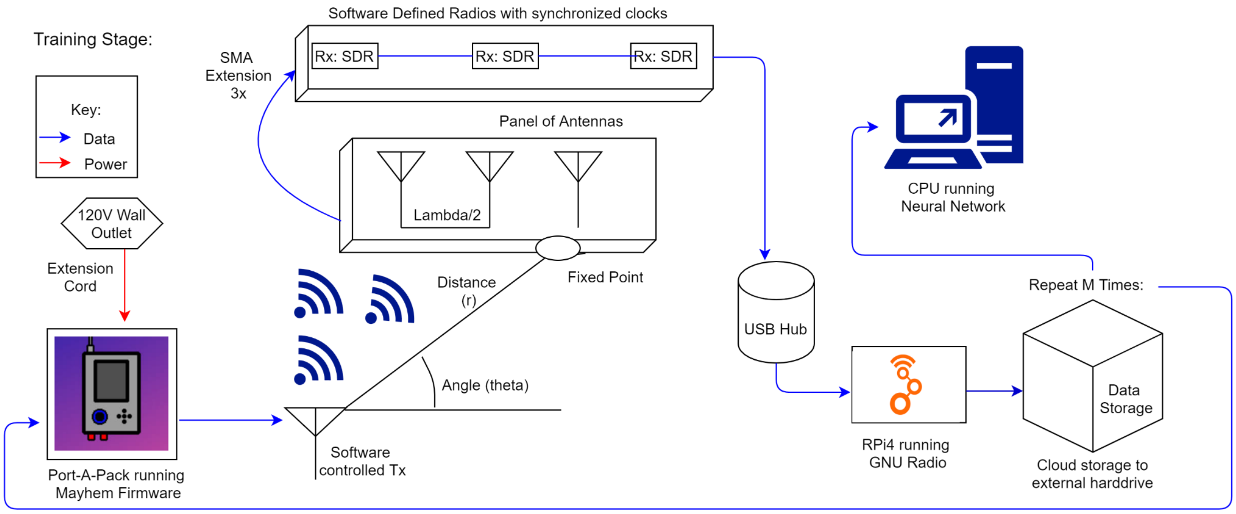

Fig. 4 - Hardware Block Diagram - Training Stage

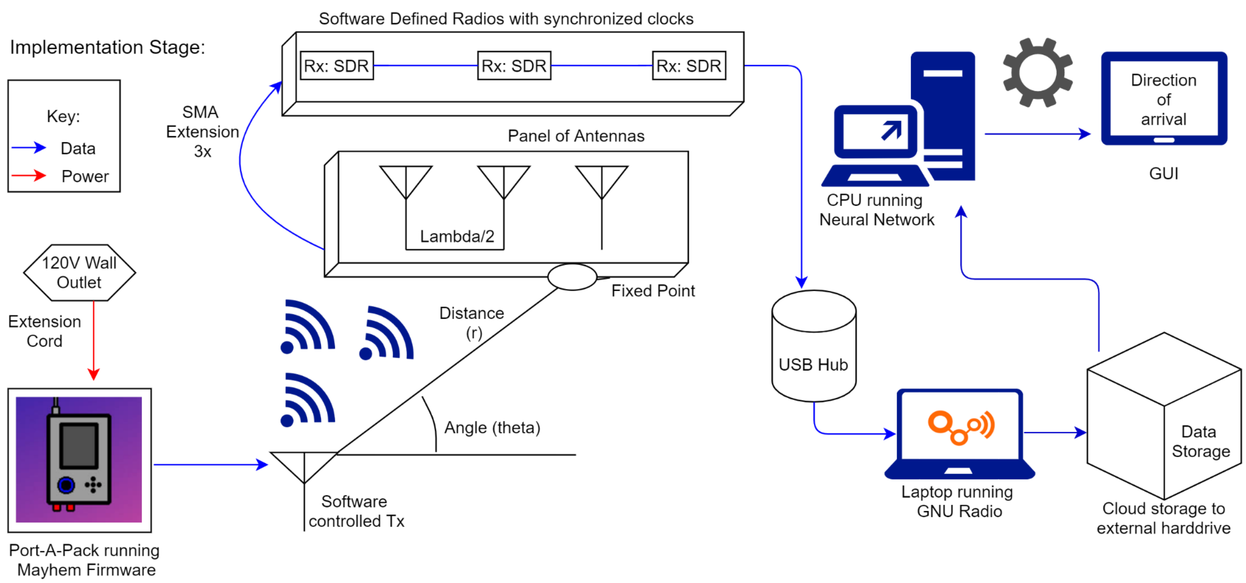

Fig. 5 - Hardware Block Diagram - Implementation Stage

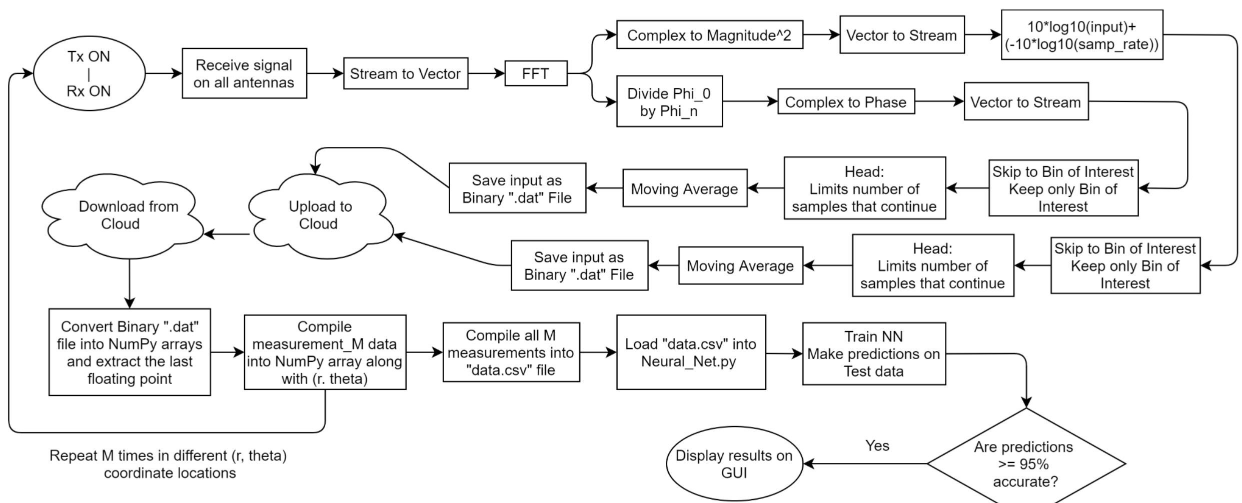

Fig. 6 - Software Flowchart Training Stage

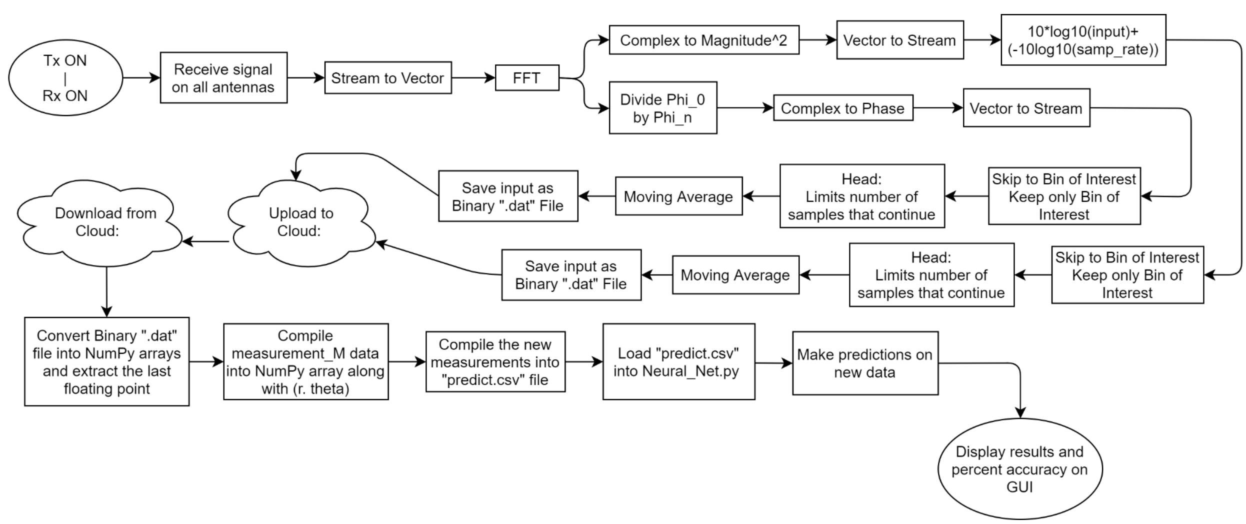

Fig. 7 - Software Flowchart Implementation Stage

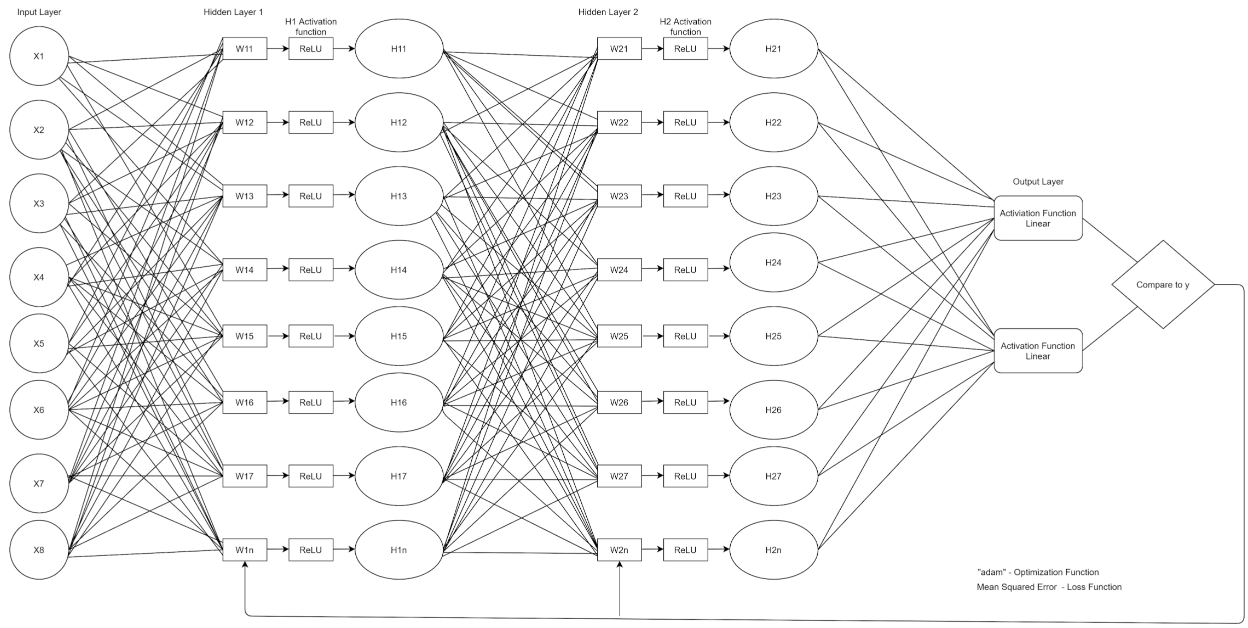

Fig. 8 - Software Flowchart - Neural Network

Components

Using the supplies tabulated below, we built our prototype, The Seer. The justification for a few of the major system components can also be seen in the "Design Matrices" below. The HackRF One was used to transmit low-band 5G signals, while the RTL-SDRs were used to receive said signals. The antennas our team used to both transmit and receive the low-band 5G signals would be the TG.35.8113 Apex III Wideband 5G/4G Dipole Terminal Antenna. We originally tested the TG.45.8113 but they were put on backorder, so we defaulted to the next best antenna from our design matrix. To control the Rx, we used GNU Radio running on a Raspberry Pi 4. Using multiple USB driven SDRs, our team used a powered USB-Hub to connect all the peripheral SDRs, utilizing low-loss SMA extension cables to connect them to their respective antennas. With multiple SDRs, each with their own clock, the team synchronized the SDR clocks to form a coherent Rx, using solder and low-loss wire. Lastly, to secure our several receiving antennas in an array, the team used adhesive to attach the antennas to our PVC pipe structure, ensuring stability of the antennas during the testing and implementation of our system. With all the components and supplies accounted for, the total cost of our prototype was $405.23.

Note: Duplicate and spare components were purchased as backup supplies, in lieu of potential hardware devices failing or being damaged in the construction stage of our system.

Design Matrices:

Software

	Cost	Usability	Compatibility	Familiarity	Total Score

PyTorch	10	5	7	4	26
Tensor Flow (Keras)	10	10	7	4	31
Matlab	1	5	4	10	20

Key: Scoring is based from 1 (unideal) - 10 (ideal), with the highest total score being desired

Table 1 - Design Matrix - Software

Transmitter

	Cost	Usability	Frequency Selectability	Output Power	Total Score
TRC101 Murata Electronics	8	2	3	7 (can add amplifier)	20
HackRF One	5	8	10	5 (can only amplify using antenna)	28
HackRF One w/Porta Pack	1	10	10	5 (can only amplify using antenna)	26

Key: Scoring is based from 1 (unideal) - 10 (ideal), with the highest total score being desired

Table 2 - Design Matrix - Transmitte**r**

Antenna

	Cost	Directionality (we want omnidirectional)	Gain at 600MHz	Gain at 850MHz	Bandwidth	Total Score
TG.35.8113 - Taoglas Apex II Wideband 5G/4G Dipole Terminal Antenna	\$17.31 3	10	(-3.5,-2) 5	(-1.2,+1.6) 7	10	35
TG.45.8113- Taoglas Apex III Ultra-Wideband 5G/4G	\$16.46 4	10	(-4.2,0) 6	(-2,+2.7) 8	10	38

Dipole Terminal Antenna						
TG.46.8113 - Taoglas Apex IV Wideband 5G/4G Dipole Terminal Antenna	NA	10	(-3,-0.7) 6	(-3.5,+0.5) 6	10	32
TG.55.8113 W - Taoglas 5G/4G Terminal Mount Monopole Antenna	$5.77 8	3 (Need ideal ground plane)	(-4,+1) 6	(-4.5,+1) 6	10	31

Key: Scoring is based from 1 (unideal) - 10 (ideal), with the highest total score being desired

Table 3 - Design Matrix - Antenna

Prototype Cost

Supplies	Description	Price (U.S. Dollars)	Quantity	Total Cost
RPi 4 (8GB RAM)	System on Chip (SoC) used for processing the received signals through GNU Radio	$104.99	1	$104.99
32GB MicroSD Card	32GB MicroSD Card	$8.25	1	$8.25
RTL-SDRs	Software defined radio receiver capable of tuning frequencies of 500 kHz to 1.7 GHz	$26.59	5	$132.95
Antennas	TG.35.8113 Apex III Wideband 5G/4G Dipole Terminal Antenna with added 450 MHz band	$16.46	5	$82.3

Coax Cables	1M low loss SMA extension RF transmission lines	$9.99	5	$49.95
Software: Keras, OsmoSDR	Deep Learning Python library that uses TensorFlow2 as a backend Library of GNU Radio blocks used to run SDR Rx and Tx	FREE	NA	(open source)
PVC Pipe & Adhesive	Used for the antenna array structure	$9.99	1	$9.99
USB Hub	USB Hub with 1.2A output power from wall outlet	$16.80	1	$16.80
Total	Prototype cost			**$405.23**

Table 4 - Prototype Cost - The Seer

Project Schedule

This past year was challenging in many regards, one of those being the division of labor for our project. Since some of the team members were in LA county while others were in Sonoma county, the work had to be delegated accordingly. The software tasks could be accomplished from anywhere, so these were mainly handled by team member Tate, in LA county, while Evan handled much of the hardware testing and system construction. Victor was in charge of the system implementation and testing, measurement collection, as well as the verification for our system. Tate and Evan handled the signal processing and the LHS, while Victor and Evan handled the GUI. The entire team was involved in many of the design decisions and major tests, working together virtually to solve whatever problems arose. All team members worked hard to design, build, and test our system, The Seer, overcoming many hurdles due to the complexity of the project, and the added hardships of COVID-19.

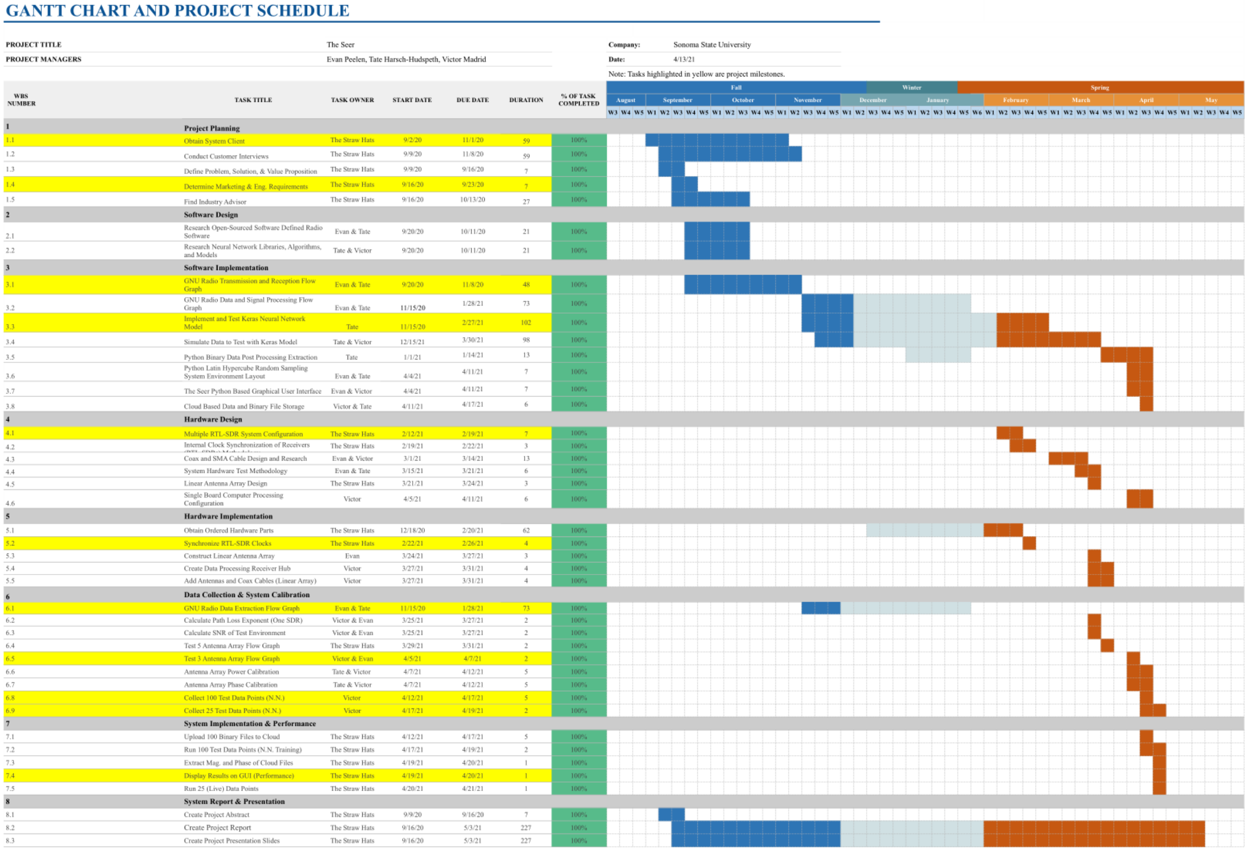

Figure 9 - Project Schedule

Tests

Test Overview:

Functional Tests:

Test 1A - Hardware limitations using an AM signal, ER - 1, 2, 3, 4

Test 1B - Hardware limitations using an FM signal, ER - 1, 2, 3, 4

Test 1C - Finding the environmental path loss exponent, ER - 1, 2, 3, 4

Test 3 - Test coax cables and find S11 parameters of antennas, ER - 1, 2, 4, 3, 5

System Verification Tests:

Test 2A - Neural Network assessment using Simulation 01 (Friis), ER - 1, 3

Test 2B - Neural Network assessment using Simulation 02 (JTC w/ Noise), ER - 1, 3, 4

Test 4 - Antenna Array clock sync. time delay analysis, ER - 1, 3, 5

Test 5 - Flowgraph test with 5 RTL-SDRs, ER - 1, 2, 3, 4, 5

Test 6A - Antenna Array test environment power calibration, ER - 1, 2, 3, 4, 5

Test 6B - Antenna Array test environment phase calibration, ER - 1, 2, 3, 4, 5

Test 7 - GUI display test and result display-time analysis, ER - 1, 6

Description of Tests:

Test 1:

Objective: To identify a parameter that is scalable by the distance of transmission and reception of a low-band 5G signal, using amplitude and frequency modulation. The team would evaluate both the noise behavior of the received AM and FM signals, as well as the testing environment, viewing and coming to conclusions on the transmission and reception performance of both amplitude and frequency modulated signals within our system's test environment. The performance of both modulation schemes would be compared graphically with one another, as the team would decide based off of performance and familiarity, what modulation scheme to use within The Seer system. This AM and FM comparison would be explored in the overall conclusion for Test 1. Additionally, the team would find the path loss exponent for each modulation scheme and once the optimal transmission scheme is determined (FM), the team would then calculate the corresponding variance and standard deviation between the distance versus received signal power trials (3), implementing one RTL-SDR and The Seer data extraction flow graph.

Setup: To complete this test, the team would implement the HackRF One and Porta Pack, along with both the TG.45.8113 and TG.35.8113 wideband 5G/4G dipole terminal antennas, driven by GNU Radio software transmitting a 1 kHz cosine wave at a carrier frequency of 600 MHz, and then in Test 1C a 1 kHz constant wave at a carrier frequency of 750 MHz. On the receiving end, the team would use a RTL-SDR paired with the TG.45.8113 wideband 5G/4G dipole terminal antenna, driven by GNU Radio software to receive and display a FFT plot containing the transmitted co-sinusoidal wave for performance and transmission distance analysis. Both the transmitter (HackRF One) and the receiver (RTL-SDR) would be spaced 2-12 meters apart, in increments of 2 meters. At each gradual 2 meter increment in distance, the team would record the received signal strength and noise floor of the surrounding test environment. In Test 1C, this test would be conducted in an outdoor-urban environment, taken at distances of 1-10 meters, in steps of 1 meter increments. Both the transmitter (HackRF One and Porta Pack) and receiver (RTL-SDR) would be on equally raised surfaces as this test was conducted. At each measured distance, the test would be run three times in order for the standard deviation and variance to be calculated between each test trial. Due to FCC regulations pertaining to transmission power, all tests would be conducted at a power of less than 1 W.

Test 1A: Hardware limitations using an AM signal, ER - 1, 2, 3, 4

Objective: To complete the Test 1 objective above using an amplitude modulated signal.

Setup: To complete this test, the team would implement the Test 1 setup. The AM signal that would be used within this setup would consist of a 1 kHz cosine wave transmitted at a carrier frequency of 600 MHz.

Results: The team would plot the data points for both the signal strength and SNR of the received AM signals at distances from 2-12 meters, in increments of 2 meters, as these graphs can be seen below.

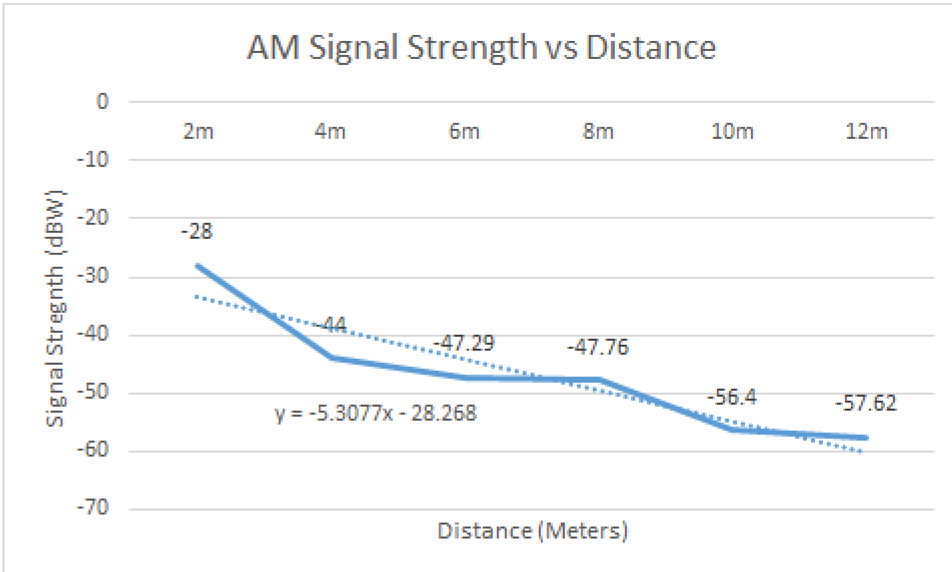

Figure 10 - Recorded AM Signal Strengths (2-12 Meters)

Figure 10 shows that as the transmitter and receiver distance grew apart, the received signal strength decreased with increased distance, proving that signal strength (dBW) is scalable by distance. The team would run a linear regression on the plotted data above, in order to view the path loss exponent of the transmitted signal. The path loss exponent is representative of the rate at which an electromagnetic wave's power decreases with distance. From the linear regression of the plotted data, the team would observe a path loss exponent of 5.3077, which is representative of the suburban test environment in which this test was conducted.

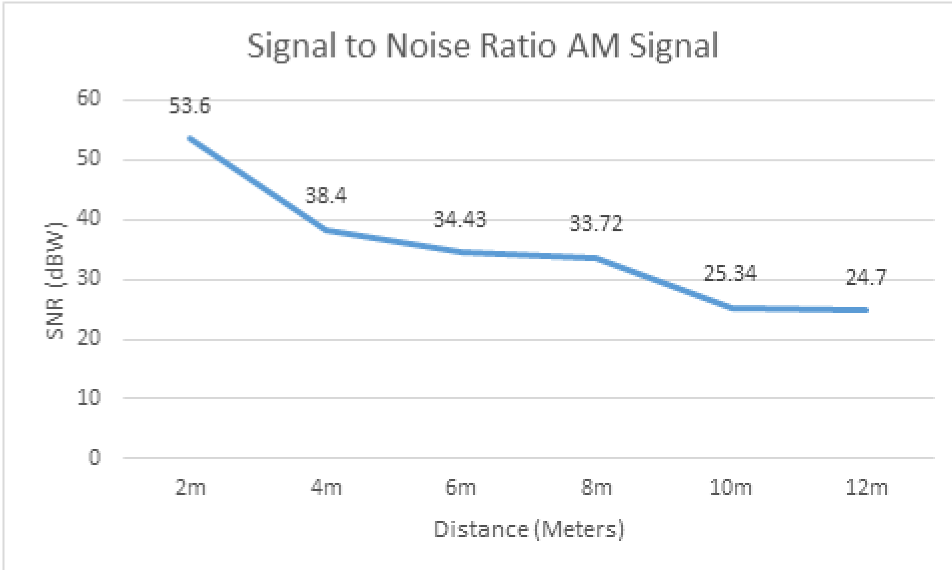

Figure 11- Signal to Noise Ratio of Received AM Signals (2-12 Meters)

Figure 11 displays the corresponding signal to noise ratio of the received low-band 5G signals, taking into account the noise floor of the testing environment and the strength of the received signals. The signal to noise ratio describes the power level of a signal compared to the power level of the noise floor within the testing environment. The highest SNR the team would record for the AM signal would be 53.6 dBW and the lowest recorded SNR would be 24.7 dBW. Through this test, the team would observe that the SNR of the test data points decrease with increased separation between both the transmitter and receiver. Having a sufficiently high level of signal power compared to noise power at each tested distance, this would confirm that the testing environment in which the team would transmit in would be viable for future AM signal tests.

Conclusion: Through this test, the team identified that AM signal strength (dBW) is scalable by the distance of transmission and reception of a low-band 5G signal. While evaluating both the noise behavior of the received AM signals as well as the testing environment, the team observed an overall sufficient signal to noise ratio at each tested data point, coming to the conclusion that the testing environment was suitable for future AM signal tests.

Test 1B: Hardware limitations using an FM signal, ER - 1, 2, 3, 4

Objective: To complete the Test 1 objective above using a frequency modulated signal.

Setup: To complete this test, the team would implement the Test 1 setup. The FM signal that would be used within this setup would consist of a 1 kHz cosine wave transmitted at a carrier frequency of 600 MHz.

Results: The team would plot the data points for both the signal strength and SNR of the received FM signal at distances from 2-12 meters, in increments of 2 meters, as these graphs can be seen below.

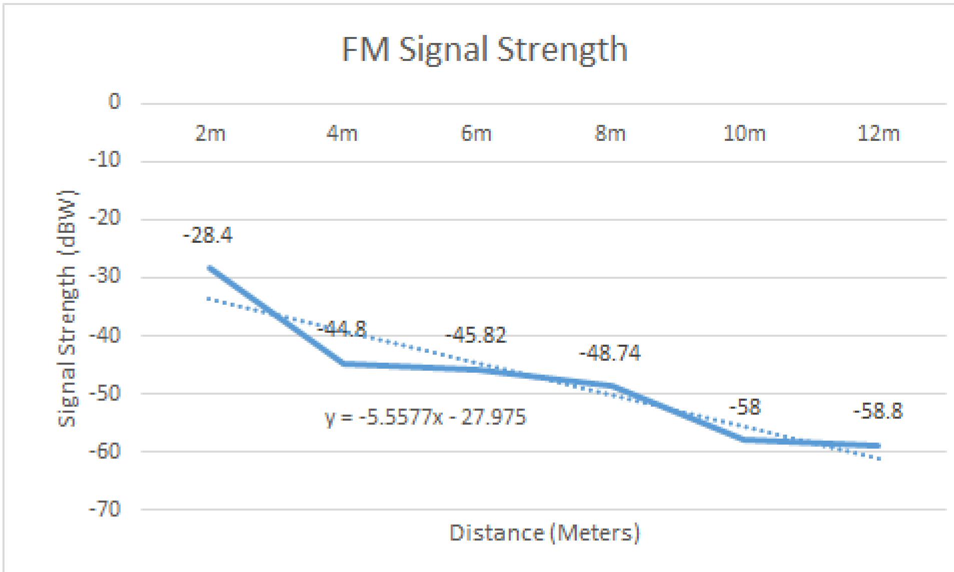

Figure 12 - Recorded FM Signal Strengths (2-12 Meters)

Figure 12 shows that as the transmitter and receiver distance grew apart, the received FM signal strength decreased with increased distance, proving that signal strength (dBW) is scalable by distance for the FM signal. The team would run a linear regression on the plotted data above, in order to view the path loss exponent of the transmitted signal. From the linear regression of the plotted data, the team would observe a path loss exponent of 5.5577, which is representative of the suburban test environment in which this test was conducted.

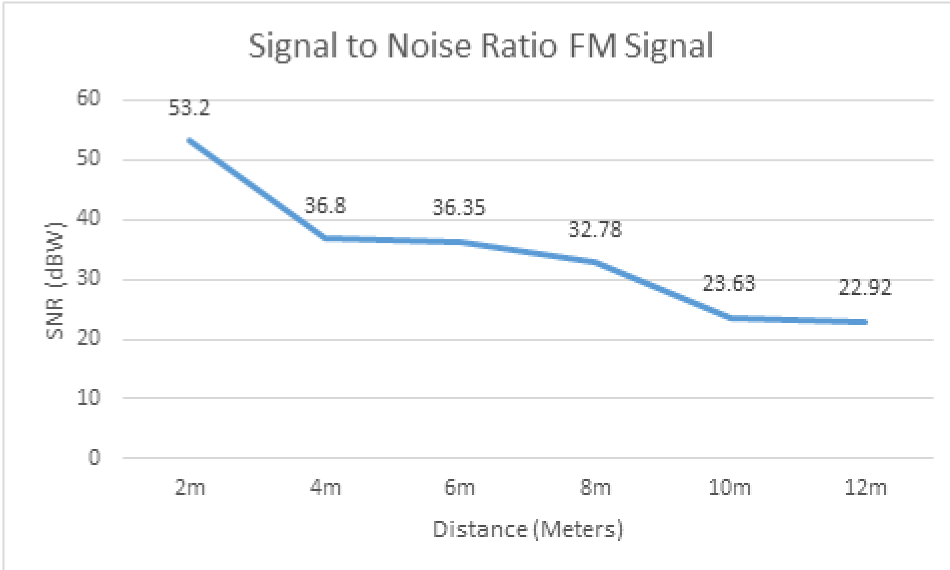

Figure 13 - Signal to Noise Ratio of Received FM Signals (2-12 Meters)

Figure 13 displays the corresponding signal to noise ratio of the received low-band 5G signals, taking into account the noise floor of the testing environment and the strength of the received FM signals. The highest SNR the team recorded for the FM signals was 53.2 dBW and the lowest recorded SNR was 22.92 dBW. Through this test, the team observed that the SNR of the test data points decreased with increased separation between both the transmitter and receiver. Having sufficiently high levels of signal power compared to noise power at each tested distance, confirmed that the testing environment in which the team transmitted in was viable for future FM signal tests.

Conclusion: Through this test, the team identified that FM signal strength (dBW) is scalable by the distance of transmission and reception of a low-band 5G signal. While evaluating both the noise behavior of the received FM signals as well as the testing environment, the team observed an overall sufficient signal to noise ratio at each tested data point, coming to the conclusion that the testing environment was suitable for future FM tests.

Test 1C: Finding the environmental path loss exponent, ER - 1, 2, 3, 4

Objective: The objective of this test was to find the path loss exponent and corresponding variance and standard deviation between the distance versus received signal power trials (3), implementing one RTL-SDR and The Seer data extraction flow graph, using frequency modulation.

Setup: Conducted in an outdoor-urban environment this test would be taken at distances of 1-10 meters, in steps of 1 meter increments. Both the transmitter (HackRF One and Porta Pack) and receiver (RTL-SDR) were raised on surfaces of the same height, as this test was conducted. At each measured distance, the test was run three times in order for the standard deviation and variance to be calculated between each test trial.

Porta Pack Settings

Gain: 15 dB

Signal Source: Constant Source

Signal Source Frequency: 1 kHz

Carrier Frequency: 750 MHz

Results

Trial 1

Pt_0:

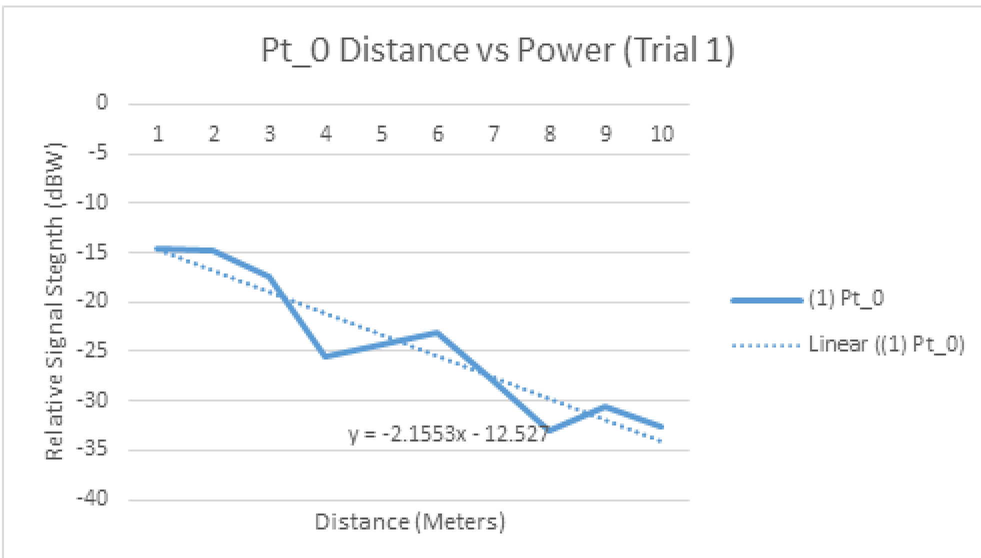

Figure 14 - Pt_0 Distance vs Power Data Trial 1

Path Loss Exponent: -2.1553

Standard Deviation: 6.906352 dBW

Variance: 47.6977 dBW

Pt_1:

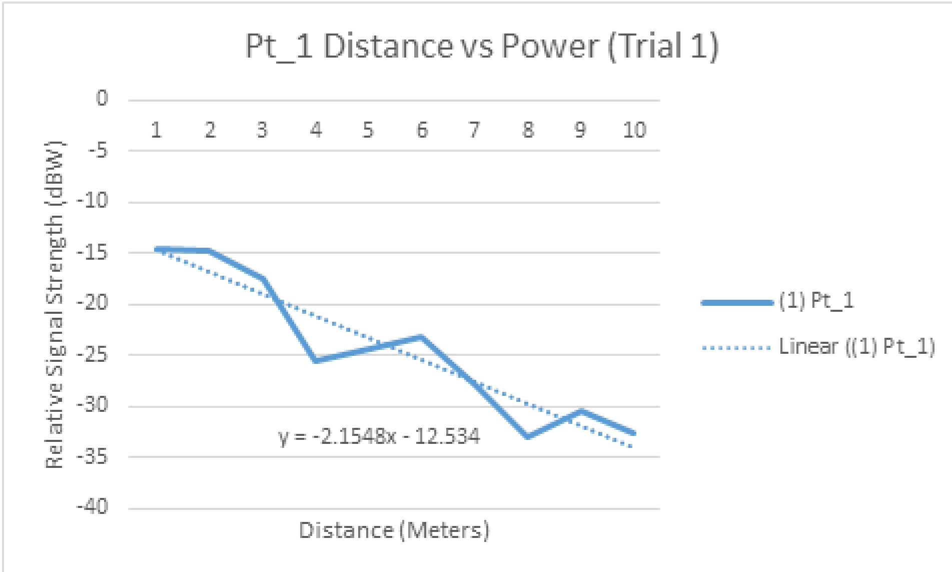

Figure 15 - Pt_1 Distance vs Power Data Trial 1

Path Loss Exponent: -2.1548

Standard Deviation: 6.905393 dBW

Variance: 47.68445 dBW

Trial 2

Pt_0:

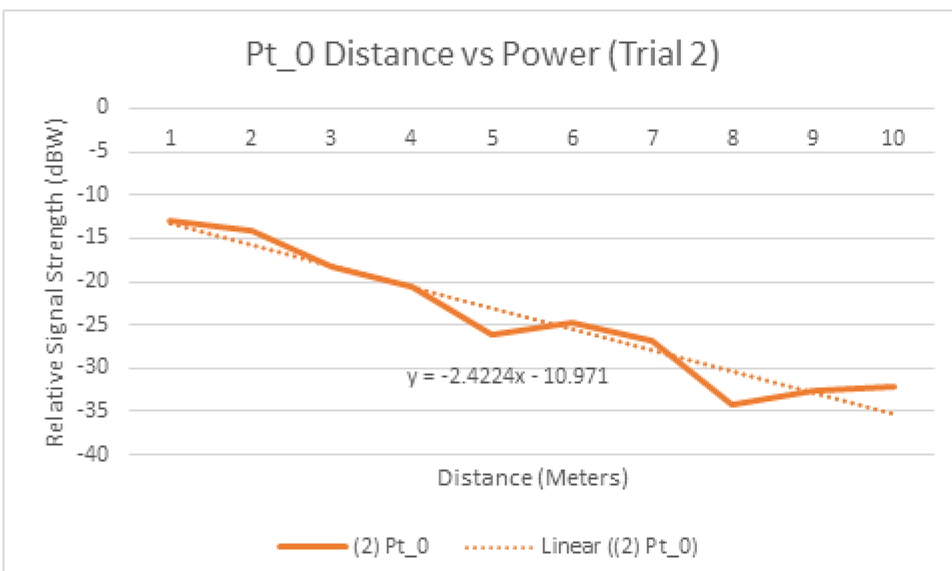

Figure 16 - Pt_0 Distance vs Power Data Trial 2

Path Loss Exponent: -2.4224

Standard Deviation: 7.619225 dBW

Variance: 58.05259 dBW

Pt_1:

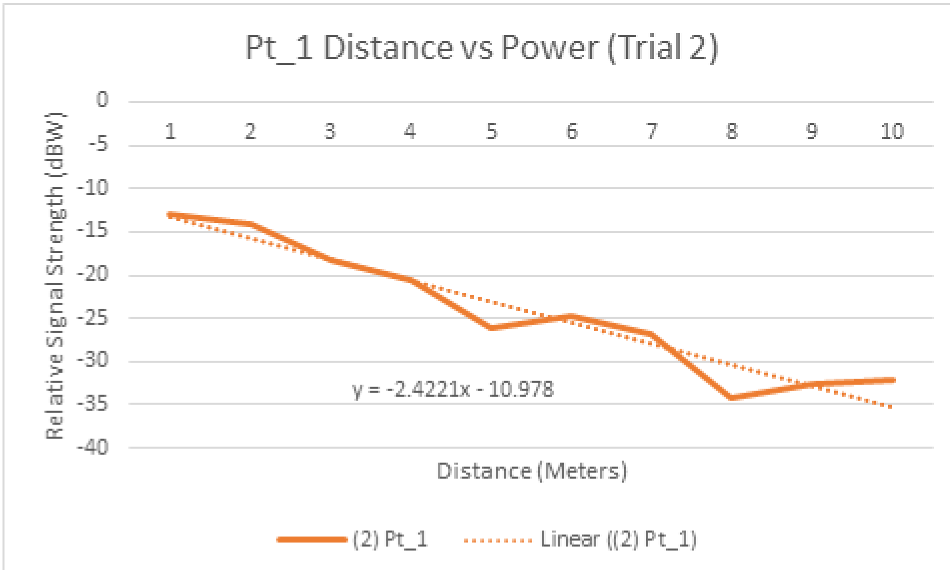

Figure 17 - Pt_1 Distance vs Power Data Trial 2

Path Loss Exponent: -2.4221

Standard Deviation: 7.617996 dBW

Variance: 58.03386 dBW

Trial 3

Pt_0:

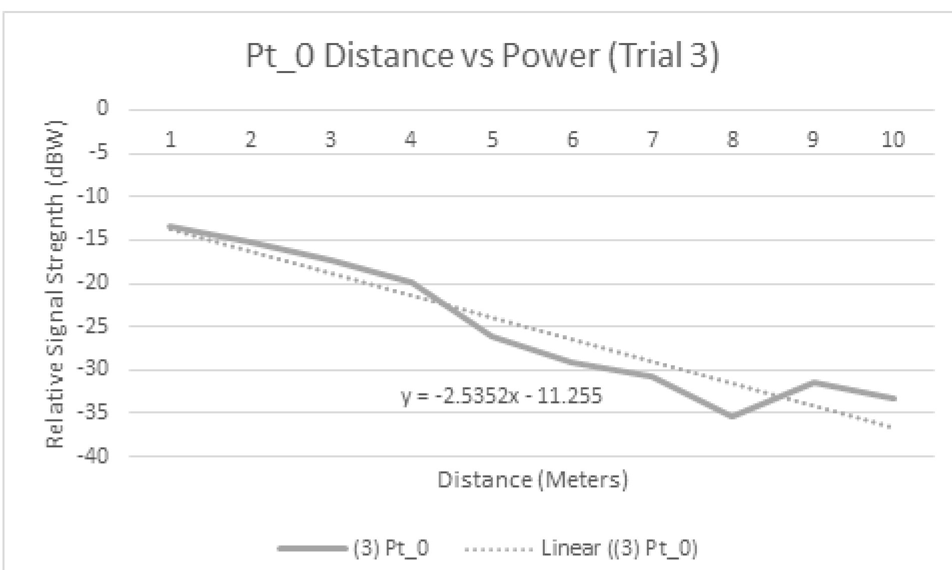

Figure 18 - Pt_0 Distance vs Power Data Trial 3

Path Loss Exponent: -2.5352

Standard Deviation: 8.065089 dBW

Variance: 65.04567 dBW

Pt_1:

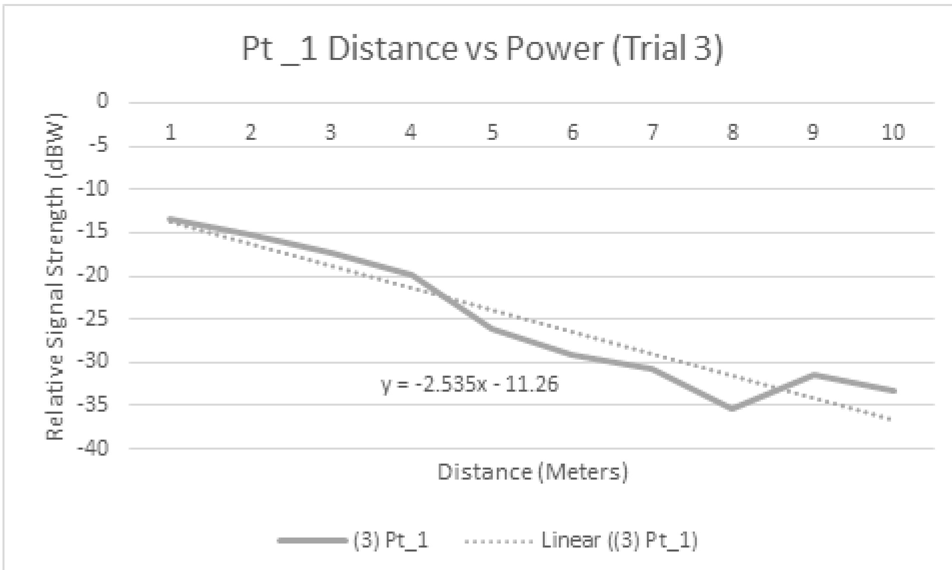

Figure 19 - Pt_1 Distance vs Power Data Trial 3

Path Loss Exponent: -2.535

Standard Deviation: 8.064476 dBW

Variance: 65.03577 dBW

Observer Notes: While taking the above measurements, the team noted that the noise floor was centered about -40.29 dBW, which was higher than previous test environments. Originally, the team had planned on taking measurements from 1-12 meters, however, at a distance past 10 meters the received signal's strength was no longer significantly larger than the noise floor and was being "overshadowed" by the test environment noise floor. The "overshadowing" effect of the large noise floor and relatively small received signal strength are demonstrated in the FFT plots as seen in Figures 20 and 21 below. The relatively low received signal strength is due to the transmitter gain being set to 15. Any larger gain over 20 may prove to be fatal to the team's SDR (RTL-SDR), as this proved to be a hardware limitation of the RTL-SDR, causing the team to decrease their maximum distance from 12 to 6 meters.

Sample FFT Plots Taken During Measurement:

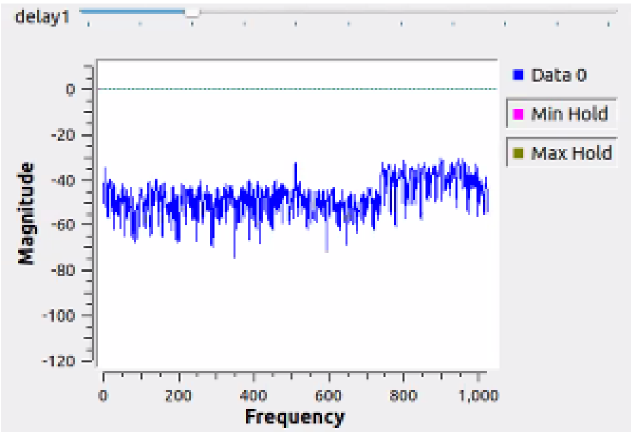

Figure 20 - FFT Plot 7 Meters

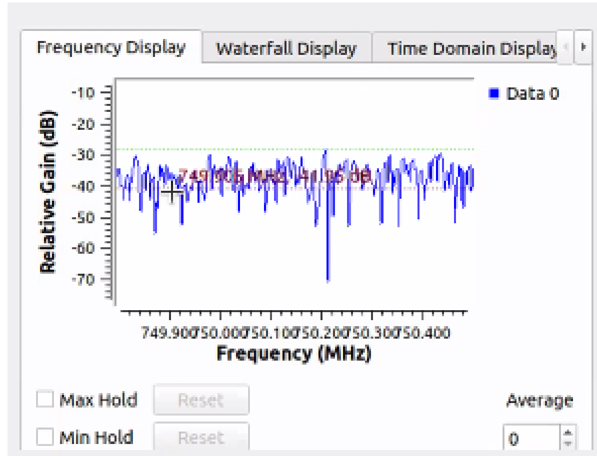

Figure 21 - FFT Plot 10 Meters

Conclusion: Through conducting this test, the team was able to successfully find the path loss exponent, corresponding variance, and standard deviation between the distance versus received signal power trials. The team calculated an average path loss exponent of -2.37 for Pt_0, that being the physical receiving antenna. An average path loss exponent of -2.371 would represent the team's outdoor-urban testing environment well, as a typical path loss exponent for such an environment lies between -2 to -3. By conducting this test, the team also observed a relatively high noise floor compared to the received signal strength. To fix this issue, the gain setting of the transmitter (HackRF One) was raised in order to put a minimum of 20 dBW between the peak received signal power and the test environment noise floor. Below, each trials' data for both Pt_0 and Pt_1 can be viewed, as each trial resulted in similar results with an acceptable standard deviation and variance.

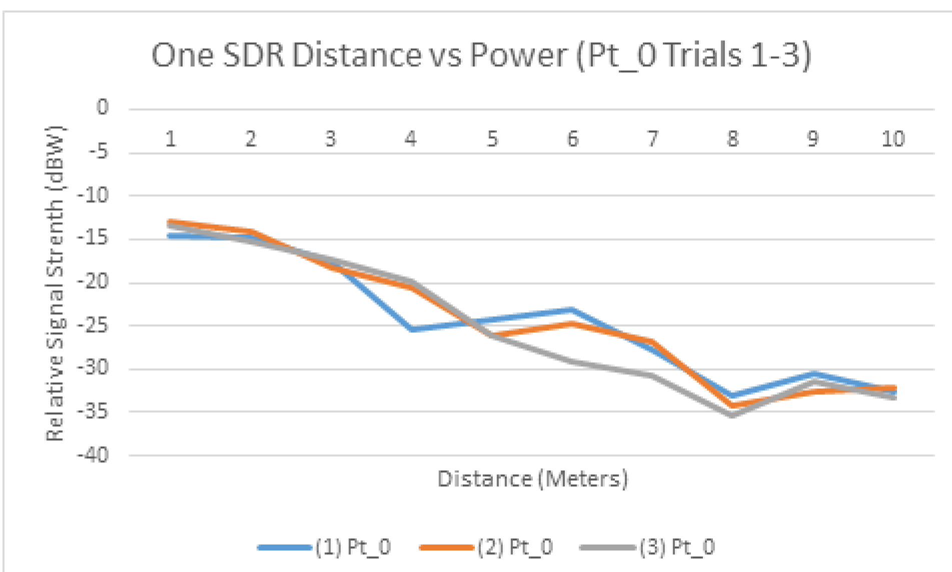

Figure 22 - One SDR Distance vs Power Plots for Each Trial (Pt_0)

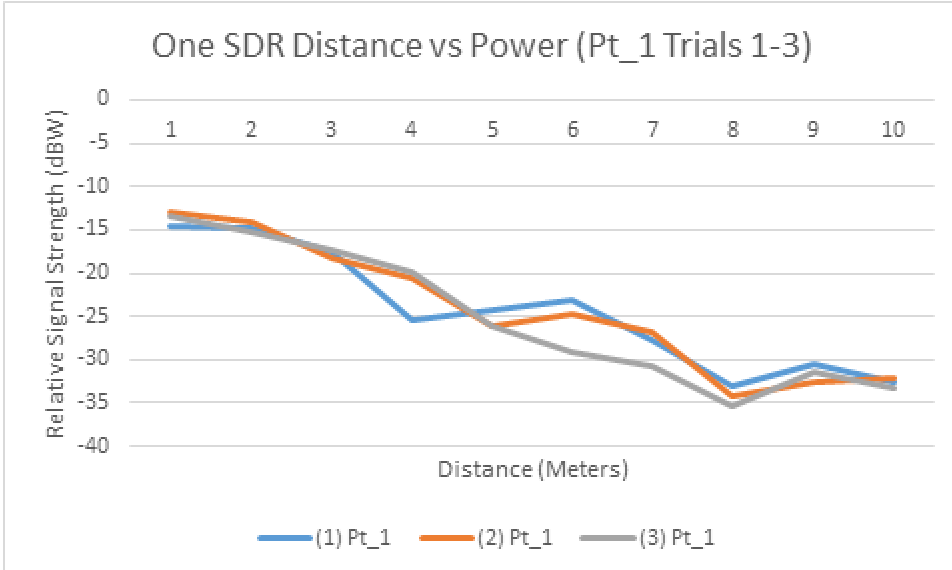

Figure 23 - One SDR Distance vs Power Plots for Each Trial (Pt_1)

Test 1 Conclusion: This test identified that the received signal power of low-band 5G signals using both AM and FM are scalable by distance. Through evaluating both the noise behavior of the received AM and FM signals, as well as the relationship between distance and power, the team determined that AM and FM would both work for our system's method of signal transmission. Because the team acquired the Port-A-Pack to use with the HackRF One, which is able to produce a constant sine wave, we chose FM as our final transmission method. We also confirmed that our path loss exponent for our testing environment aligned with the team's expectations. This shows that our hardware was functioning correctly and was not going to cause problems later down the line. Lastly, the team reduced our maximum distance from 12 meters to 6 meters due to the limitations of the power level accepted by the SDRs.

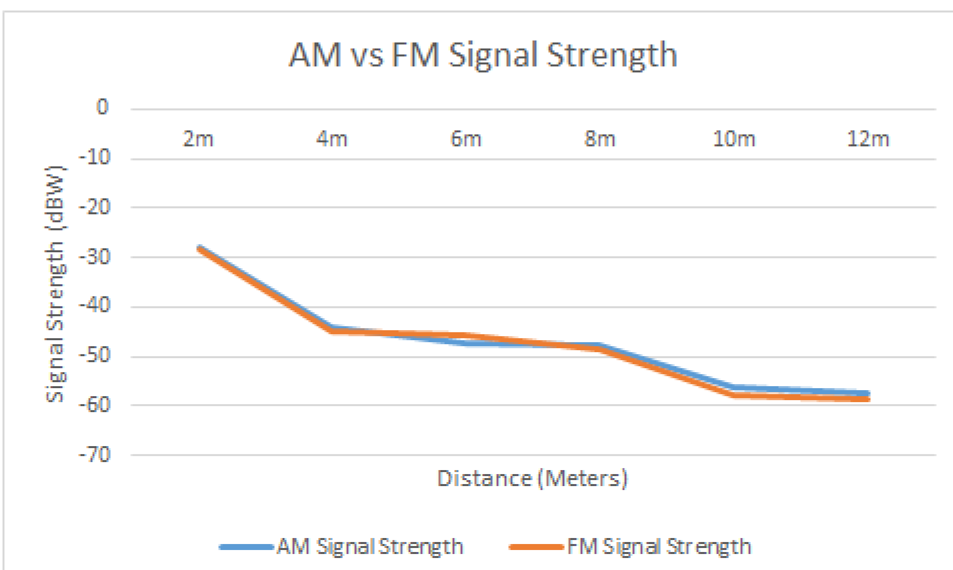

Figure 24 - AM vs FM Signal Strength Comparisons

Figure 24 above shows both the plotted AM and FM signal strength data, taken from 2-12 meters in increments of 2 meters for each tested data point. Evaluating the plotted data, the team would notice that the AM and FM signal strengths for each tested distance (2-12 meters) was nearly identical within our suburban testing environment. Both AM and FM received signal strengths would be scalable by distance and displayed nearly identical path loss exponents of -5.31 for the AM received signals and -5.56 for the FM received signals.

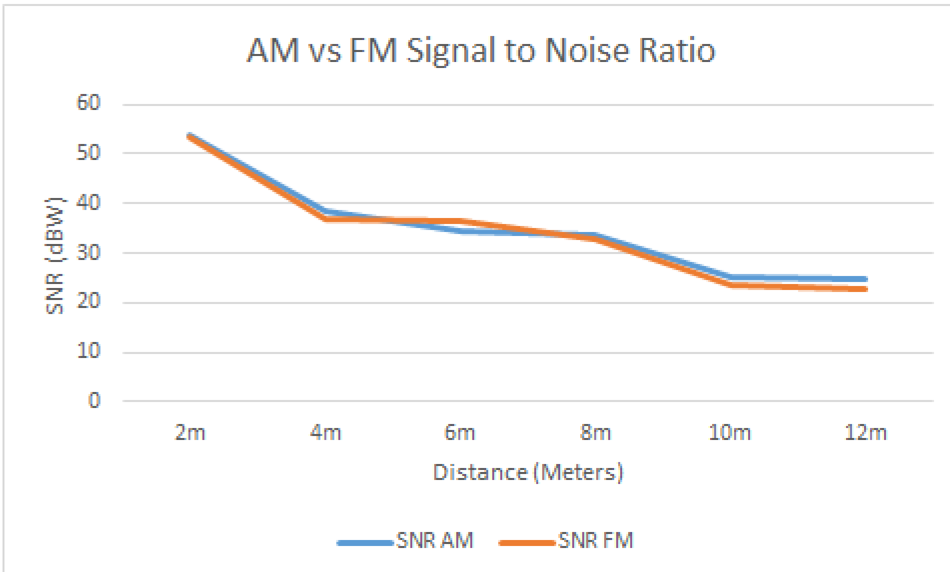

Figure 25- AM vs FM Signal to Noise Ratio Comparisons

With the data sets for each signal modulation scheme superimposed on one another, displaying nearly identical performance characteristics, the team would decide to use frequency modulation as the signal modulation scheme within the Seer System, due to the addition of the Port-A-Pack paired with the HackRF One.

Test 2:

Objective: We are analyzing our first Keras model. Our model attempts to solve our multi-output regression problem. The goal of this test is to measure our neural network's ability to handle relevant data while gathering information on the complexity of our model and determining any differences between the ideal free-space propagation model (Friis) and a more complex model (JTC with added noise). After initial testing, the team worked to update the model in order to find the best arrangement of layers and weights for each propagation model in preparation for measured data.

Setup:

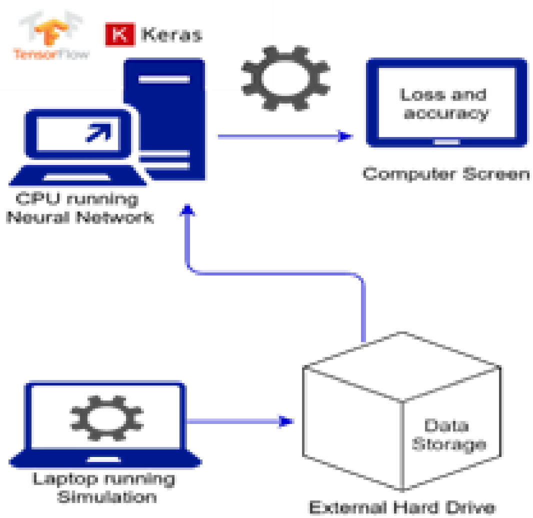

Figure 26 - Test 2 Setup

Expectations & Results:

Test 2A - Neural Network assessment using Simulation 01 (Friis), ER – 1, 3

Objective: Use the Friis transmission equation free space path loss RF propagation model to assess our Keras model's ability to handle relevant data.

Expectation: We expect our model to handle this RF propagation model well, producing a high loss that falls relative to the number of epochs run.

Results: The learning rate of the NN, utilizing Test 2A data:

Early epochs yield high loss but also high learning rate. Loss starts off at 5,180, then drops under 1,000 within the first 35 epochs. The NN reaches a loss of 1.51 by the 1,150th epoch. Using 25 predictions, our model produced an average difference:

$$\frac{\sum(\mathrm{ABS}(y\_predicted - y\_collected))}{25}$$

of 0.38 meters for the r value, and 4.42 degrees for theta with a standard deviation of 0.42 meters for r and 4.43 degrees for theta.

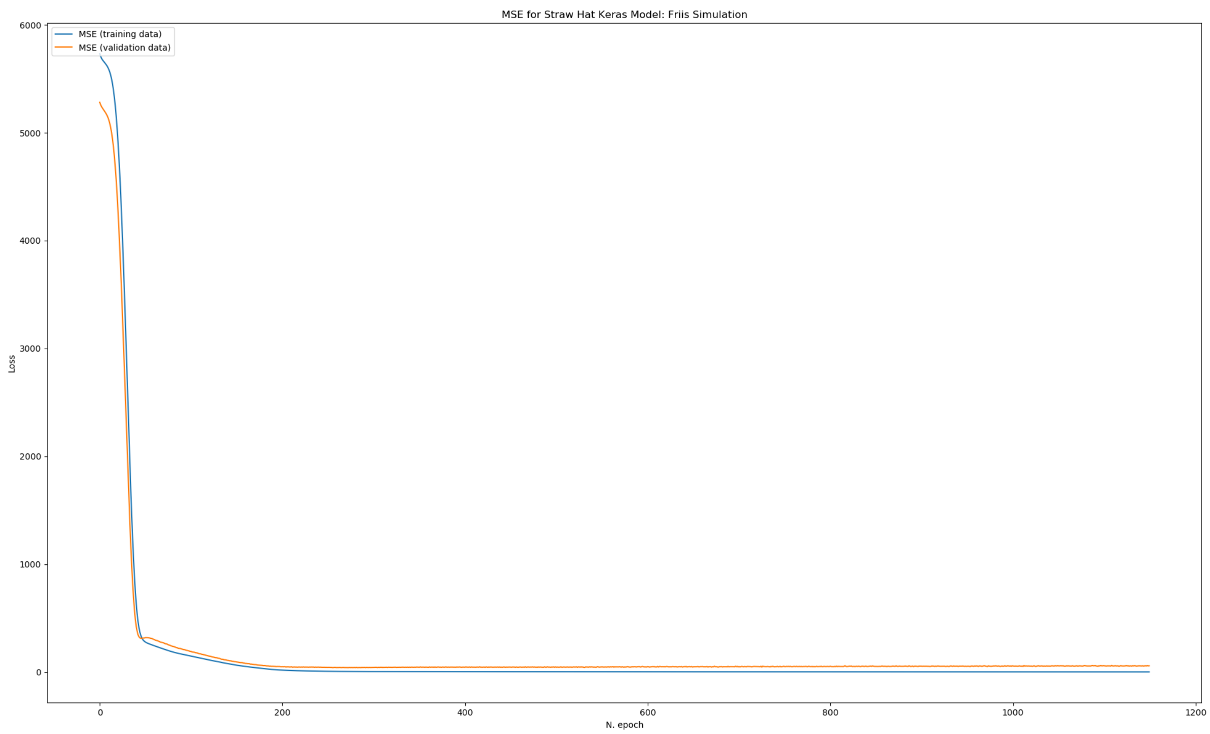

Figure 27- Plot of NN Learning Curve - Friis Simulation

Test 2B - Neural Network assessment using Sim. 02 (JTC w/ Noise), ER – 1, 3, 4

Objective: Use the JTC transmission equation indoor environment path loss RF propagation model to assess our Keras model's ability to handle relevant data.

Expectation: Since the propagation model is slightly more complex, and we have added gaussian noise to the signals, we expect to see a more drawn out learning curve, needing more epochs to assess the problem.

Results: The learning rate of the NN, utilizing Test 2B data:

Early epochs yield high loss but also high learning rate, which slows as it continues beyond 300 epochs. Loss drops from 5,162 down to 300 in the first 27 epochs, then falls to 8.42 by the 255th epoch. Using 25 predictions, our model produced an average difference:

$$\frac{\sum(\text{ABS}(y\_predicted - y\_collected))}{25}$$

of 0.88 meters for the r value, and 11.9 degrees for theta with a standard deviation, of 0.64 meters for r and 6.55 degrees for theta.

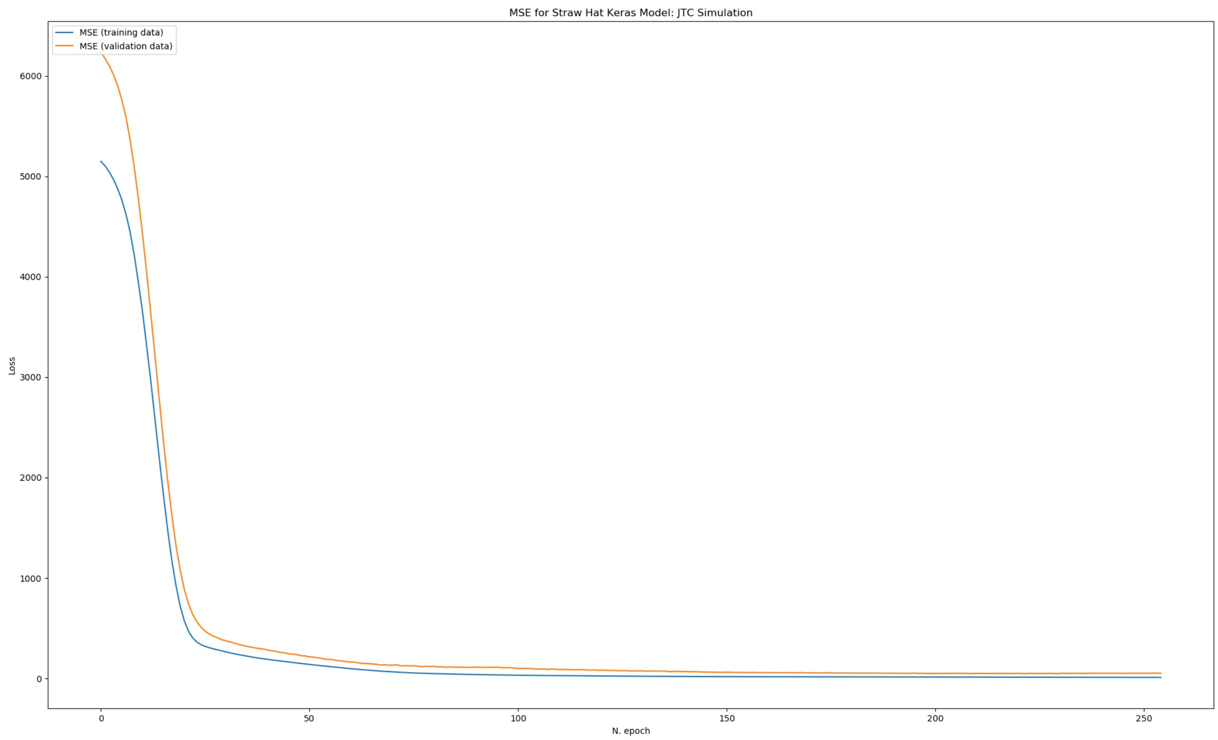

Figure 28 - Plot of NN Learning Curve - JTC Simulation

Test 2 Conclusion:

Test 2A: Batch Size = 32, 2-Layer Model (16w,8w), Epochs = 1,150, Samples = 475, Validation Samples = 25, Training Accuracy = 0.993, Final Training Loss (MSE) = 1.51

Test 2B: Batch Size = 32, 2-Layer Model (32w,16w), Epochs = 255, Samples = 475, Validation Samples = 25, Training Accuracy = 0.97, Final Training Loss (MSE) = 8.42

Friis propagation model performed best after training over 1,150 epochs while the JTC with added noise performed better after 255 epochs. The JTC model had twice as many weights per hidden layer compared to the Friis. The added complexity of the JTC model is evidence that our physical system needs a more complex model (added layers or weights) in order to capture the complexity of the environment. Test 2 results prove that our neural network is capable of achieving the accuracy stated in our original engineering requirement goal of 95% or greater.

Test 3: Test coax cables and find S11 parameters of antennas, ER - 1, 2, 4, 5

Objective: To test the S11 performance characteristics (return loss) of the team's acquired low-loss coax cables and TG.35.8113 wideband 5G/4G dipole terminal antennas. It must be noted that any S11 parameter value below -15 dB per coax cable and S11 parameter value below -10 dB per antenna would signify that the coax cables and antennas being tested are suitable for use.

Setup:

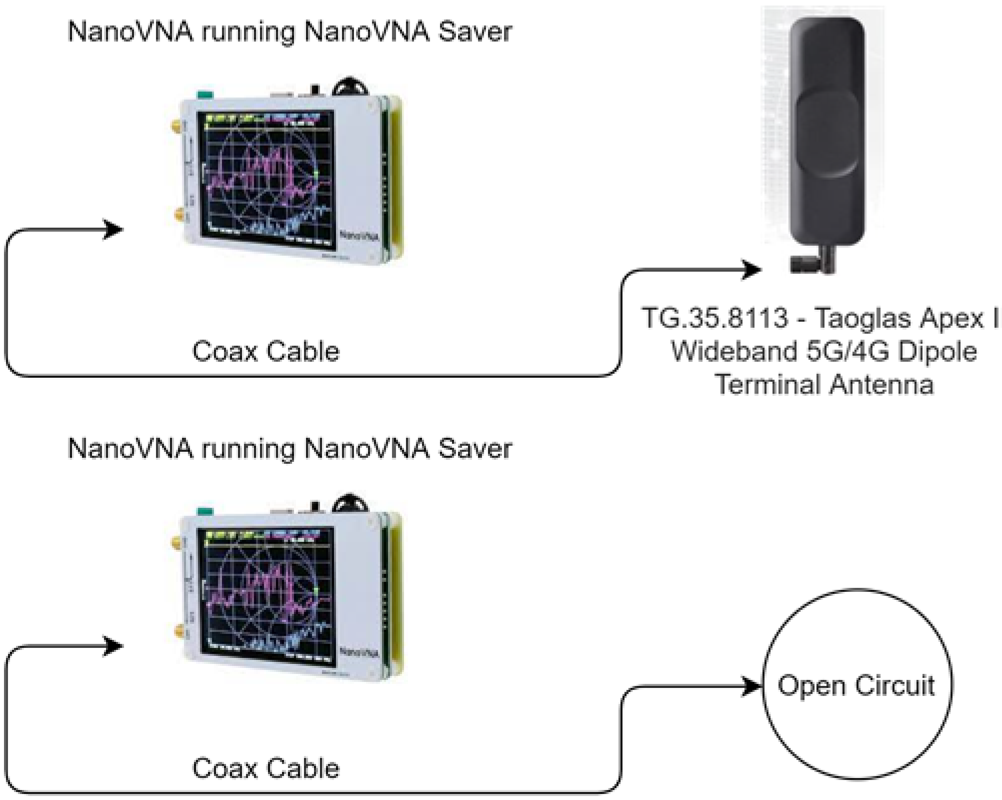

Figure 29 - Test 3 Setup

Results: Using a vector network analyzer (VNA), the team would record the S11 return loss (dB) for a variety of antenna and coax cable configurations, including the S11 return loss of the coax cables, the TG.35.8113 wideband 5G/4G dipole terminal antennas, and a combination of these two pieces of hardware. The results recorded from the VNA can be seen in the figures below.

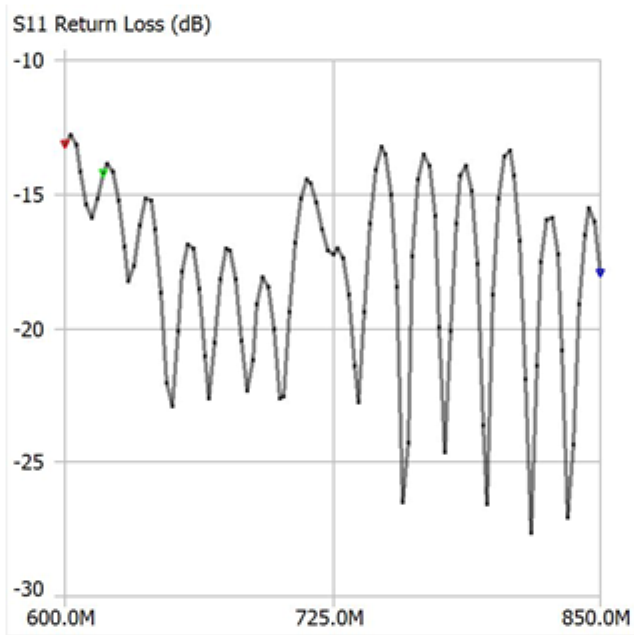

Figure 30- S11 Return Loss (dB) of Antenna (TG.35.8113) with Coax Cable

While viewing Figure 30 above, the team would notice the presence of a standing wave within the results produced by the network analyzer. Connected to the network analyzer, the team would attach both the TG.35.8113 antenna paired with a purchased coax cable. The presence of the standing wave alerted the team that one of the components was flawed.

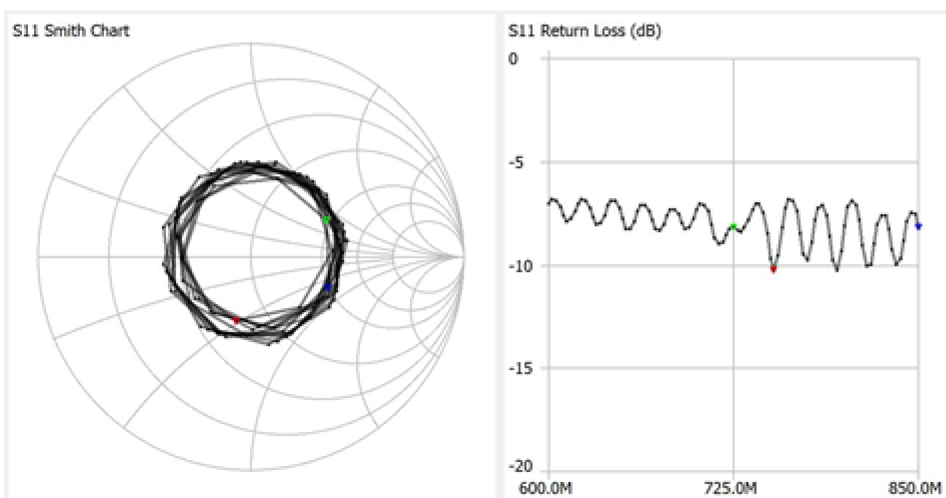

Figure 31- Open Circuit S11 Return Loss (dB) of Coax Cable

To identify the source of the standing wave and the flawed component, the team individually tested each piece of hardware using the VNA. The above figure is the S11 return loss (dB) of the coax cable. In Figure 31 above, we can see that the S11 return loss is not below -15 dB for the coax, signifying that much of the transmitted signal is being reflected internally within the coax cable. Additionally, the return loss of an open circuit is characteristically 0 dB, while the displayed results above are around -6 to -10 dB. Along with these observations, an oscillation can be seen in the return loss plot, showing the standing wave mentioned earlier. The team concluded that these internal reflections within the coax cable were the source of the standing wave, as seen in Figure 30 above.

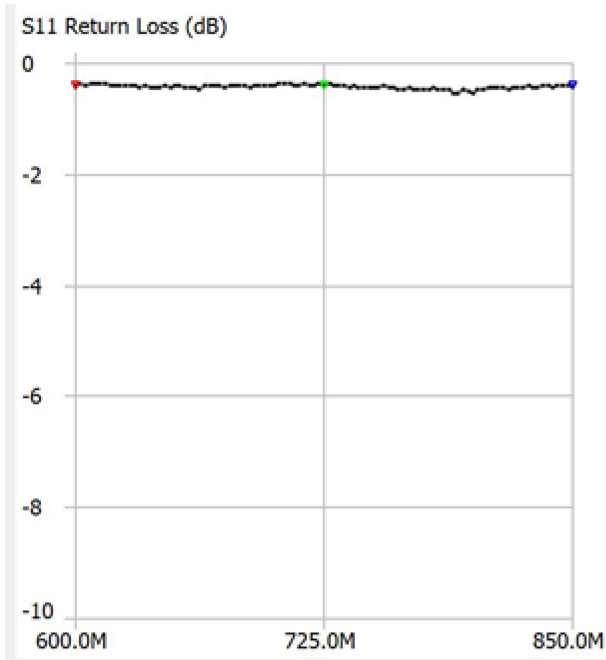

Figure 32- S11 Return Loss (dB) Open Circuit (New Coax)

To verify that the coax cable was the cause of the standing wave, the team would purchase new coax cables and again test them with the VNA. Figure 32 above shows the S11 return loss for the new coax cables, attached to an open circuit. The return loss of an open circuit is characteristically 0 dB, this nearly matches the S11 return loss of the new coax cable, verifying the proper functionality of the team's newly acquired coax cables.

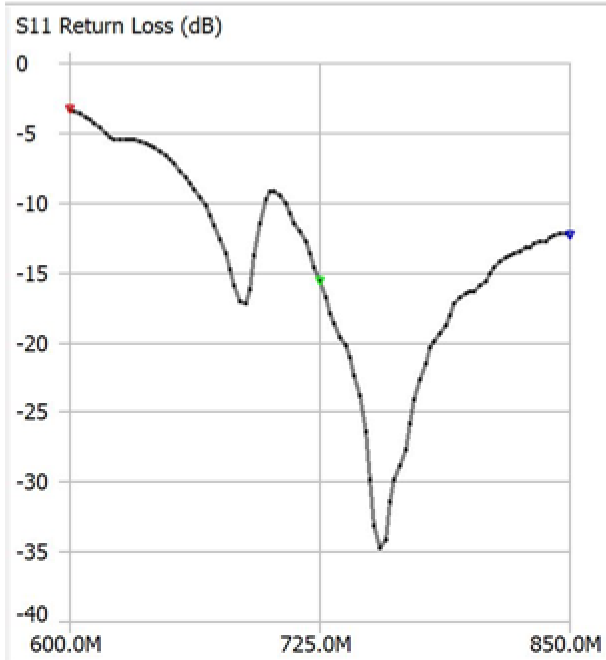

Figure 33- S11 Return Loss (dB) Antenna (TG.35.8113) Paired with New Coax

In order to verify that the standing wave was no longer present within our transmission lines, and that the hardware in use was properly functioning, the team would again attach the newly acquired coax cables, paired with a TG.35.8113 antenna, to the VNA and view the resulting S11 return loss of the newly configured transmission line. By viewing Figure 33 above, our team could see that the standing wave was no longer present, and that the S11 return loss (~ -35 dB) was suitable for our desired carrier frequency of 750 MHz.

Conclusion: While testing the S11 performance characteristics (return loss) of both the team's acquired coax cables and TG.35.8113 wideband 5G/4G dipole terminal antennas, it was discovered that a faulty and defective coax cable was causing the presence of a standing wave within our transmission line. To fix this issue, the team would purchase new coax cables, testing them again on a VNA and evaluating the resulting S11 return loss (dB) of the newly purchased coax cables. Upon testing the new coax cables, the team no longer observed a standing wave within the cables. Once the new coax cables were cleared for use, the team tested the TG.35.8113 paired with the new coax, and observed an S11 return loss of approximately -35 dB for our team's desired carrier frequency with no significant oscillations present, verifying the performances of both the newly purchased coax cables and the TG.35.8113 antennas.

Test 4: Antenna Array clock sync. time delay analysis, ER - 1, 3, 5

Objective: The purpose of this test was to daisy chain the clocks of the five RTL-SDRs together and to verify that they had synchronized clocks. This would imply that this set up would be coherent, and that the SDRs were running on the same clock frequency with little to no latency between them.

Setup: The set up of this test includes the master clock of the Rx shorted between clock pad 1 and 4. The puppets are then shorted between clock pads 1 and 2, with the bypass resistor removed. This configuration is set up such that the puppets have no internal clock and receive an input signal from the master. To test the latency, a Digilent Analog Discovery 2 USB oscilloscope along with the WaveForms software was used to probe each clock input at each receiver to make sure they each had less than a ten nanosecond difference between them.

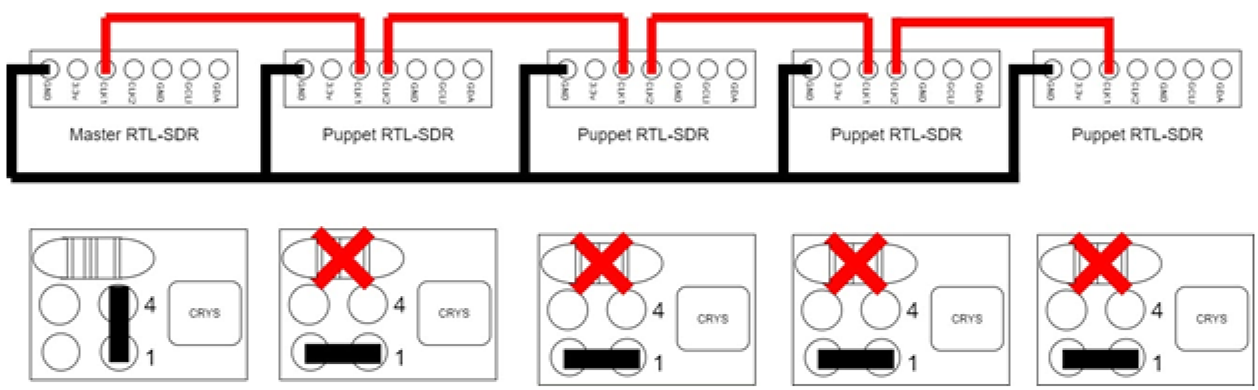

Figure 34-Antenna array soldering schematic.

Results: The analysis showed the SDRs were successfully soldered as specified by the drawn soldering schematic above. They had little to no latency in the period of the clocks after analyzing the oscilloscope measurements. These results can be seen in the figures below.

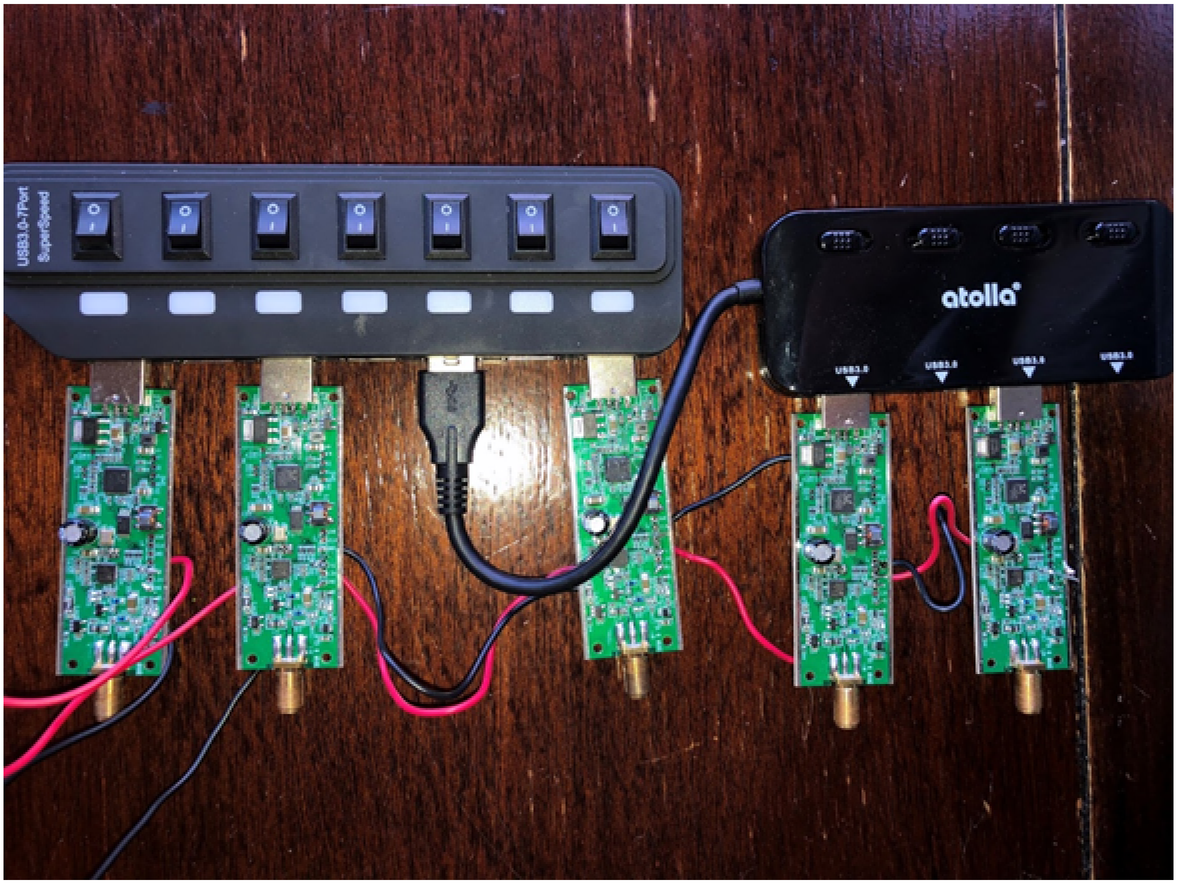

Figure 35 - Clock Sync. 5 RTL-SDRs

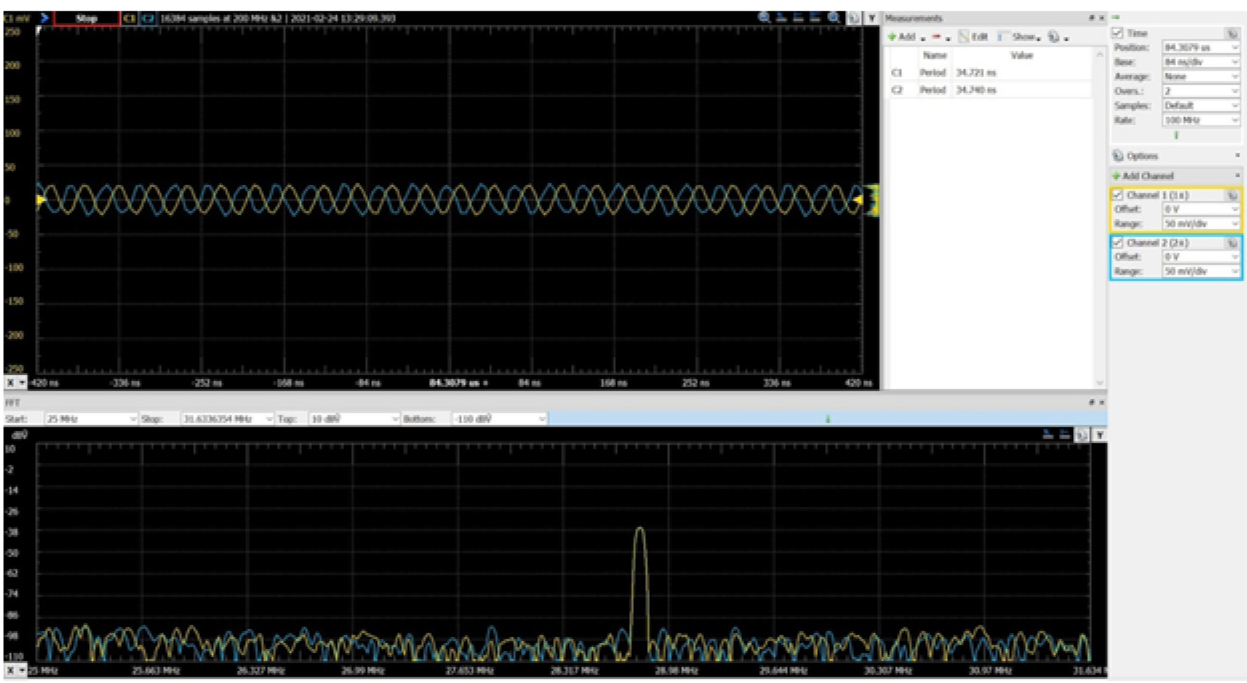

Figure 36 - Master clock (yellow) vs. Puppet clock #1 (blue)

There is a 0.019 ns shift in period between the master and the first puppet receiver.

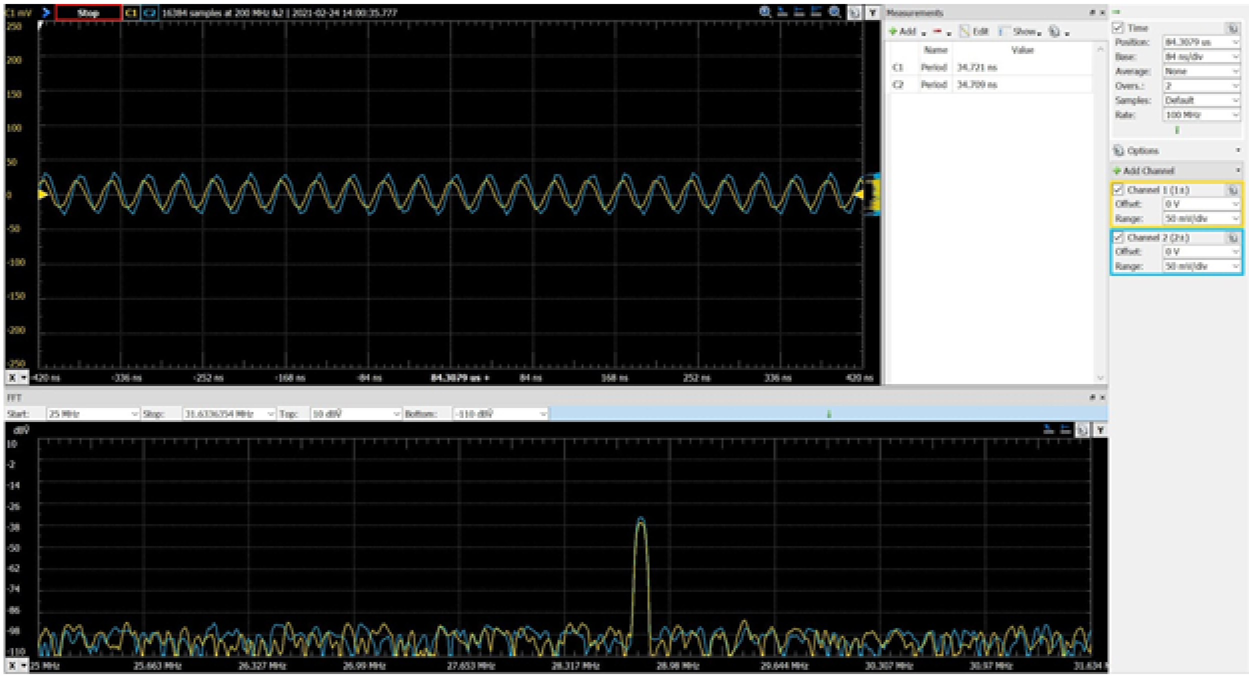

Figure 37 - Master clock (yellow) vs. Puppet clock #4 (blue)

There is a 0.012 ns shift in period between the master and the last puppet receiver.

Test 4 Conclusion: The test was concluded as a success. The team was able to solder all of the RTL-SDRs together with less than a 0.02 ns difference between the clock periods.

Test 5: Flowgraph test with 5 RTL-SDRs, ER - 1, 2, 3, 4, 5

Objective: Run the GNU Radio Flowgraph using the 5-antenna array and verify that the values recorded are consistent with the team's expectations. This test also serves as a functionality test to see if all 5 SDR's can run off the flowgraph simultaneously.

Setup: For this test the team set up the 5-antenna array inside of the training environment, with each antenna connected to an SMA extension coax cable and an RTL-SDR. The SDRs were connected to the ports of an externally powered USB hub. The team expected there to be issues with the amount of output power since the V3 RTL-SDRs consume 275mA of current, while a USB 3.0 port on the RPi 4 can supply 1.2A to peripheral devices. To combat this issue, we used a USB Hub that is capable of drawing 1.2A from a standard 120V wall outlet.

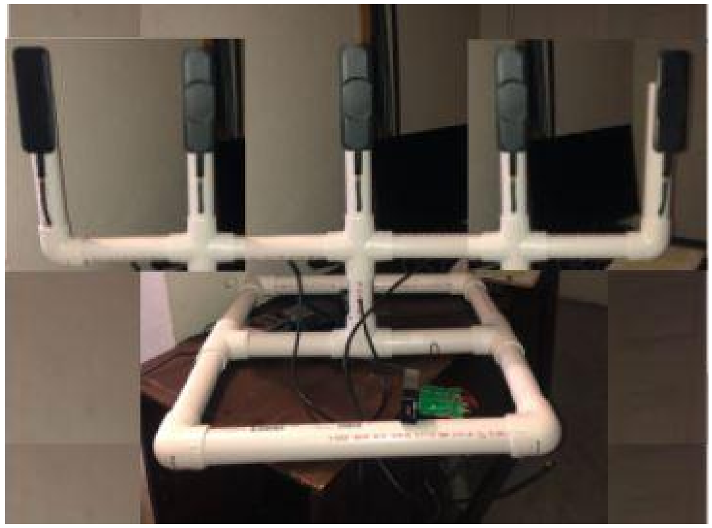

Figure 38 - 5 Antenna Array (5AA)

Results & Conclusion: This test was a failure, causing the team to reduce the antenna array from 5 SDRs to 3 SDRs. Since the 5 SDRs together would require 1.35A of current, we added a USB hub with the capacity to supply 1.2A from a wall outlet. Unfortunately, this did not solve our power issue and the RPi4 was not able to run all 5 SDRs simultaneously. The team's data processing flow graph would run for a few seconds and then the SDRs would disconnect and become unrecognizable by the SoC. We tried running them at the lowest allowable sampling rate of 256KSps to no avail. The team also tried running the flowgraph using a PC with an intel i7 core processor and encountered the same scenario. When the 5 SDRs are run simultaneously without enough power to drive them, they engage a failsafe that disconnects them from the computer and ceases the ability to reconnect until they have been powered off and rebooted. This is to protect the PCB and memory on board the SDRs.

Test 6: The Seer System Calibration

Test 6A - Antenna array test environment power calibration, ER - 2, 5

Objective: The objective of this test was to evaluate the performance of The Seer system and to collect preliminary data within the system's test environment in order to calibrate the Rx. The collected data includes the relative signal strength of the received low-band 5G signal (dBW) for each of the three receiving antennas that form The Seer's linear antenna array.

Setup: Both the Tx (HackRF One and Port-A-Pack) and receiving antenna array were elevated off the ground and set to equal heights. A total of three trials were conducted within The Seer's test environment (indoor-urban) in which the receiving antenna array remained in a fixed location, while the Tx distance increased by 1 meter, the furthest distance being 6 meters. At each incremental distance, three measurements were taken, in which both the Tx and Rx sent and received data processed by our data extraction flow graph. The team measured three times to account for any system or data variance that may have occurred throughout testing.

Porta Pack Settings: Gain: 15 dB, Waveform (Signal Source): 1 kHz Constant Wave Source, Carrier Frequency: 750 MHz

Results:

Antenna 1:

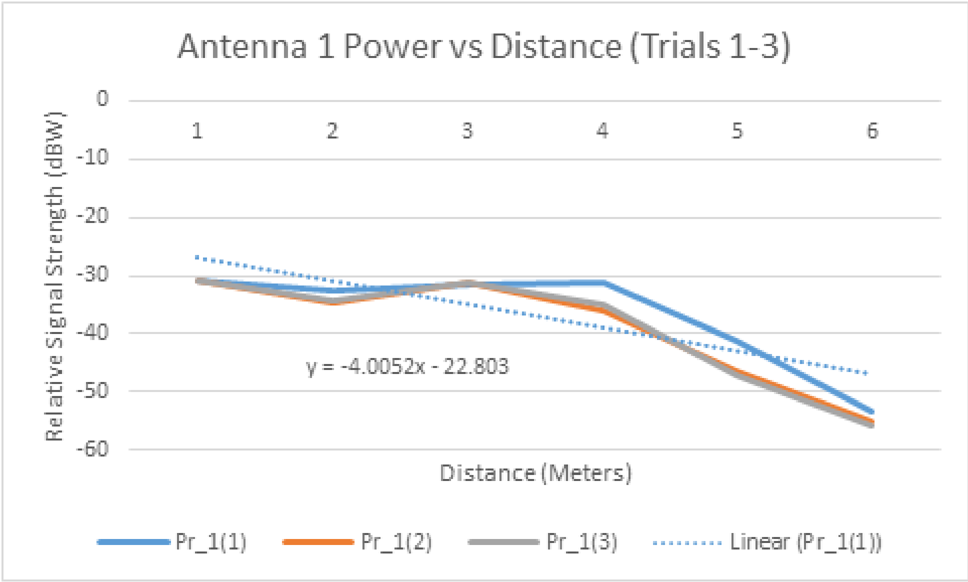

Figure 39 - Antenna 1 Power vs Distance (Trials 1-3)

Standard Deviation (Trials 1-3): 8.89 dBW

System Variance (Trials 1-3): 79.04 dBW

Path Loss Exponent (Trial 1): -4.00

Antenna 2:

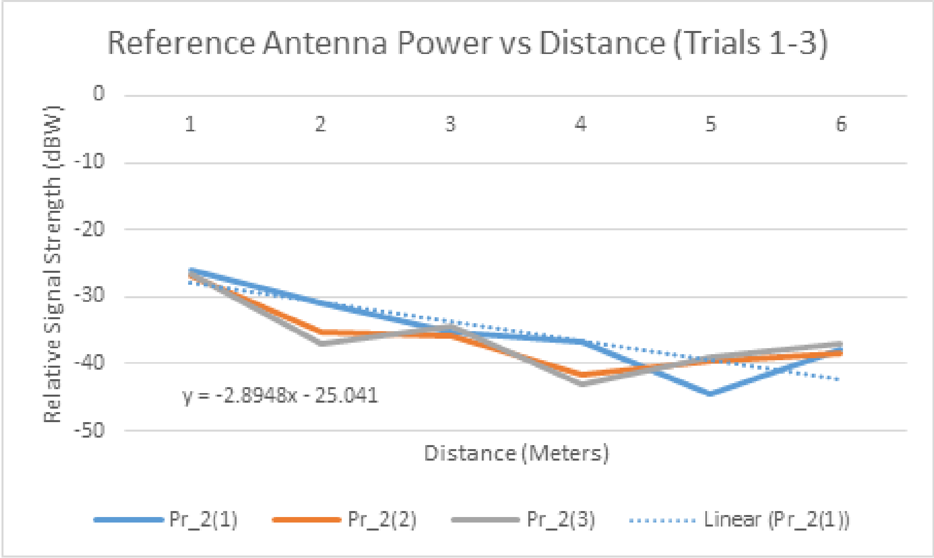

Figure 40 Reference Antenna Power vs Distance (Trials 1-3)

Standard Deviation (Trials 1-3): 5.22 dBW

System Variance (Trials 1-3): 27.25 dBW

Path Loss Exponent (Trial 1): -2.89

Antenna 3:

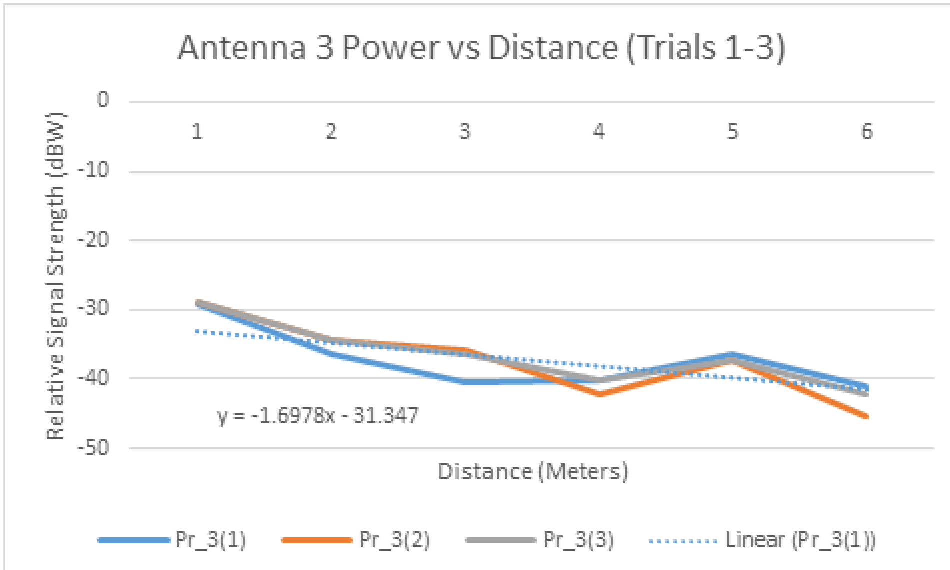

Figure 41 - Antenna 3 Power vs Distance (Trials 1-3)

Standard Deviation (Trials 1-3): 4.59 dBW

System Variance (Trials 1-3): 21.03 dBW

Path Loss Exponent (Trial 1): -1.70

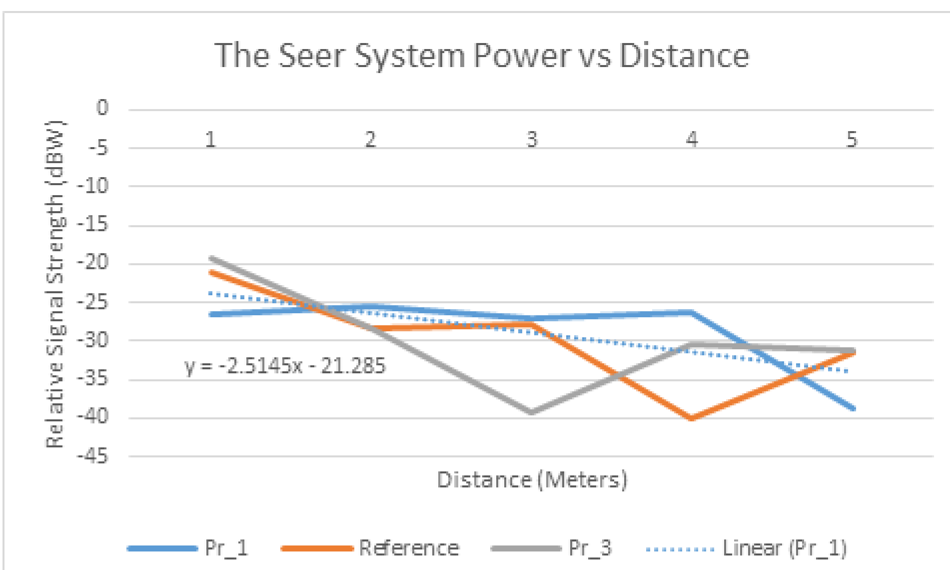

Figure 42 -The Seer System Power vs Distance

Standard Deviation: 5.88 dBW

System Variance: 34.56 dBW

Path Loss Exponent (Trial 1): -2.51

Antennas (1-3)	Path Loss Exponent (dBW/m)
Antenna 1	4.005 +/- 0.709
Antenna 2 (Reference)	2.174 +/- 0.489
Antenna 3	2.260 +/- 0.537

Table 5 - The Seer Antenna Array Path Loss Exponent (Each Antenna)

Conclusion: The results for each receiving antenna of The Seer's linear antenna array can be seen above, as the relative received signal strength of the transmitted low-band 5G signal is displayed for each trial (3x) and distance (1-6 meters). The results pertaining to the received signal strength are consistent with the test environment the system will be operated in (indoor-urban), and are consistent with the environmental factors found within this test environment. The received signal strengths for each receiving antenna (3) in dBW, had relatively low system variance and standard deviation between the three trials.

Test 6B - Antenna array test environment phase calibration, ER - 2, 5

Objective: The goal for this test is to confirm that our antenna array relative phase difference data will be useful to the neural network. The project design takes the phase of the middle antenna and uses it as a reference, dividing the complex phasors of the other antennas by the reference resulting in a subtraction of the phases, a relative phase difference. The relative phase difference holds information about the angle of the incoming signal using the radiation pattern of the antenna. This data is important for the neural network to establish an accurate model.

Setup: For this experiment the team set up the Tx at varying distances and angles relative to the Rx. The position of the Tx was analyzed how the phase changed accordingly. Since this is the first test the team conducted with the antenna array regarding phase information, the goal was to look for an appropriate difference/similarity in phases depending on the position of the antenna.

Figure 43 - Final 3 Antenna Array (3AA)

Results: This test was a failure, causing the team to drop the relative phase difference from the N.N. inputs. After conducting the test for multiple positions in the environment it can be seen in the figures below that the phase changes with time, which overshadowed any relevant phase data from the array. In the graphs it is depicted that the phase value is not constant for a nonmoving transmitter. Prior to unwrapping the data, we can see the phase is cycling between 0 and pi without stabilizing at any particular value.

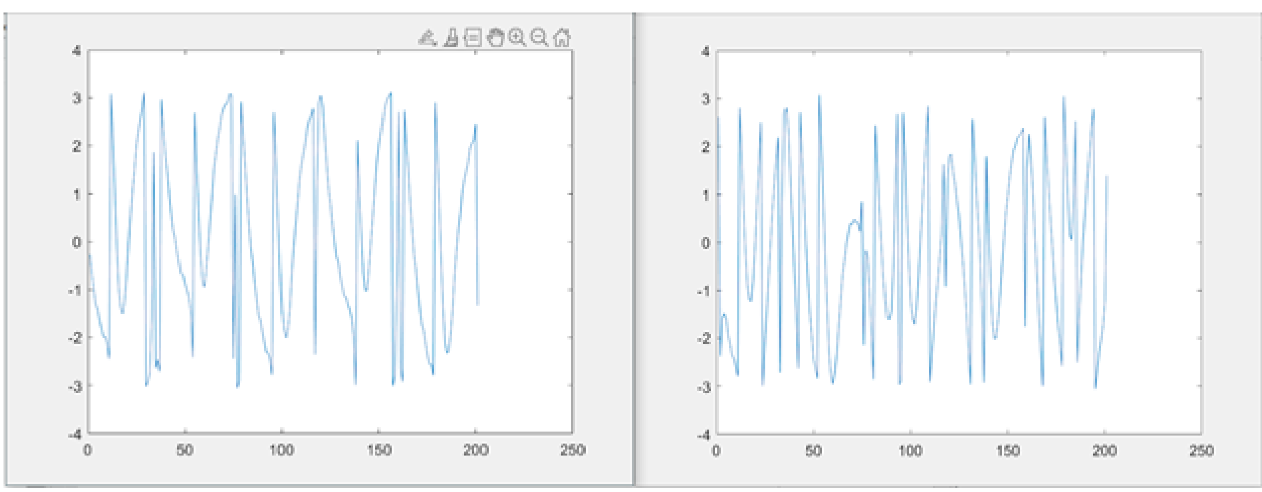

Figure 44 - Unprocessed Phase Data

After unwrapping the data, the phase values are constantly dropping. This is expected by the team's research to be an unsynchronized connection between the clock and the sampling. Since the clock is out of sync with the sampling, we never see a stable phase. The expected value being a constant value across the graph for a nonmoving Tx.

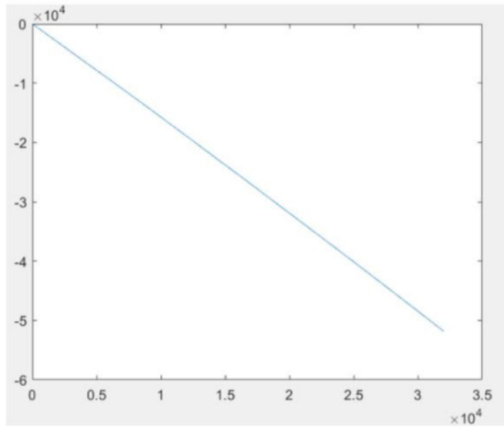

Figure 45 - Unwrapped Phase Data

The device takes samples at different times in the sinusoids rise and fall making the output useless for the team. Each antenna's graph follows a similar trend. The problem does not lie with the connection of the clocks, since an introduced time delay due to long wires or bad connections would manifest as a constant shift in the phase of an antenna, which could be accounted for using signal processing techniques and GNU Radio. The problem is a combination of the low-quality of the RTL-SDR clock, which was being used by the master SDR to drive the other puppet SDRs, and overheating due to the removal of the heat sinks on the SDRs that was necessary for connecting the clocks. The SDRs already heat up, but without the aluminum casing that acts as a heat sink this heat can have a more significant effect on the components of the circuit. The accumulation of these issues led to a phase drift that rendered our extracted phase data completely useless.

Test 6B Conclusion: As a result of the SDR's not being able to provide any relevant phase information, the team had to abandon the idea of using phase in the project all together. Fixing the issue could have been very time consuming and expensive all without guaranteeing a solution to unreliable hardware. Dropping the phase meant that the team's neural network would have only 3 inputs, the three antenna magnitudes, to model the environment. Because of this reduction in input data, The Seer team decided to reduce the target accuracy to 90% from the previous goal of 95%. Deep learning is all about the amount of data that you provide the neural network with and limiting the amount of data hinders the network's ability to learn.

Test 7: GUI display test and result delay-time analysis, ER - 1, 6

Objective: A graphical user interface is used to display both the input and output data. The inputs, the corresponding predicted outputs, and the average difference of the r and theta values are displayed alongside a polar plot showing the related Tx location relative to the Rx. This test serves as a verification for ER-7, reassuring the team that the GUI works well with the code used for the neural network.

Setup: The GUI is set up such that data is extracted from CSV files provided by the neural network. The GUI takes the predicted values from the neural net, the correct values from the validation data, and the input data from the antenna array, and displays them along with the average difference of r and theta for all of the validation data. The GUI also graphically displays the predicted values in polar coordinates corresponding to the magnitude and angle, as well as showing the system accuracy.

Results: The GUI was able to display the data and the corresponding graph to the predicted values in sleek green and black colors. The average difference was calculated by taking a sum of the difference between each predicted r or theta value and the measured value, and then dividing that sum by the number of validation measurements. For our batch of validation measurements, the average difference for the r value was 0.65 m, and the average difference for theta was 22 degrees. Our system accuracy was calculated by setting an acceptable threshold of less than 1m difference for r and less than 45 degrees difference for theta. If the predicted value was within this acceptable error for both r and theta, then the prediction was considered accurate. The percent of validation measurements that met these criteria relative to the total number of predictions is displayed as our system accuracy.

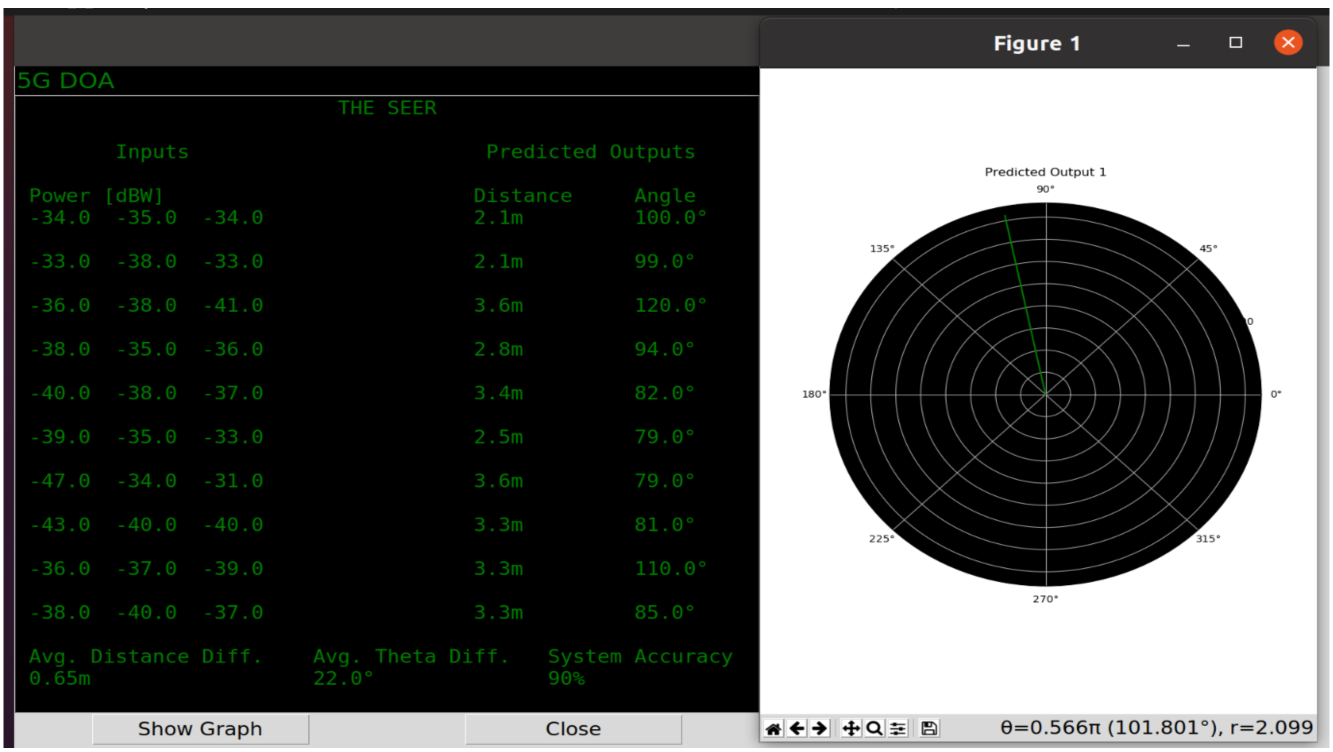

Figure 46 - Graphical User Interface (GUI)

Conclusion: The team was able to implement a functioning user interface that correctly displays the corresponding data and polar plots in under 60 seconds, satisfying ER-7.

System Results

With the reduction from 5 antennas to 3, the neural network lost a significant amount of data, reducing the number of inputs from 8 to 5, impacting our systems ability to accurately estimate the DOA. With the removal of the relative phase difference, the neural network lost even more data, reducing the number of inputs from 5 to 3. Using this tensor made up of the 3 magnitudes, our system was able to predict the DOA. with a system accuracy of 90%. Our system accuracy was based on the percentage of validation samples that passed our teams predetermined metric:

r difference < 1 meter

rheta difference < 45 degrees

This is impressive and speaks to the ability of our deep learning algorithm to build a model representative of our complex inverse problem.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 32)                128
_________________________________________________________________
dense_1 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 66
=================================================================
Total params: 1,250
Trainable params: 1,250
Non-trainable params: 0
_________________________________________________________________
```

Figure 47 - Model Summary

Our final neural network model consisted of an input layer with 3 nodes (corresponding to the 3 magnitudes) feeding into two hidden layers, each with 32 weights. Both hidden layers used ReLU (rectified linear unit) as the activation function, which is an introduced nonlinearity used to escape local minima that is applied to the weighted sum at each layer. The final layer was a 3 node output layer corresponding to (r,theta) with a linear activation function. Using simulations, we expect that the same arrangement would work well, yielding even higher accuracy, using the 5-antenna array with an 8 input tensor consisting of both magnitude and relative phase difference from 4 antennas and a reference. The codes corresponding to both our 5AA and final 3AA can be found on our GitHub.

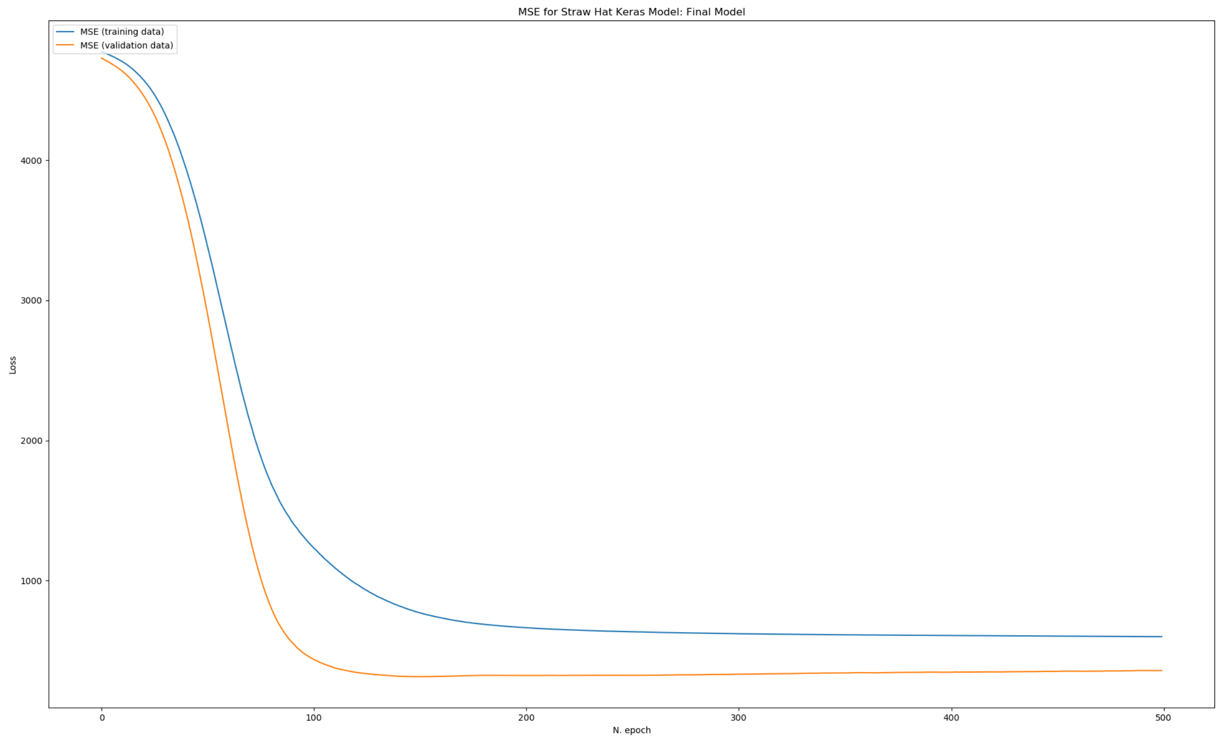

Figure 48 - Final Model Learning Rate

Future of The Seer

To meet our original set of engineering requirements the team would need to design an external power supply that also functions as a USB hub capable of powering 5 RTL-SDRs (V3 or V4) with 275 mA of current each, allowing bidirectional data flow to and from USB 3.0 port. In the design, a synchronization of SDR clocks via highly accurate, temperature steady external clock source would need to be included. The team would need to ensure that the SDRs are truly synchronized, forming a coherent Rx without any phase drift. Using this external power supply to successfully run our 5-antenna flowgraph, over 500 measurements worth of data would be run through the 5-antenna neural network, followed by testing on over 100 validation measurements, accomplishing >95% system accuracy up to 12m from the Rx, satisfying the teams original ER 1 & ER 2 goals. We created a GitHub repository that can be found on the project website with all of our project codes for both the 3AA and 5AA in hopes that a future group of engineering students will continue the work we have started.

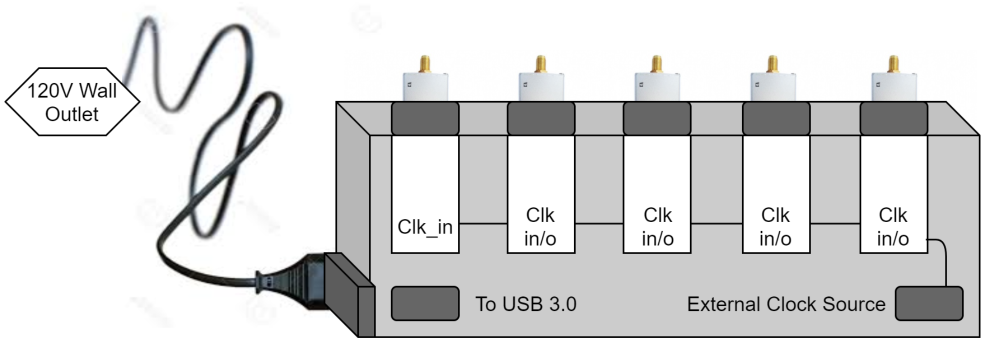

Figure 49 - Potential External Power Supply

Engineering Ethics and Our Project

An ethical design is a project that is created in accordance with the bylaws set up by the National Society of Professional Engineers code of ethics. The guidelines include holding safety paramount, only performing services we are qualified to do, to be honest in our claims, and to conduct ourselves honorably and ethically. In our project we have been nothing but transparent by keeping all the data as is and not cherry picking anything that might support our claim, rather, we are actively attempting to disprove our hypothesis ensuring that the scientific method is being followed. Along with the National Society of Professional Engineers or the IEEE we also adhere to the laws of the FCC. Since our project utilizes the radio frequency spectrum, we must adhere to the federal laws in place to be sure we do not cause any interference to the people around us with our project. To be sure of that, we maintained a transmitted signal below one watt of power and used a modulation scheme when transmitting.

With the ongoing climate crisis, we hope that our project will help by allowing cellular companies to optimize power when transmitting RF waves to their customers. With less power use comes a much lower carbon footprint left by our cellular utility companies. Since our project is a proof of concept of a system there is no loose end on the manufacturing. We do not contribute to the use of silicon wafer plants or other pieces of the technological global supply chain. Contrary to increasing consumerism, we hope that the system reduces the use of technologies that need more hardware, more cellular devices, and base stations. Our project also is most useful in the beamforming area of 5G. The current beamforming technique expels more concentrated areas of radiation under a smaller transmission path. Less radiation in the air means less disruption to wildlife that use the earth's electro-magnetic polarizations to travel, e.g butterflies, wasps, bees, etc. We also hope that our project can help better educate the population about the truth of 5G and to steer away from the conspiracy theories that have been circulating on the internet and on the news against the new technology.

References

[1] - George Godby,"Using GNU Radio for Signal Phase Measurements", Department of Engineering, Michigan State University, East Lansing, Michigan, United States, (March 27 2014). https://www.egr.msu.edu/classes/ece480/capstone/spring14/group02/docs/Application%20Note%20-%20Phase%20George%20Godby%20Team%202.pdf

[2] - H. Li, S. Sun and J. Wang, "Direction of Arrival Estimation Using Amplitude and Phase Information in Low-Profile MIMO Arrays," in IEEE Transactions on Antennas and Propagation, vol. 66, no. 11, pp. 6457-6462, Nov. 2018

[3] - Khan, Usman. et al. "Neural Networks." Academia.edu, www.academia.edu/Documents/in/Neural_Networks. (Oct. 31 2020)

[4] - Schulz, Karen. "Verizon and Ericsson Team up to Deploy Massive MIMO." www.verizon.com/about/news/verizon-and-ericsson-team-deploy-massive-mimo. (May 14, 2021)

[5] - Brownlee, J. "Loss and Loss Functions for Training Deep Learning Neural Networks" https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/.(May 14, 2021)

[6] - Team, K. "Keras documentation: Introduction to Keras for Engineers" from https://keras.io/getting_started/intro_to_keras_for_engineers/ (May 14, 2021)

[7] - Sklearn."preprocessing.scale".https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.scale.html (May 14, 2021)

[8] - Grover, P. "5 Regression Loss Functions All Machine Learners Should Know" .https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0.(May 14, 2021)

[9] - Brownlee, J."Your First Deep Learning Project in Python with Keras Step-By-Step". https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/(May 14, 2021)

[10] - Stolnikov, D. "Gr-osmosdr" https://osmocom.org/projects/gr-osmosdr/wiki/GrOsmoSDR. (May 14, 2021)

[11] - Tindall, Leonora. "HackRF Tripups with GNURadio". https://nora.codes/post/hackrf-tripups-with-gnuradio/.(May 14, 2021)

[12] - Connolly, Paul. "HackRF One coverage" https://hackrf-dev.greatscottgadgets.narkive.com/JLDbvEA9/hackrf-one-coverage. (May 14, 2021)

[13] - Brownlee, Jason. “How to Save and Load Your Keras Deep Learning Model.” machinelearningmastery.com/save-load-keras-deep-learning-models/. (May 14, 2021)

[14] - K, Sam. “Analog Discovery 2 Specifications.” reference.digilentinc.com/reference/test-and-measurement/analog-discovery-2/specifications. (May 14, 2021)

[15] - Matplotlib development team. “Polar Plot”. matplotlib.org/stable/gallery/pie_and_polar_charts/polar_demo.html. (May 14, 2021)

[16] - Jared L. Deutsch, Clayton V. Deutsch, Latin hypercube sampling with multidimensional uniformity, Journal of Statistical Planning and Inference, Volume 142, Issue 3, 2012