



Engineering and Physics Clinic

Final Report for
MIT Lincoln Laboratory

Mid Year Technical Memorandum for Drones with Software-Defined-Radio

November 13, 2024

Team Members

Nandini Garg and Sean Wu (fall)
Noah Haig (fall Project Manager)
Domenico Ottolia (spring Project Manager)
Kaanthi Pandhigunta and Mason Wray (spring)
Harry Sanchez

Advisors

Jason Gallicchio
Peter N. Saeta

Liaison

Alexia Schulz

The aim of the project is to use an antenna array on a fixed-wing drone to estimate the direction from which a source wave has come and to combine that information with the drone's position and orientation to develop a set of estimates of the source position that can be updated and refined by modifying the flight plan of the drone. Leaving aside the challenges of mounting an antenna array and companion electronics on the drone, we need a way of turning positions, orientations, and antenna phases into estimates of the position of the source of the radio-frequency waves detected by the antennas.

In the current section, we develop a method for taking a known set of antenna positions and phases to estimate the wave vector \mathbf{k} of the source wave, working first in a $\xi\eta\zeta$ coordinate system tied to the antenna array. From that system, we transform first into airplane coordinates XYZ and then to earth coordinates xyz to yield a unit vector $\hat{\mathbf{u}}$ pointing from the plane's position toward the source. A subsequent section will then address how to combine pairs of such observations into estimates of the source's position.

1 Analyzing Antenna Phases

Given an array of 4 antennas occupying the $\xi\eta$ plane, with antenna positions given by \mathbf{a}_n for $n = 1, \dots, 4$, and an incident plane wave with wave vector \mathbf{k} , the (ideal) phases that will be measured on the antennas are

$$\phi_n = \mathbf{k} \cdot \mathbf{a}_n = \boldsymbol{\kappa} \cdot \mathbf{a}_n \quad (1)$$

where $\boldsymbol{\kappa}$ is the component of \mathbf{k} in the $\xi\eta$ plane. To determine a unit vector pointing towards the source (in the $-\mathbf{k}$ direction), we need to combine the phase

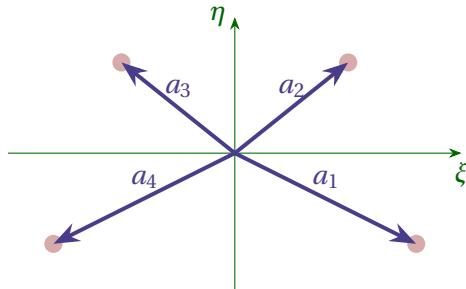


Figure 1 We assume an array of antennas occupying the $\xi\eta$ plane, with antenna position given by vectors \mathbf{a}_n .

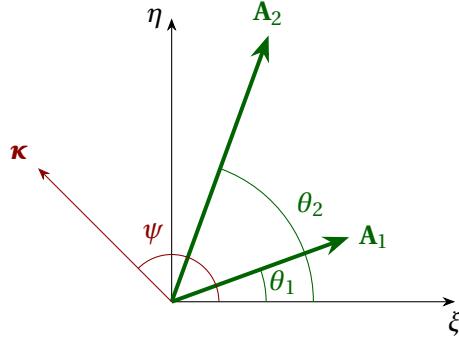


Figure 2 Determining the angle ψ that indicates the orientation of the in-plane component of the incident \mathbf{k} vector.

information to determine $\boldsymbol{\kappa}$, infer the out-of-plane component of the wave vector,

$$\mathbf{k} = \boldsymbol{\kappa} + k_{\perp} \hat{\zeta} \quad (2)$$

and then transform this vector from antenna coordinates to airplane coordinates $X Y Z$, with X directed from the tail to the nose of the drone, Y pointing from the left wing to the right wing, and Z pointing downward. From the plane coordinate system, we then transform to earth coordinates in which x points east, y points north, and z points up.

The orientation of the plane is given by the (yaw, roll, pitch) angles. According to https://en.wikipedia.org/wiki/Euler_angles, the most common definition of the Tait-Bryan angles used in navigation is (Z_0, X_1, Y_2) (yaw, roll, pitch), meaning that the first rotation from the earth system is by the yaw angle about the vertical, followed by a roll about the plane's X axis, and finally a pitch about Y . The matrix that rotates a vector from earth into plane coordinates is

$$M_p = \begin{pmatrix} c_0 c_2 - s_0 s_1 s_2 & -c_1 s_0 & c_0 s_2 + s_0 s_1 c_2 \\ s_0 c_2 + c_0 s_1 s_2 & c_0 c_1 & s_0 s_2 - c_0 s_1 c_2 \\ -c_1 s_2 & s_1 & c_1 c_2 \end{pmatrix} \quad (3)$$

where c_n is the cosine of the n th angle, s_n is the sine of the n th angle, and I have shifted indices by one to conform to Python's zero-based numbering convention.

The phase of each signal has an offset with respect to the (unknown) phase of the wave at the origin, but by comparing the phases of a pair of antennas, this unknown phase can be eliminated. Let $\mathbf{A}_{ij} = \mathbf{a}_j - \mathbf{a}_i$. Then,

$$\phi_{ij} = \mathbf{A}_{ij} \cdot \boldsymbol{\kappa} \quad (4)$$

Now a slight shift in notation. In the figure below, \mathbf{A}_μ is the difference between two antenna positions and \mathbf{A}_v is the difference between another pair (one of which may be in common with the first pair). We seek the angle ψ that describes the in-plane component $\boldsymbol{\kappa}$, as illustrated in Fig. 2.

Then

$$\phi_\mu = A_\mu \boldsymbol{\kappa} \cos(\psi - \theta_\mu) \quad (5)$$

$$\phi_v = A_v \boldsymbol{\kappa} \cos(\psi - \theta_v) \quad (6)$$

Let

$$\rho_{\mu v} = \frac{\phi_\mu A_v}{\phi_v A_\mu} = \frac{\cos(\psi - \theta_\mu)}{\cos(\psi - \theta_v)} \quad (7)$$

from which we readily obtain

$$\tan \psi = \frac{\cos \theta_\mu - \rho_{\mu v} \cos \theta_v}{\rho_{\mu v} \sin \theta_v - \sin \theta_\mu} \quad (8)$$

Each pair of antenna pairs can be used to determine a value of ψ ; from the statistics on these, we can get a sense for the quality of the measurement. However, there remains an ambiguity of π in the value of ψ determined by Eq. (8). To resolve this ambiguity, we return to Eq. (5) and evaluate

$$\boldsymbol{\kappa} = \frac{\phi_\mu}{A_\mu \cos(\psi - \theta_\mu)} \quad (9)$$

The correct value of ψ is the one that yields a positive value for $\boldsymbol{\kappa}$.

A value of ψ can be determined for each pair of antenna pairs, except pairs that are parallel. Once a value for $\psi \pm \delta\psi$ has been determined, we can return to the determination of the out-of-plane component of \mathbf{k} , which we will assume is positive (since the drone should be above the source). I think the former. Again, from the spread in values of $\boldsymbol{\kappa}$, we get a value and uncertainty, from which we can estimate $k_\zeta = \sqrt{k^2 - \boldsymbol{\kappa}^2}$. The direction to the source is then identified as

$$\hat{\mathbf{u}} = -\frac{\boldsymbol{\kappa} + k_\zeta \hat{\boldsymbol{\zeta}}}{k} \quad (10)$$

which can then be transformed in plane coordinates and then earth coordinates, using the inverse of M_p given by Eq. (3).

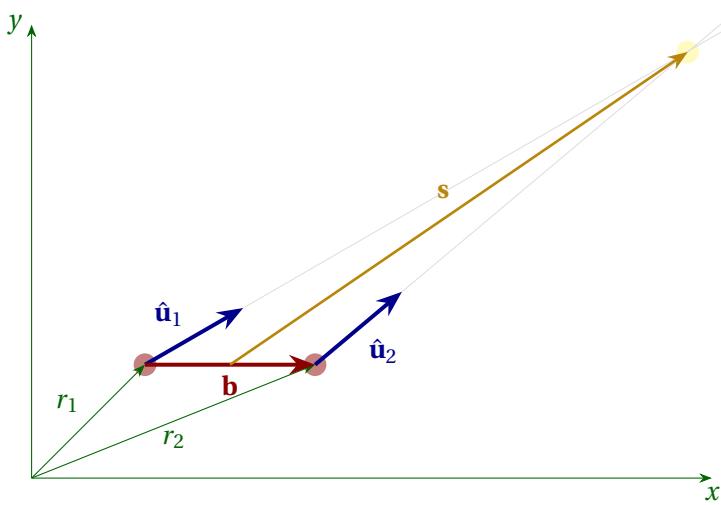


Figure 3 Combining two observations to estimate the position of the source (shown in a yellow circle). The positions of the observations are marked with the red circles, which are separated by the baseline vector \mathbf{b} . The vector \mathbf{s} goes from the midpoint of the two observation positions to the inferred position of the source. The sensitivity of the determination is greater the larger b and the closer to 90° the angle between \mathbf{b} and \mathbf{s} .

2 Estimating Source Position

As the drone flies it acquires observations that include position \mathbf{r}_n and unit vectors pointing towards the source $\hat{\mathbf{u}}_n$. Using pairs of observations, we can estimate the position of the source.

The ray of points indicated by a unit vector $\hat{\mathbf{u}}_1$ emanating from \mathbf{r}_1 is $\mathbf{r}_1 + \lambda_1 \hat{\mathbf{u}}_1$ for positive real numbers λ_1 , and similarly for a second pair $\mathbf{r}_2 + \lambda_2 \hat{\mathbf{u}}_2$. The directed distance between any pair of points, one on each ray is $\mathbf{s} = (\mathbf{r}_1 + \lambda_1 \hat{\mathbf{u}}_1) - (\mathbf{r}_2 + \lambda_2 \hat{\mathbf{u}}_2)$. Therefore, the square of the distance between them is

$$\mathbf{s} \cdot \mathbf{s} = (\mathbf{r}_1 - \mathbf{r}_2 + \lambda_1 \hat{\mathbf{u}}_1 - \lambda_2 \hat{\mathbf{u}}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2 + \lambda_1 \hat{\mathbf{u}}_1 - \lambda_2 \hat{\mathbf{u}}_2) \quad (11)$$

Define $\boldsymbol{\rho} = \mathbf{r}_1 - \mathbf{r}_2$. Then

$$s^2 = \rho^2 + 2\boldsymbol{\rho} \cdot (\lambda_1 \hat{\mathbf{u}}_1 - \lambda_2 \hat{\mathbf{u}}_2) + \lambda_1^2 + \lambda_2^2 - 2\lambda_1 \lambda_2 \hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2 \quad (12)$$

At the minimum value of s^2 we must have

$$\frac{\partial s^2}{\partial \lambda_1} = 0 \quad \text{and} \quad \frac{\partial s^2}{\partial \lambda_2} = 0 \quad (13)$$

so

$$\begin{pmatrix} 1 & -\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2 \\ -\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\rho} \cdot \hat{\mathbf{u}}_1 \\ \boldsymbol{\rho} \cdot \hat{\mathbf{u}}_2 \end{pmatrix} \quad (14)$$

If $\hat{\mathbf{u}}_1$ is parallel to $\hat{\mathbf{u}}_2$, the matrix is not invertible and we cannot use the two observations to estimate the source position. Otherwise, solving for the values at closest approach gives

$$\lambda_1 = \frac{\boldsymbol{\rho}}{1 - (\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2)^2} \cdot [\hat{\mathbf{u}}_2(\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2) - \hat{\mathbf{u}}_1] \quad (15)$$

$$\lambda_2 = -\frac{\boldsymbol{\rho}}{1 - (\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2)^2} \cdot [\hat{\mathbf{u}}_1(\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2) - \hat{\mathbf{u}}_2] \quad (16)$$

The location of their quasi-intersection is then

$$\mathbf{r}_{\text{cl}} = \frac{\mathbf{r}_1 + \mathbf{r}_2 + \lambda_1 \hat{\mathbf{u}}_1 + \lambda_2 \hat{\mathbf{u}}_2}{2} \quad (17)$$

and their minimum separation is

$$S = |\mathbf{r}_1 - \mathbf{r}_2 + \lambda_1 \hat{\mathbf{u}}_1 - \lambda_2 \hat{\mathbf{u}}_2| \quad (18)$$

A more credible determination of position has a small value of S .

2.1 Example

As an illustration of the soundness of this analysis (at least in the absence of noise), Fig. 4 shows an example flight in which the drone starts at the origin heading at yaw = 10° at 5 m/s, taking measurements every 10 seconds. The gray lines show the directions determined from the noiseless phase “measurements” and the yellow circles show the (overlapping) position determinations arising from each pair of observations.

2.2 Noise

In the previous section, we presumed that all measurements were perfect, which is far from the case. The drone’s position \mathbf{r}_n will have some measure of uncertainty, as will its orientation angles. However, we suspect that the dominant source of error will be the phase measurements from which the direction to the source is determined. Errors in these propagate to unreliable determination of the unit vectors $\hat{\mathbf{u}}_n$ that point to the source. Developing a robust method for determining

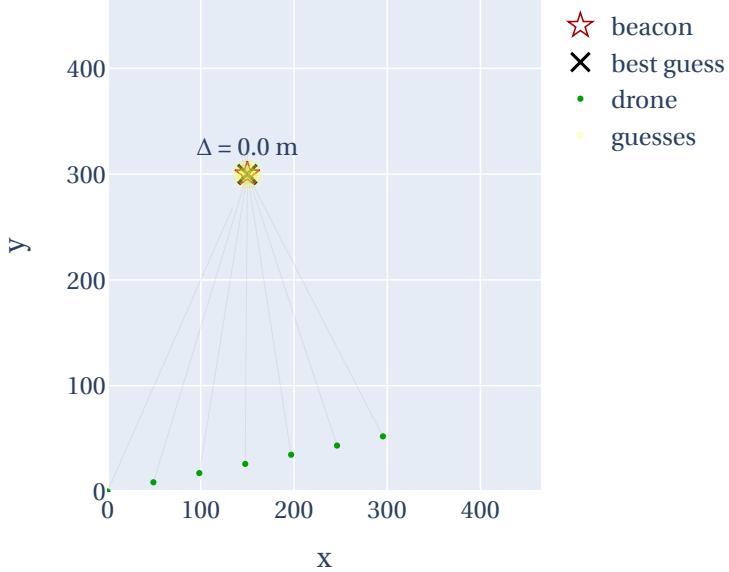


Figure 4 Illustration of source finding using phases computed without noise. In this case, the drone flew from the origin at an altitude of 50 m, at yaw angle 10° , at $v = 5\text{m/s}$, and sampled at 10-s intervals.

```
from drone import Drone

center = 6.7 # cm
square = 5.3 # cm
antennas = np.array([
    [-square-0.5*center, 0, 0],
    [-0.5*center, -square, 0],
    [0.5*center, -square, 0],
    [square + 0.5*center, 0, 0]])

d = Drone(antennas, yrp=(10,0,0), position=(0,0,50),
          beacon=(150,300,0), sample_period=10, phase_noise=0)
d.fly(60)
d.plot(estimates=True, rays=True, height=500, width=500)
d.figure.write_image('nonoise.pdf')
```

Listing 1 Code used to produce Fig. 4.

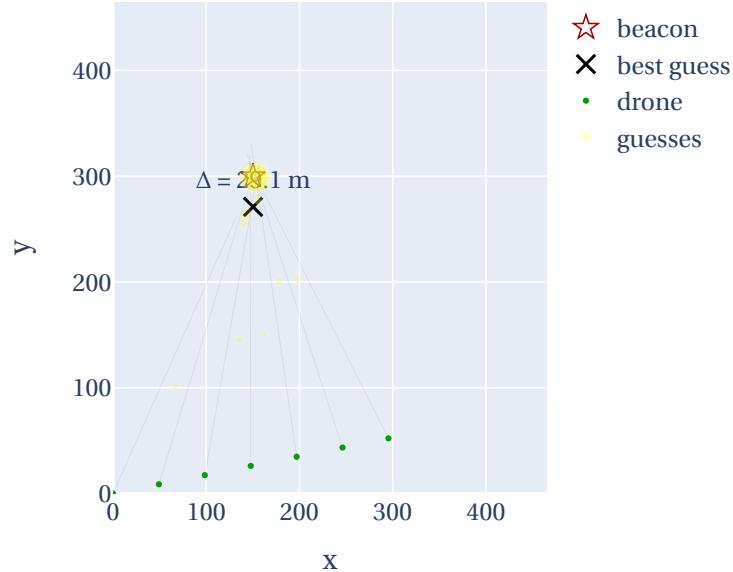


Figure 5 Illustration of source finding using phases computed with a phase noise of 1° on each measurement. The other parameters are identical with Fig. 4. Note: the size of each yellow circle vaguely represents the confidence in the position determined by the corresponding pair of observations. Note that even with a tiny amount of phase noise, some of the intersections are way off target. Yellow dots closer to the drone than the target arise because of significantly different z components of the $\hat{\mathbf{u}}$ vector.

the quality of the individual measurements and of the intersection positions, as illustrated in Fig. 3 will be necessary.

A further (major) challenge arises when the drone antenna array receives signals in the same narrow frequency band from multiple sources. In principle, the MUSIC algorithm can distinguish the sources, returning phase sets for sources in order of descending strength. As the drone flies around, the relative strength of sources will change. Furthermore, individual sources will likely be intermittent, making it challenging to ensure that phase sets from different observations correspond to the same source.

3 Problem Statement

Design and document the process of using radio receivers on a fixed-wing drone to map the radio emissions of United States Navy or Marine Corp installations.

The end-user will require a detailed description of the physical setup of the drone as well as the software used in guiding the drone and interpreting the received radio signals. Additionally, the team needs to document the process of using the system, so that sailors and marines can quickly and reliably deploy it in the field.

4 Goals

This language seems a bit stilted

We, **herein the student team,** initially received proposed goals detailed by our advisors:

- Map out the antenna pattern of a radio telescope from 1–800 MHz by flying a hexacopter that transmits a pseudo-random signal in a hemispherical pattern
- Map out the location and directionality of emitters on the ground (cell phones, WiFi, bluetooth, or beacons) by flying an airplane with a 4-antenna receiver; refine the flight path while in flight.
- Produce educational materials for software defined radio

The proposed scope exceeded the resources and work hours of the team over one year of the Clinic Program. Professor David Harris, with 20+ years of experience working with Clinic teams, even noted how it would take two full teams to tackle the work (Design Review 1). In order to better organize our time and ensure our goals are met, we decided to take on the second proposed goal as our focus for the fall semester. The first proposed goal would be addressed in the spring semester, and the third proposed goal would be taken on by our advisors with students to support and offer feedback.

The first goal for the team was to onboard: to better understand the project design space, to acquire the fundamental skills necessary for building systems with software defined radios (SDR), and to learn the legal regulations with unmanned aerial vehicles. Activities included trial running LearnSDR lessons led by our advisor Professor Jason Gallicchio and studying for the FAA Part 107 exam to be licensed as Remote Pilots.

Our fall deliverable was to have a working direction finding system running on a fixed-wing plane. To meet this deliverable, we created a drone subteam and a radio subteam, each with their respective goals:

- The drone subteam will build, fly, and test the Bixler Trainer planes and a larger fixed-wing plane that will carry and protect the direction finding system.
- The radio subteam will build and test a direction finding system that calculates the direction of arrival (DoA) from radio signals.

At the time of writing for this report, we have made significant progress in each of these goals, but we have a few more challenges to resolve. Significant milestones include synced radio receiver chip, DoA algorithm has been implemented in GNURadio, and trainer planes built and tested to fly.

5 Design Requirements

Our design is a fixed-wing drone which maps the location of radio transmitters.

5.1 Functions

- Operate on a fixed-wing drone
- Compute the angle of arrival from live radio emissions to create a map of radio sources
- Communicate with drone flight controller to update flight path

5.2 Objectives and Metrics

Objectives and Metrics	Justification
Be Lightweight	The drone and the electronics needs to be lightweight so that they can be in compliance with the FAA regulations.
Be Accurate	Since we are doing work for the national defence laboratory and our work will be implemented into the field, it is important to be accurate.
Be Durable	This is important due to the fact that our drone and electronics will be used in the field. If they are flimsy then implementing them in the field will be difficult.

5.3 Constraints

Constraints	Justification
Drone and payload weighs less than 55 lbs	Comply with FAA small UAV regulation
Radio transmitter frequency from 30 MHz to 2.4 GHz	Tactical radio and satcom ranges (liaison meeting Oct. 19)
Radio transmitter transmission power from 250 mW to 10 W	Handheld military radio capabilities (liaison meeting Oct. 19)

6 Design Testing Parameters

When testing the design, we decided to use a transmission frequency of 915 MHz, as this frequency is for general-purpose industrial/scientific use, and so does not require a specialized license. Using this frequency also minimizes the chance that our test-setup transmitters will interfere with third-party radio devices

7 Design

Our prototype relies on a time-delay between when different antennas receive radio transmissions to determine their origin. We use four antennas to minimize

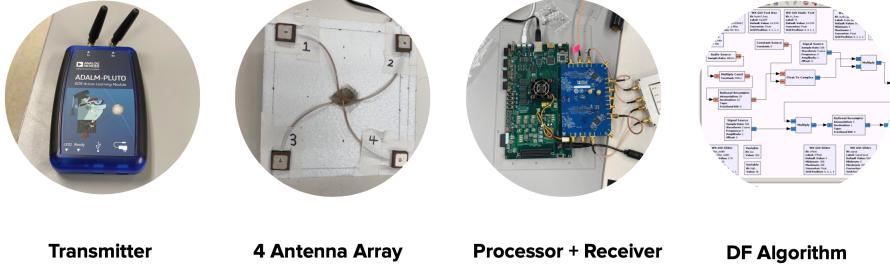


Figure 6 Hardware used in the tests, from left to right. We transmitted signals using the ADALM PLUTO SDR. The antenna array was four 915 MHz patch antennas mounted to a styrofoam board, making a square with edge lengths of 12.9 cm (41% of the wavelength of 915 MHz). The processor was a Xilinx ZC706, the green board in the third image, and the radio receiver was an AnalogDevices fmcomms5, the blue board in the third image. The direction finding algorithm was implemented in GNURadio. A portion of the GNURadio flow graph is shown in the fourth image.

possible ambiguity in the reported direction. To do direction finding, we use the MUSIC algorithm, implemented in GNURadio.

We did not find any off-the-shelf devices under \$10,000 for radio direction finding that did not require manual calibration. So, we decided to move forward with a design which uses the Xilinx ZC706 FPGA along with the AnalogDevices fmcomms5. The fmcomms5 board has two separate radio chips (ad9361), each with two rx/tx ports. This hardware is shown in Figure 6

Because the MUSIC algorithm relies on time-delay between when different antennas receive a radio signal, it is important that all four radio receivers have the same phase relationship (or, at least, a known relationship that can be accounted for).

We can assume that the phase between the two rx/tx ports on the same ad9361 chip is constant. However, every time the frequency, bandwidth, or sample rate is changed, the two ad9361 chips come up with an arbitrary phase relationship (the internal local oscillator on channels 1 and 2 on the first chip have some arbitrary phase difference from the internal local oscillator on channels 3 and 4 on the second chip). Thankfully, the fmcomms5 has built-in circuitry that can determine and correct for this arbitrary phase offset. AnalogDevices provides C code for doing this phase-synchronization, which we implemented in GNURadio.

8 Tests

Key Takeaways:

- Synchronized receivers and calibrated antennas are critical hardware prerequisites for direction finding. One of our sub-goals this semester is to provide accurate phase difference data, which feed into direction finding algorithms.
- Anechoic chamber tests demonstrate synchronization of fmcomms5 chips and n=4 rx channels for input radio waves
- Outdoor tests with patch antenna array compare phase difference between pairs of receivers. Despite radio reflections from complex outdoor environments, after accounting for constant delays from antenna to chip, the relative change in phase differences between antennas tracks the change in antenna array orientation rotating about its center.
- Recommendations: test small systems first to build up known and reliable parts. With coplanar rx and tx, use 2D geometry to calculate the predicted phase difference between antennas to benchmark experimental results.

Good testing helps us evaluate the performance of our direction finding system. Early tests in the semester tested the entire receiver prototype. We ran gr-doa blocks developed by Henry Pick on GNURadio with the four antenna array wired to the fmcomms5 board and Dell laptop. An SDR source was placed near the system, transmitting a radio signal from a known position. GNURadio would then report an angle of arrival relative to a fixed bearing on the antenna array. Then we varied the SDR's spatial position and compared the actual SDR-antenna array bearing to the reported angle of arrival on GNURadio. They were not consistent in all initial tests.

This approach to testing proved to be overambitious. Some of the challenges we faced include:

- We could not deduce which piece of hardware or software was responsible for the inconsistent results. There were too many fallible pieces in a full prototype to know which ones were certainly reliable.
- We could not eliminate significant noise from reflections. Radio waves from the SDR reflect off of flat surfaces: the walls of the lab, the laptop near

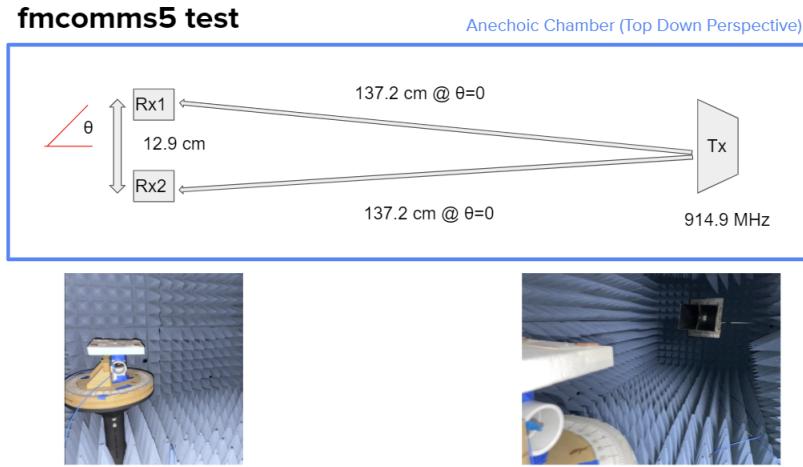


Figure 7 The schematic of the radio receiver four channel phase input (ESC).

to the antenna array, the clothes of the student tinkering the system, and even the ground of an open football field.

- We did not know how to interpret the data from Henry's 2D graphs.

We adapted our testing strategies in light of these challenges. Unit testing is a better approach to narrow down the bugs.

To reduce noise from radio reflections, we first opted to eliminate radio waves entirely by using wired radio signals. Then, using a signal generator and signal splitter, we hoped to send a sinusoidal signal simultaneously to the four receive channels on the fmcomms5 board. However, the signal splitter did not deliver simultaneously. Nevertheless, our tests with the signal generator were still fruitful. Thanks to these wired signal tests, we developed the C script to synchronize the fmcomms5 chips in GNURadio. They also helped us detect a high frequency cosine signal in the receiver data to filter out.

First, we tested each individual patch antenna to ensure that they were resonant at 915 MHz. Now that we were confident with the antenna array's reliability, we could use it to test the fmcomms5 board with radio waves. We still had the issue with radio reflections in field tests. So we sought and received access to the anechoic chamber in the HMC RF Lab. The main tests we performed was here.

We did tests in an anechoic chamber, with the results shown in Figure 8. As we can see, we achieved the desired data, achieving the desired sinusoidal shape.

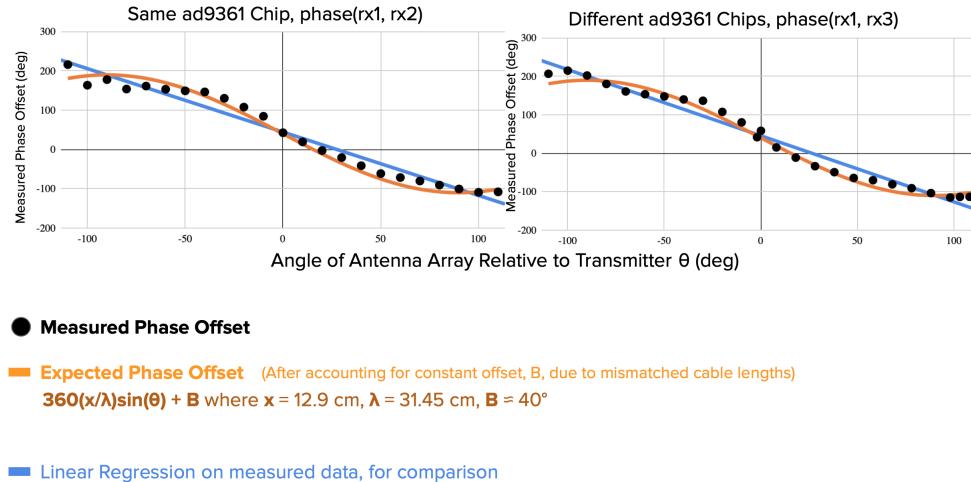


Figure 8 Results from tests in the Anechoic Chamber

After doing tests in the anechoic chamber, we completed additional field tests, ensuring the setup was far away from any potential flat reflection surfaces, and that we used a directional transmitter.

In this test, placed a directional transmitter in the field and pointed it at the antenna array setup. The setup is shown in the Appendix in Figures 11, 12, and 13. Figure 9 shows the results of this test. Part of the reason this test is noisy can be attributed to the fact that it is run at a very low power (0.5 mW from the Pluto, versus the 5W minimum transmission power expected). In addition, unmatched cable lengths were used. These caused a constant offset in phase offsets, which were accounted for in the raw data (antenna 1 vs antenna 3, for example), but not in the direction finding results.

9 Fixed-Wing Drone

Another major component of the problem we are tackling is building and efficiently flying a fixed-wing drone which can support the radio equipment. In this section we will discuss the goals and progress of the fixed-wing drone team.

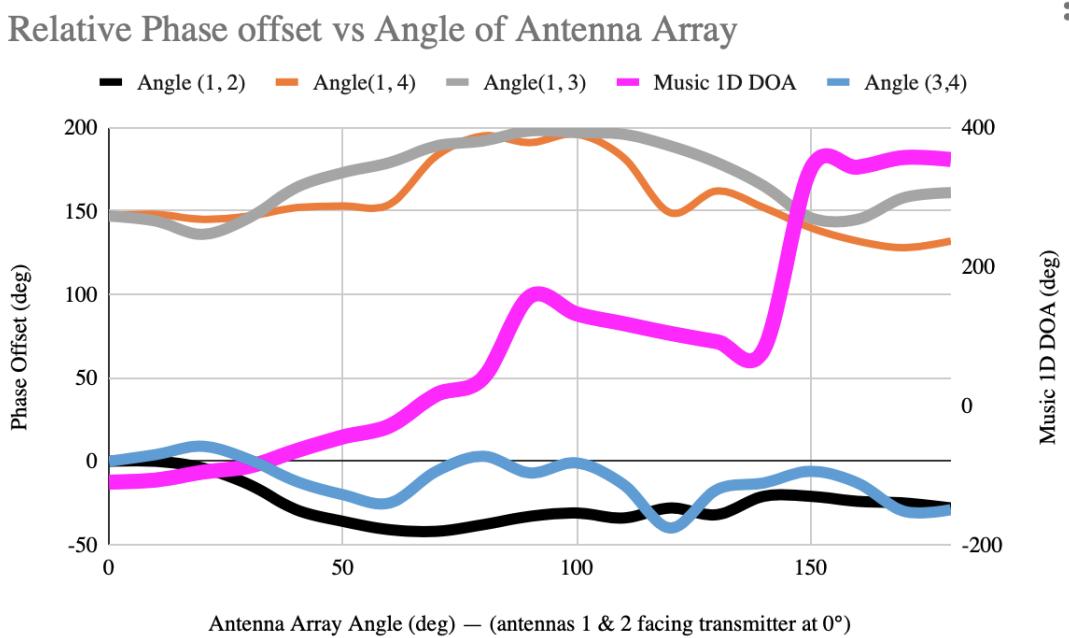


Figure 9 Results of outdoor tests, after normalizing the results to account for unmatched cable lengths. The pink line shows the direction of arrival computed by the MUSIC algorithm, with the y-axis on the right side of the graph.

9.1 Goals

As mentioned in our goals section, the fall deliverable for the fixed wing drone subteam was to

1. Get a Bixler 3 fixed wing trainer plane assembled and in a working condition.
2. Be adept at flying the working Bixler 3 fixed wing trainer plane.

A stretch goal was to successfully fly the planes autonomously with GPS waypoints and host some radio equipment on the trainer planes to partially test the direction finding scheme. This goal was set in place, so if we finished our work early enough then we can start to work on our spring semester goals.

9.2 Progress

The drone team spent this semester building two trainer airplanes (both Bixler 3) and began work on the full-scale drone, the Believer. The two trainer planes were fully assembled with electronics and software completely set up. The purpose of this structure was to

- (a) become familiar with the build process for a fixed-wing drone, especially the electronics and software setup, and
- (b) to practice flying a fixed-wing drone before proceeding to the more expensive Believer.

The electronics and body of both planes were done in parallel and are completed. One of the Bixler models was completed with the complete suite of electronics. This included not only the GPS, flight controller, and receiver which are the bare minimum for flight, but also a telemetry radio as well as a video camera and the associated transmitter. As drone flying practice began the Bixler with the additional electronics was crashed, and can be rebuilt for continuing flying practice. Unfortunately, the second Bixler plane with the more limited electronics set was crashed in test flying and is now in a irreparable state. Due to these two experiences in which flight training led to major time setbacks a new plan for practicing flight must be implemented going into next semester, so our Team will reach the confidence level required to fly the Believer drone. The current plan moving forward is to acquire out of the box model planes which fly more slowly in order to save building time, and grant the pilots much needed practice time.

Flight Controller Setup Documentation

Over the summer, another Bixler 3 was built and setup. We followed these notes in assembling our planes, but found them lacking in some parts. To remedy this, we rewrote the documentation for preparing the flight controller. It is copied here. This exercise will pay dividends as we proceed with the Believer because we plan on using the same or a very similar flight controller. We now are very familiar with the operation of this flight controller.

Flight Controller

Set up firmware on the MATEKH743 flight controller. Use ArduPilot to do so. This is complicated because the flight controller comes preloaded with Betaflight firmware, so it does play very nice with ArduPilot. Follow these steps to get it working. Take the back plate off the flight controller by undoing the screws so that you have access to the servo pins. Next, solder the header pins onto the servo pins and reattach the back plate. Break the header strips and solder them only where the servos will be plugged in. The electronic speed controller (ESC) that comes with the trainer has a battery elimination circuit (BEC), which is regulator that turns 16V battery into 5V to power a non-flight-controller receiver where the servos hook up directly. Our flight controller has a 5V regular that we want to use instead, so for the connection to the first ESC, only S1 and G should have pins. Remove the middle Vx pin with pliers before you solder the other 2. Same for the Vx pin between S2 and G in case we use this flight controller with a two-propeller plane. The plane as an ESC in it. We need to connect it by power and signal to the flight controller. Solder 14AWG to the active and ground of the battery port and a female XT60 connector to the other end of those wires. This will connect to the battery. Solder a wire to one of the active and ground sites on the ESC and the other end of these wires to a male XT60 connector. This will connect to the ESC on the plane. Finally, solder the active, ground, and signal wire to the servo site on the flight controller. Below is a schematic of the setup.

Flying

One of the purposes of building trainer planes first and goals of the semester was to practice flying the drone. Because of higher than anticipated difficulty of getting the plane in working order, we were not able to fly nearly as much as desired. On our first and so far only attempt at flying the Bixler 3, the drone nosedived and crashed shortly after takeoff. The impact of the crash led to damage in the nose of the plane and the loss of a piece required to hold the propeller

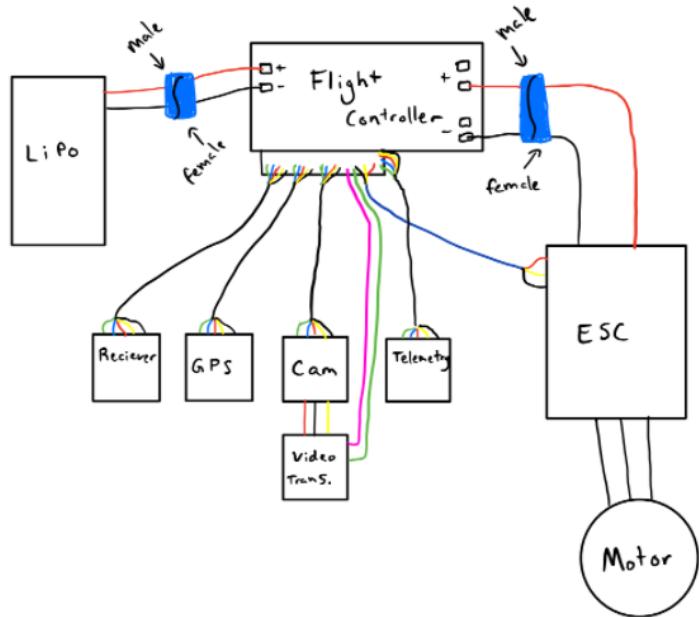


Figure 10 The schematic of the connections for the plane's flight controller.

in place. The damage is repairable and the plane will fly again. On our second attempt with flying we made significant progress learning how to throw the planes and getting them airborne. We had multiple successful flight runs, where we took off made it around the park and preformed somewhat controlled crashed landings, which resulted in very little damage. That being said one of these runs ended with a much more significant crash which resulted in the complete destruction of one of the models. From these experiences significant lessons were learned. We learned proper technique for launching the planes. We learned about their speed and turning capabilities, the latter of which was surprisingly slow. Due to the speed and turning considerations we know that going forward a much bigger practice area is required. An area over the size of a football field is a prerequisite for flying a Bixler sized mode. We also learned about the fragility of large Styrofoam models, and will carefully keep that in mind as we continue working on the payload carrying Believer.

10 Work for Spring Semester

10.1 Direction Finding

- (a) Complete one more outdoor test at a higher transmission power (5W rather than the Pluto's 5 mW)
- (b) Set up the DOA algorithm to run directly on the FPGA board (essentially just means running GNURadio on the FPGA rather than a laptop connected to it)

10.2 Fixed Wing Drone

The primary tasks for the fixed-wing drone team in the spring will be

- (a) finish building and fly the Believer,
- (b) integrate radio equipment onto Believer body, and
- (c) develop algorithm for navigation of the plane, using GPS and direction finding data.

We are well positioned to complete the first two tasks. Although they are a considerable amount of work, we have experience building a fixed-wing drone from kit to the air and have ample space to place radio equipment on the large Believer. However, the autonomous portion will be more challenging. We are charged with designing an algorithm to direct the plane based on the location of radio sources detected to most efficiently and accurately map out all radio sources. This is the terminal goal of the drone team and arguably the entire project as it involves a fusion of data between the direction finding hardware and drone electronics. As such, we have as yet given little thought to the design of the flight algorithm. Some possible designs are

1. Upon detecting a radio source, fly directly over it once or several times.
2. Fly in concentric circles around a radio source.
3. Fly inward in concentric circles around the entire area.
4. Fly in a grid over the entire area.

The latter two possibilities would not require data fusion between the direction finding and plane and thus would be easier. However, they will certainly be less time efficient than methods which leverage the approximate location of radio sources to inform their path, such as the former two in the list above. The appropriate approach must be ascertained through testing of the direction finding scheme on the Believer, which will not happen until the Spring.

11 Second Problem Statement

There is a second problem in this clinic which is closely related to the first, but distinct in some key ways. The problem statement is below.

The second problem is to use a drone with radio transmitters to calibrate radio telescope antennas. Radio telescope antennas are powerful only in a narrow direction, and calibrating their receiving strengths in all frequencies is challenging.

The goal of this project is to use a radio transmitter on board a drone to perform this calibration for all frequencies over the entire hemisphere of sky. The spectral range of the radio telescopes is quite large (~1 MHz – 900 MHz), so to calibrate the radio telescope over its entire spectral range will require a creative signal sent from the drone's transmitter.

11.1 Goals for Fall Semester

The main deliverable for the radio telescope project this fall will be a hypothesis radio signal that can be sent from the drone transmitters to calibrate antenna radio telescopes.

11.2 Progress on Deliverables

We did not do any work on this problem the Fall semester, nor do we have serious plans to do so in the Spring semester. The team decided early on to focus on the direction finding problem exclusively.

12 Appendix



Figure 11 Setup of outdoor tests. Four antennas are mounted on the Styrofoam pad on top of the PVC pipe, and a directional antenna is placed in the field. Four unmatched-length cables enter the blue fmcomms5 board, near the bottom left of the image. The setup is plugged into the laptop at the drone right of the screen, which runs the GNURadio algorithms. For the final prototype, GNURadio will be run directly on the Xilinx board onboard the drone.



Figure 12 Image of outdoor test setup from another angle, with the 4-antenna array at the top of the PVC pipe in the background, and a directional antenna transmitter in the foreground. The person standing behind the antenna array is for scale.



Figure 13 Outdoor setup from another angle. As we can see, the antenna is positioned to point very nearly directly at the antenna array setup, and any reflections should be off of the tree, which should cause random reflections