NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

REAL-TIME DETECTION OF SMALL UNMANNED AERIAL VEHICLES USING A BIO-INSPIRED MEMS SENSOR

by

Rafael Alberto Cruz-Gomez

June 2023

Thesis Advisor: Fabio Durante Pereira Alves
Co-Advisor: Monique P. Fargues

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2023	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
REAL-TIME DETECTION OF SMALL UNMANNED AERIAL VEHICLES USING A BIO-INSPIRED MEMS SENSOR

5. FUNDING NUMBERS

6. AUTHOR(S) Rafael Alberto Cruz-Gomez

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Ormia Ochracea-inspired microelectromechanical systems (MEMS) sensors can be arranged in a configuration that detects the direction of arrival (DoA) of incident sound. Previous research results indicate that unambiguous DoA can be determined over 360 degrees in azimuth. To this date, analog readouts have been conducted using laboratory instrumentation. The objective of this study was to develop, build and test a circuit configuration that includes housing and power for the MEMS sensor, and the design of a graphical user interface (GUI) to read the DoA from an array of sensors and triangulate the position of a multi-rotor small UAV using GPS position data. Test fields were performed using a configuration of two nodes to detect a small rotor UAV. The operation scenario was displayed on a map. This new configuration can detect sound from any detectable source and provide the coordinates of the source of the sound.

14. SUBJECT TERMS UAV, direction of arrival, bio-inspired MEMS	**15. NUMBER OF PAGES** 73
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

REAL-TIME DETECTION OF SMALL UNMANNED AERIAL VEHICLES USING A BIO-INSPIRED MEMS SENSOR

Rafael Alberto Cruz-Gomez
Lieutenant, Mexican Navy
BSEE, Escuela de Ingenieros de la Armada de México, 2012

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN ENGINEERING ACOUSTICS

and

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2023

Approved by: Fabio Durante Pereira Alves
Advisor

Monique P. Fargues
Co-Advisor

Oleg A. Godin
Chair, Department of Physics

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Ormia Ochracea-inspired microelectromechanical systems (MEMS) sensors can be arranged in a configuration that detects the direction of arrival (DoA) of incident sound. Previous research results indicate that unambiguous DoA can be determined over 360 degrees in azimuth. To this date, analog readouts have been conducted using laboratory instrumentation. The objective of this study was to develop, build and test a circuit configuration that includes housing and power for the MEMS sensor, and the design of a graphical user interface (GUI) to read the DoA from an array of sensors and triangulate the position of a multi-rotor small UAV using GPS position data. Test fields were performed using a configuration of two nodes to detect a small rotor UAV. The operation scenario was displayed on a map. This new configuration can detect sound from any detectable source and provide the coordinates of the source of the sound.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction **1**

1.1 Background 1

1.2 Objectives 2

1.3 Thesis Organization 3

2 Literature Review **5**

2.1 Review of Unmanned Aerial Vehicles (UAV) 5

2.2 Fundamentals of Acoustics 7

2.3 Acoustic Signature of a UAV 9

2.4 Description of the MEMS Sensor 11

3 Real-time Detection of a UAV **13**

3.1 DoA Determination Algorithm 13

3.2 NPS Current DoA Determination Architecture. 15

3.3 Proposed Architecture 16

3.4 Hardware Description 18

3.5 Low-level Firmware Description 23

3.6 Communications Protocol 28

3.7 Description of the Graphical User Interface 29

4 Results **37**

4.1 Test Setup 37

4.2 Localization 38

4.3 GPS Data Log 40

4.4 Field Test Results 40

5 Conclusions **45**

5.1 Summary 45

5.2 Future Work 46

List of References **49**

Initial Distribution List **53**

List of Figures

Figure 2.1 Architecture of an Unmanned Aerial Vehicle (UAV) 6

Figure 2.2 Small UAV Models 6

Figure 2.3 Intertial and Body Frame of a Quadcopter 10

Figure 2.4 Analysis of the Acoustic Signature the Quadcopter Typhoon H . . 11

Figure 2.5 Design of the MEMS Sensor. 12

Figure 3.1 Cross-dipole Configuration of the MEMS Sensor 13

Figure 3.2 Current Architecture Designed in the Sensor Research Lab (SRL). 15

Figure 3.3 Architecture of the Direction of Arrival (DoA). Determination and Localization. 17

Figure 3.4 Self-contained Single Node DoA Detector. 19

Figure 3.5 Internal Components of the Sensor Node. 19

Figure 3.6 Schematic Circuit of the Sensor Node. 20

Figure 3.7 Low-level Architecture of the Sensor Microcontroller. 23

Figure 3.8 DoA Acquisition Flow. 25

Figure 3.9 Low-level Architecture of the Communications Microcontroller. . 27

Figure 3.10 Graphical User Interface (GUI). 30

Figure 3.11 GUI Data Flow. 31

Figure 3.12 GUI Communications Options. 32

Figure 3.13 Software Node Widget. 34

Figure 3.14 Software Initial Map Representation. 35

Figure 4.1 Graphical Representation of the Triangulation Estimator. 38

Figure 4.2 GPS Data Logger Used to Store the Position of the Drone. . . . 40
Figure 4.3 Paso Robles Test Setup. 41
Figure 4.4 Paso Robles Test Setup DoA. 42
Figure 4.5 Paso Robles Test Setup: Estimation of the Drone Position 43

List of Tables

Table 2.1 Classification of UAV 7

Table 3.1 Calculation of the Current Consumption for the Sensor Node. . . . 22

Table 3.2 Definition of the MavLink Messages. 28

Table 4.1 Configuration Parameters for the Field Tests. 37

Table 4.2 Position Data for the Sensor Nodes. 42

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

DoA	direction of arrival
FFT	fast Fourier transform
GUI	graphical user interface
GCS	ground control station
ISR	intelligence, surveillance, and reconnaissance
MEMS	microelectromechanical systems
SRL	sensor research lab
SONAR	sound detection and ranging
UAV	unmanned aerial vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

This thesis is part of ongoing research conducted at the Physics Department of the NPS to develop a microelectromechanical systems (MEMS) sensor to detect drones and gunshots.

This sensor presents a cosine-angular dependence of the angle of the arrival of the sound. The direction of arrival (DoA) is determined by a cross-dipole configuration of the sensor with an omnidirectional microphone to determine the quadrant of the incoming sound.

Current architecture uses one set of cross-dipole configuration to determine the DoA of the sound. Two software solutions are implemented to display the results of the detection algorithm, Matlab and LabView programs running on a Windows laptop using RS-232 protocol.

Two main problems are addressed in this thesis. The first is to expand the capabilities of the current system to receive more than one configuration using a defined communications protocol and creating a sensor network using Wi-Fi communications. The second addresses the implementation of an auto-positioning system using GPS and magnetometer sensors to triangulate the absolute position of the incoming sound. This position is displayed in a map and can be broadcast to more end-user applications in the same local area network (LAN) or stored in a telemetry file for further analysis.

Field tests were performed in Paso Robles and Oceanside, CA.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

Primeramente quisiera agradecer al Creador por su guía durante toda mi vida. Quiero agradecer a mi familia por todo su apoyo y comprensión para que yo pueda continuar con mis estudios.

I would like to thank the Mexican Navy for the opportunity to attend NPS, especially CDR (Ret) Mariano Lizarraga, whose teaching and advice directed my interest toward the realm of unmanned aerial vehicles (UAV). I am grateful to CDR Julian Ramos, Alejandro Molina and LT Ivan Barbosa, of the Autonomous Systems Lab (ASL), for their friendship.

My sincerest gratitude to my advisor Dr. Fabio Alves and Dr. Gamani Karunasiri for giving me the opportunity to work in the Sensor Research Lab (SRL) in this interesting project. Thank you to pushing me to work harder.

Special thanks for my co-advisor Dr. Monique Fargues, whose teaching helped me to understand and apply the fundamentals of signal processing.

My gratitude to Dr. Peter Crooked, whose endless hours of debugging and testing the algorithms used in the MEMS sensor were crucial to achieve a reliable operation of the sensor.

Finally, I want to thank Dr. Bruce Denardo and Dr. Oleg Godin for introducing me to the world of acoustics. Thank you for your teachings.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Background

Unmanned aerial Vehicles (UAV) have become an abundant technology due to the wide variety of size, capabilities and applications [1]. They have applications in agriculture, imagery, exploration, and ISR operations. Due to the wide variety and abundance of drones,[1] there is a need to regulate where drones can be flown and which areas are restricted. In order to prevent drones from flying in restricted areas, the first task is to detect them.

Modern detection systems can include radars, electro-optic cameras, and acoustic sensors using signal processing and machine learning techniques to detect and track drones [2]. The major challenge to detect an UAV is that they have a radar cross-section so small that it might be undetected by radars, although many efforts in that area have been conducted with satisfactory results [3], [4]. Other problems related to drone detection using electro-optic cameras are that the drones can be camouflaged to be indistinguishable to these sensors [5].

Sound detection and ranging (SONAR) is a technique that uses mechanical waves for communications, detection, navigation and measuring distances [6], a technique typically used in underwater applications and for submarine navigation where attenuation is a limitation factor to use electromagnetic waves [7]. Most available microphones are omnidirectional, meaning that they only sense the power (in decibels) of the incoming sound [8]. It is common to use sensor arrays in robotics applications, with the array installed in the front, and sides of the robot to avoid obstacles or walls to allow motion or in an angular setup that spans the entire 360 degrees of the robot [9]. It is worth mentioning that the obstacle avoidance algorithm in drone navigation can be implemented using acoustic sensors [10]. Sonar detection is not applicable for detecting drones in air. A possible way is to use passive acoustic detection [11].

In acoustics, the most popular solution for drone detection is using an array of acoustic

[1]For the purposes of this thesis, the terms drone and UAV are used interchangeably.

sensors [12]. In this configuration, the angle of arrival of the sound can be determined based on the time of arrival or phase differences on each sensor [13]. The position and number of elements in the array provide the means to choose the best signal processing technique to create the adaptive beam and therefore obtain the angle of arrival of the incoming sound. There are powerful techniques that allow detection of multiple sound sources and also track motion of these sources [12].

The downside of using sensor arrays, besides the size, is the need of powerful processor capabilities, which increases as the number of elements of the array does. To reduce the burden of the processing unit and also the complexity of the overall system, research groups have been investigating other technologies. One such example is the use of piezoelectric sensors and MEMS [14], where the angle of arrival can be estimated from a voltage provided by a circuit conditioner attached to this sensor.

Current research at the NPS, SRL focuses on the development of a MEMS sensor based on the hearing system of the parasitic fly Ormia Ochracea [15]. This sensor exhibits dipole-like sound angle dependence, and it is meant to be operated at resonance, to improve the signal-to-noise ratio. This sensor can be arranged in a cross-dipole configuration in a self-contained housing that provides power and signal conditioning allowing for unambiguous direction of arrival detection of small UAVs.

1.2 Objectives

The objective of this thesis is to design, build, and test a system that expands the existing architecture developed in the SRL to determine the DoA of a UAV using a configuration of at least two nodes of acoustic sensors, each one containing two bio-inspired MEMS sensors arranged in a cross-dipole configuration with an omnidirectional microphone. This system requires the following features:

- Acoustic data collection and processing.
- GPS and magnetometer sensor readings to determine the absolute position of the sensor in north-east-down (NED) coordinates.
- Microcontroller unit with Wi-Fi communications capabilities to broadcast the collected data from the sensors.

- A communications protocol for the overall system.
- Algorithm that triangulates and provides georeferenced position of a drone based on the collected data of at least two sensors.
- GUI that collects data from every sensor and displays the detected angle of arrival of every sensor node and triangulation of the drone in a map window.
- Data log system to store all the collected data for further post-processing.

1.3 Thesis Organization

This thesis is organized as follows: Chapter 2 presents a summary of the existing hardware architecture in the SRL, and the algorithms implemented to determine the angle of arrival of UAV. Chapter 3 presents the contribution of this development explaining the hardware components, the power considerations, firmware design, communications protocols implemented and description of the GUI developed to collect data, compute triangulation algorithms, maps server accessed, and telemetry recording. Chapter 4 summarizes the data collected in the field tests. Finally, Chapter 5 presents conclusions and future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Literature Review

2.1 Review of Unmanned Aerial Vehicles (UAV)

UAVs are complex systems that comprise different components to work either with a remote pilot that directly commands the aircraft manually or autonomously by using an autopilot with a flight navigation path. The most common UAVs have the following elements [16]:

- The aircraft: Aerial vehicle capable of flight with onboard power supply (batteries), autopilot, propulsion, control surfaces, positioning sensors, communications protocol.
- GCS: system with capabilities to receive information from the aerial vehicle, send orders, store the received information (usually referred as telemetry), play the video or information collected from the payload. This system can be as simple as a radio controller with a smartphone in the case of simple UAV or a complete data processing center for the most complex systems. The GCS is also the generic term to refer to the software that controls the aircraft. This software is capable of send waypoints, paths, autoland routines, lost communications procedures and with a user-friendly GUI.
- Payload: The reason to have a vehicle in the sky is to surveil. In this sense, a payload generally is a gimbal camera with capabilities to take pictures and record video that can be sent to the GCS in real-time or store in a memory device to be retrieved and post-processed later. Other common payloads are LIDAR, infrared cameras, weapons, etc.
- Link: In order to communicate with the aerial vehicle, the link is a set of antennas that can send and receive telemetry using radiofrequency. For civilian purposes, the link frequency for UAV is 900 MHz. When the payload is a gimbal camera there is also a video link to stream this video. The common frequency for the video link ranges from 2 GHz to 5 GHz.
- Support equipment: non-essential systems to flight such as battery chargers, storage boxes, maintenance kits, post-processing software.

Figure 2.1 shows the components of a UAV. We can see that a radiofrequency link is needed

with a common communications protocol for both elements to communicate the aerial vehicle with the GCS.

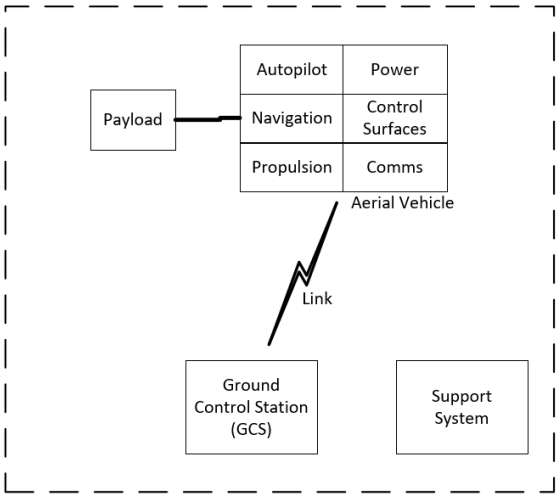

Figure 2.1. Architecture of an Unmanned Aerial Vehicle (UAV). Adapted from [16].

Figure 2.2 shows different drones. These drones were used to collect data and test the algorithm implemented in this thesis.

Figure 2.2. Small UAV models. Top left: Matrice 600 Pro; top right: Phantom 3 Standard; bottom left: Skydio x2; bottom right: Typhoon H. Source: [17]–[20].

Drones can be sorted in different categories depending on size, mission, equipment, endurance, vehicle design approach, etc. [21]. Table 2.1 summarizes the most common classification criteria. In this context, a small UAV is a drone with a weight under 25 kg and with short-range capabilities.

Table 2.1. Classification of UAV. Adapted from [21].

Criterion	Classes	Description
Size	Micro	Weight up to 250 g
	Mini	Weight from 250 g to 2 kg
	Small	Weight from 2 to 25 kg
	Medium	Weight from 25 to 150 kg
	Large	Weight greater than 150 kg
Range	Very Close Range	Range less than 5 km
	Close Range	From 5 to 50 km
	Short Range	From 50 to 150 km
	Medium Range	From 150 to 650 km
	Large Range	Range greater than 650 km
Vehicle design	Fixed wing	Conventional take off and landing
	Helicopter	Lift and trust (L/T) are supplied by a rotor
	Multicopter	L/T supplied by a set of horizontal rotors
	VTOL	Hybrid design fixed wing vehicle and multicopters, for vertical takeoff and landing (VTOL)

2.2 Fundamentals of Acoustics

As presented in section 2.1, there are a wide variety of drones. In order to analyze the acoustic signature of a UAV, we first explain what sound is and where it comes from. The acoustic signature of a drone is unique and depends on its constituent parts.

Sound is another term for vibration waves that are generated as a result of a pressure fluctuation. This fluctuation occurs in a compressible fluid. These waves propagate through a medium that can be solid or fluid. The velocity of propagation depends on the characteristic

impedance (ρ) of the medium [22]. In air, $\rho = 415Pa \cdot s/m$ and the speed of sound (c) is $c = 343m/s$. The wave equation is given as:

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{2.1}$$

where p is the acoustic pressure, c is the speed of sound, and ∇ is the Laplace operator. The solution to the wave equation in cartesian coordinates is given by:

$$p(r, t, k) = Ae^{\pm kr}e^{i\omega t}, \tag{2.2}$$

where $\omega = 2\pi f$ is the angular frequency and $k = \omega/c$ is the wave number. Therefore, the acoustic pressure can be described in terms of the wave number (k), the propagation vector (r) and time (t).

According to Fourier analysis [23], any periodic signal can be represented as an infinite sum of trigonometric functions whose frequency is a multiple of a common frequency called the fundamental. Equations 2.3 and 2.4 show the representation of a signal into its constituent harmonics.

$$x(t) = \sum_{k=-\infty}^{+\infty} a_k e^{jk\omega_0 t} \tag{2.3}$$

$$a_k = \frac{1}{T}\int_T x(t)e^{-jk\omega_0 t}dt, \tag{2.4}$$

where w_0 represents the fundamental frequency and T is the period of the signal. The components of the periodic signal are called harmonics.

When the signal is not periodic, the Fourier series can be expanded by considering the period (T) to be infinite, which leads to the Fourier transform (FT). Equations 2.5 and 2.6 present the FT and the IFT expressions, respectively.

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(j\omega) e^{j\omega t} d\omega \tag{2.5}$$

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt. \tag{2.6}$$

The FT is meant to be used for continuous signals. This is the ideal case when the signal is infinitely sampled. In real-life systems, signals are sampled before processing. Therefore, the signal is no longer continuous but discrete. Computers can evaluate the discrete FT using algorithms known as FFT. The FFT algorithm is important because allows the computer system to determine the amplitude and frequency components of the detected signal and, therefore analyze the acoustic signature of any incoming sound. Equations 2.7 and 2.8 present the discrete Fourier transform and the inverse Fourier transform expressions:

$$x[n] = \frac{1}{2\pi} \int_{2\pi} X(e^{j\omega}) e^{j\omega n} d\omega \tag{2.7}$$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j\omega n}. \tag{2.8}$$

2.3 Acoustic Signature of a UAV

For small UAVs, the main propulsion system is a DC motor. Depending on the vehicle design, the motor can be used in a push-pull configuration for a fixed-wing aerial vehicle, a single rotorcraft for a helicopter or multiples rotorcrafts for multicopters [24]. The most common configurations of multicopters are quadcopter, hexacopter, and octocopter, having four, six, and eight motor-propeller sets respectively.

As mentioned in section 2.2, the sound is generated by a pressure variation in a fluid. This variation is caused by the lift force in the drone. Thus, the main component of the sound emitted by a drone is due to the DC motor and the propeller. Previous research has been conducted on drone propellers, comparing their behavior with the behavior of a conventional

propeller [25].

Figure 2.3 shows the body frame for a quadcopter [26]. The quadcopter includes the torques, angular velocities, and forces generated by the rotors. The position of the quadcopter is determined by the three Euler angles: pitch angle (θ), represents the rotation of the vehicle around the y-axis. The yaw angle (ψ) represents the rotation of the vehicle around the z-axis. The roll angle (ϕ) represents the rotation of the vehicle around the x-axis. These three angles represent the body frame of the quadcopter. In order to generate a change in the regime of the navigation of the quadcopter and, thus the Euler angles, the rotor must change the velocity. This change in velocity of a single rotor affects the sound emitted by the vehicle. Therefore, the sound emitted by a drone is determined by the vehicle design, by the number of motor-propeller, by the speed of each rotor, by the path of the drone, as well as its size and distance from the acoustic sensor.

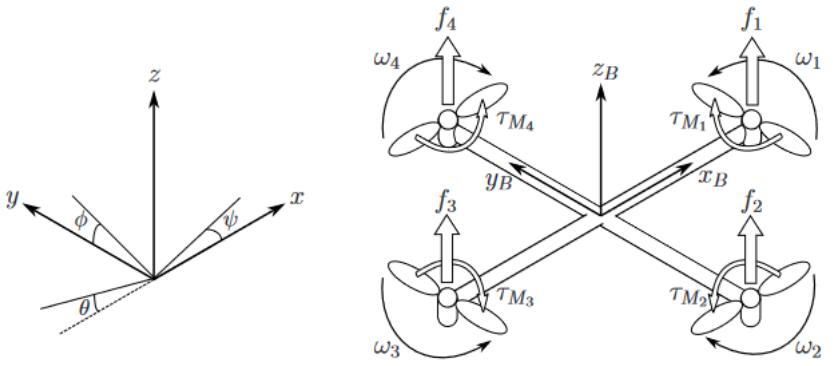

Figure 2.3. Intertial and body frame of a quadcopter. Source: [26].

The sound emitted by the rotors of a drone is acquired by a MEMS sensor, then a routine is executed in a microcontroller to process this information and determine the presence of a drone. A recorded acoustic signature of the drone Typhoon H is shown in figure 2.4. The top plot shows the amplitude in millivolts of five seconds of recording. The bottom plot shows the FFT of this sound. In the plot, the frequency component with the highest amplitude, 1.37 mV corresponds to 537 Hz.

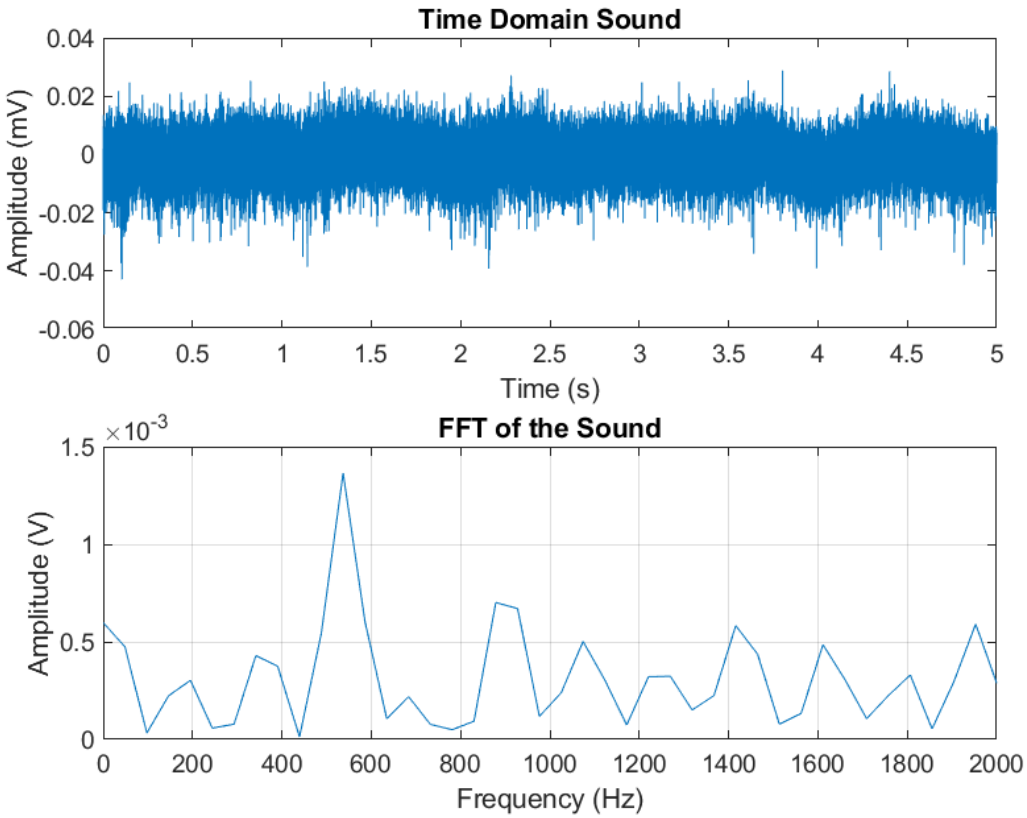

Figure 2.4. Analysis of the Acoustic Signature the Quadcopter Typhoon H. Top Plot: Amplitude of 5 seconds of recording. Bottom Plot: FFT of the Sound, frequency components of the acoustic signature. Data collected in a field test.

Understanding the frequency components of the drones such as those presented in Figure 2.4, is important for the design of the MEMS sensor developed at NPS.

2.4 Description of the MEMS Sensor

The sensor designed in the Physics Department is inspired in the tympana configuration of the Ormia Ochracea [15], [27]. The mechanically connected ears of the Ormia Ochracea fly have been used as the foundation for the development of a narrowband MEMS direction finding sensor. The sensor is made up of two wings joined together in the center and two legs

supporting it on a substrate. Two vibration modes are present in this design, bending, and rocking modes. For this work the sensor will be used at its bending resonance frequency. Figure 2.5 shows the design of the sensor (a) and its frequency response (b). This design provides two close resonances to broaden the sensitivity and is used in an orthogonal configuration, as presented in section 3.2, to determine the DoA of incident sound.

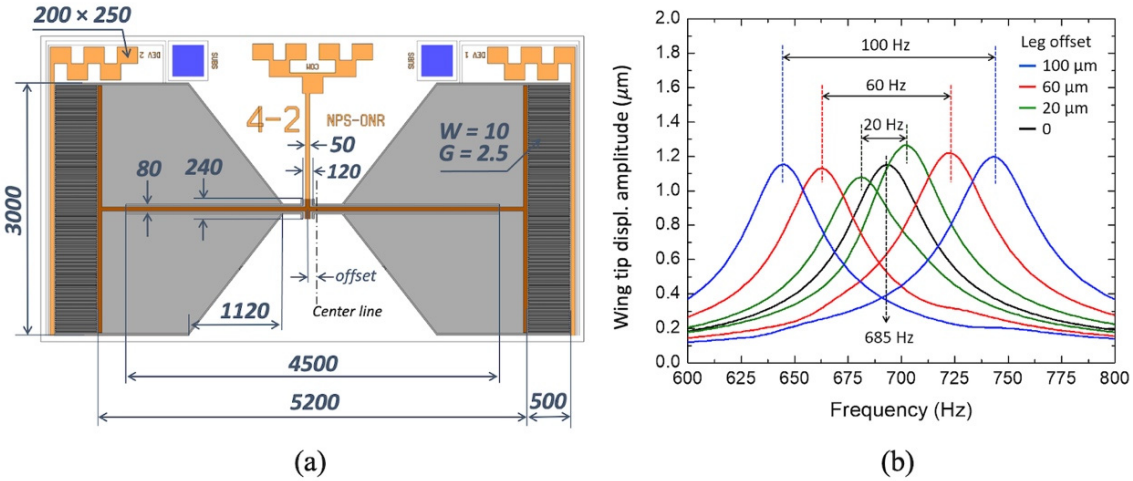

Figure 2.5. Design of the MEMS Sensor in the NPS Physics Department (a) Physical Design of the Sensor (all dimensions are in micrometers). (b) Frequency response of the Sensor. Source: [15].

CHAPTER 3:
Real-time Detection of a UAV

3.1 DoA Determination Algorithm

To determine the angle of arrival of sound, three MEMS sensors are placed in an orthogonal configuration. In this configuration, presented in figure 3.1, the cosine sensor is oriented pointing towards the acoustic front, the sine sensor is placed orthogonal and the omnidirectional sensor is placed in the center of these two sensors.

Figure 3.1. Cross-dipole Configuration of the MEMS Sensor. Cosine Sensor (Top), Omnidirectional Sensor (Center), Sine Sensor (Bottom). Sine and Cosine sensors are orthogonal.

In this configuration, unambiguous direction of arrival of the sound can be determined based on the arctan estimator [28]. Based on this method, each sensor will detect the acoustic signal as presented in equations 3.1, 3.2, and 3.3:

$$Vc(f) = P(f)H_c \cos\beta \qquad \text{cos sensor,} \tag{3.1}$$

$$Vo(f) = P(f)H_o \qquad \text{omni sensor,} \tag{3.2}$$

$$Vs(f) = P(f)H_s \sin\beta \qquad \text{sin sensor,} \tag{3.3}$$

where $Vc(f)$, $Vo(f)$, and $Vs(f)$ are the frequency components of the detected voltage of the cosine, omnidirectional, and sine sensors, respectively; $P(f)$ is the Fourier transform of the acoustic pressure, H_c, H_o, and H_s are the complex transfer functions of the three sensors and β is the angle of arrival with respect to the orientation of the cosine sensor.

Based on this estimator, the angle of arrival β is presented in Equation 3.5:

$$\beta = \arctan\left(\frac{P(f)H_s \sin\beta}{P(f)H_c \cos\beta}\right) \tag{3.4}$$

$$\beta = \arctan\left(\frac{Vs(f)}{Vc(f)}\right). \tag{3.5}$$

The previous equation requires a new factor C, to correct the phases of H_s and H_c since both are 90° out of phase. Equation 3.6 presents the correction factor:

$$C = \frac{H_c}{H_s} \tag{3.6}$$

Multiplying this correction factor by the sine transfer function (H_s) yields the correct answer as shown in 3.7

$$\beta = \arctan\left(\frac{VsC}{Vc}\right) \tag{3.7}$$

Finally, to determine the quadrant, the information carried in the omnidirectional sensor, Vo is added as a factor to the previous equation, obtaining the angle of arrival base on the

phase shifting added by this sensor, resulting in the following expression:

$$\beta = \arctan 2\left(\frac{VoVsC}{VoVc}\right) \tag{3.8}$$

This expression will be used in the calculation of the angle of arrival. The correction factor C is different for every cross-dipole sensors. For this reason, each sensor assembly will have its own correction factor table.

3.2 NPS Current DoA Determination Architecture

Current architecture shown in figure 3.2 consists of an orthogonal configuration of two MEMS sensors and one omnidirectional microphone. Each transducer provides an analog voltage that represents the acoustic signal level read.

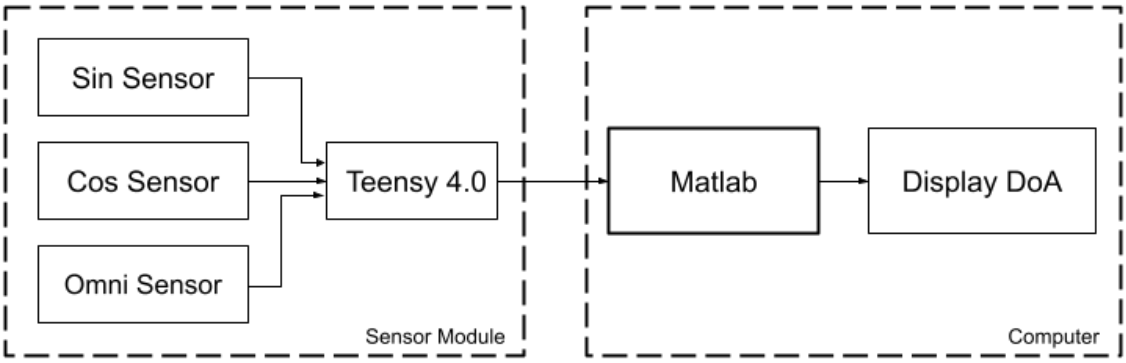

Figure 3.2. Current Architecture Designed in the Sensor Research Lab (SRL): Configuration composed of an assembly of two MEMS sensors, one omnidirectional microphone, connected to a computer to display the DoA.

The current architecture consists of a sensor module and a computer running a Matlab script that reads the calculated DoA. The sensor board consists of the following elements:

- MEMS Sensors: There is a set of two MEMS sensors placed in an orthogonal configuration and an omnidirectional microphone. Each one of the three sensors provide

a maximum output of 2.6 V, and represents the sine, cosine and omnidirectional amplitude and phase of the detected acoustic signal [15].

- Signal conditioning block: The three voltages are offset and scaled to provide a range of 0–3.3 Vdc. This signal is routed to the microcontroller.
- Microcontroller: The microcontroller Teensy 4.0 has an ARM Cortex-M6 CPU that runs at 600 MHz with a floating-point unit (FPU), built using a 32-bits architecture. The microcontroller has the following peripherals: seven UART interfaces TTL-level, three SPI, three I2C ports, 14 analog input pins, 40 digital GPIO, and three CAN bus [29]. The purpose of the microcontroller is to read the three analog voltages coming from the signal condition circuit, based on the DoA algorithm, and provide an angle which represents the direction of arrival of the signal.

In this configuration, the microcontroller reads the analog voltage of the MEMS sensor, process this data, and displays the results in a Matlab figure. An RS-232 communications link is established between the sensor board and the computer. The sensor board sends the angle of arrival of the acoustic sound.

3.3 Proposed Architecture

The current architecture requires a wired connection between the computer and the sensor board using a serial port. This configuration can only communicate with one sensor module at a time, and it is limited to the length of the wire. In order to triangulate the position of the drone, a minimum of two sensor modules is required. Every sensor module should be able to detect and broadcast its position to a computer that can perform the calculations and determine the position of the drone.

The proposed architecture uses the protocols (Transmission Control Protocol/Internet Protocol) TCP/IP through the standard IEEE 802.11g protocol, commonly referred as Wi-Fi. This communications link uses the 2.4 GHz band [30]. The use of the 802.11g standard over RS-232 standard, allows the distance of transmission to increase between the sensors modules and the computer. Another advantage is the reading of multiple sensors at a time. This configuration also allows multiple computers to read the information provided by the nodes and send commands to them.

The clear benefits of using multiple sensor nodes are to increase the range of the MEMS sensor and the accuracy of the detection. Other signal processing techniques can be used to determine a failure in any sensor used in the triangulation calculations.

Two nodes can provide the position of the drone in two dimensions but increasing the number of nodes can provide a 3-D position. The proposed architecture is presented in figure 3.3 for a two-node configuration.

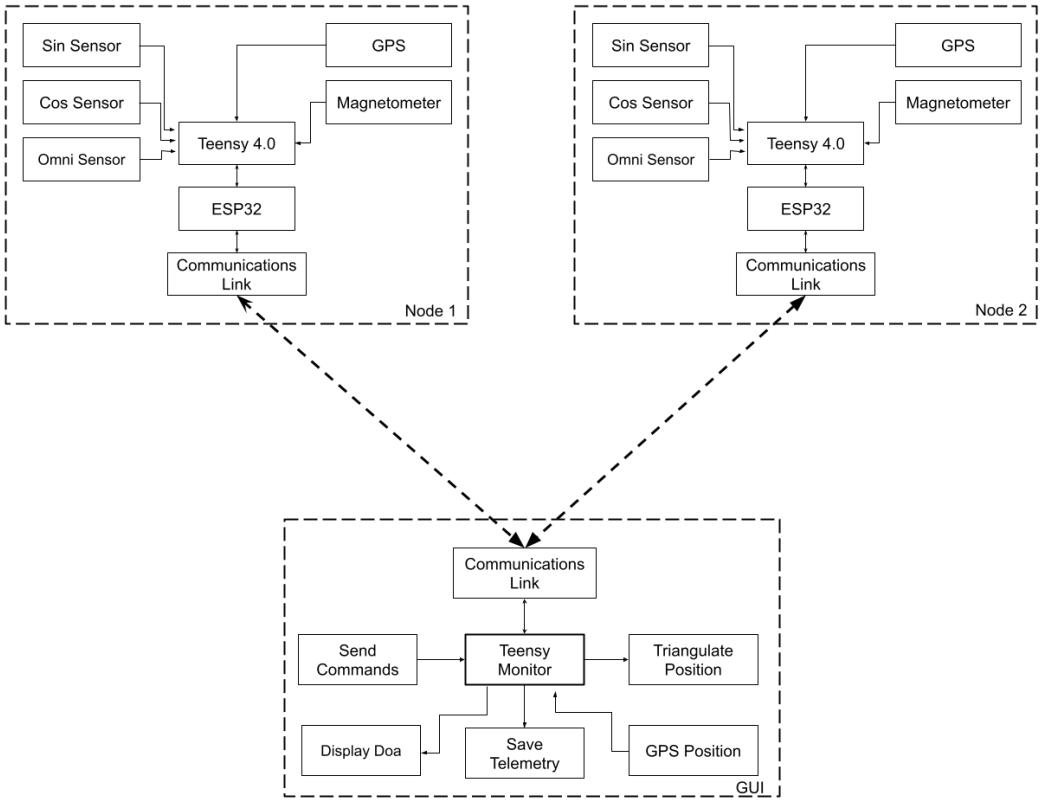

Figure 3.3. Architecture of the Direction of Arrival (DoA). Determination and Localization. Configuration using two-nodes with one sensors assembly each one and a (GUI) running a developed software, using Wi-Fi communications.

For this solution, two modules were added to each node sensor: GPS and magnetometer to

provide position and orientation data. Additionally, an ESP32 microcontroller was added to provide each sensor node with Wi-Fi capabilities.

Using this architecture, every node will comprise a set of two MEMS sensor, position sensors, and communications link. A computer GUI was developed to read the provided data from every node and to send commands.

In the following sections, every element of the architecture will be described. Section 3.4 introduces the different modules and their interconnection. The firmware, described in section 3.5 is the program loaded in the Teensy 4.0 and ESP32 boards that perform the task in every node. The communications protocol is described in section 3.6. This protocol is a common protocol for every element in the network. Finally, the GUI detailed in section 3.7, is the software that reads/sends telemetry from/to the sensor nodes, stores telemetry and does the calculations to display the drone position in a map.

3.4 Hardware Description

New components were assembled to create a Sensor Node. A sensor node consists of an assembly of three MEMS sensors and the necessary electronics to determine and send the DoA wirelessly.

3.4.1 Physical Components

The sensor box has the necessary electronic components to provide power, estimate the angle of arrival, determine the GPS position and send this data to the GUI, detailed in section 3.7.

Figure 3.4 shows an image of the sensor node. This node is compossed of two important elements: the sensors assembly and the sensor box.

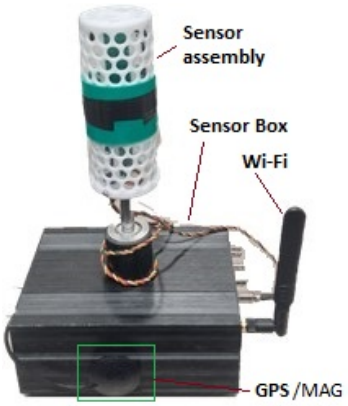

Figure 3.4. Self-contained Single Node DoA Detector.

The internal components of the sensor box are presented in Figure 3.5. The left image shows the internal components of the sensor box. In the right image the configuration of the MEMS sensors is shown. In this configuration, the sine and cosine sensors are set in an orthogonal configuration with the omnidirectional sensor placed in the center.

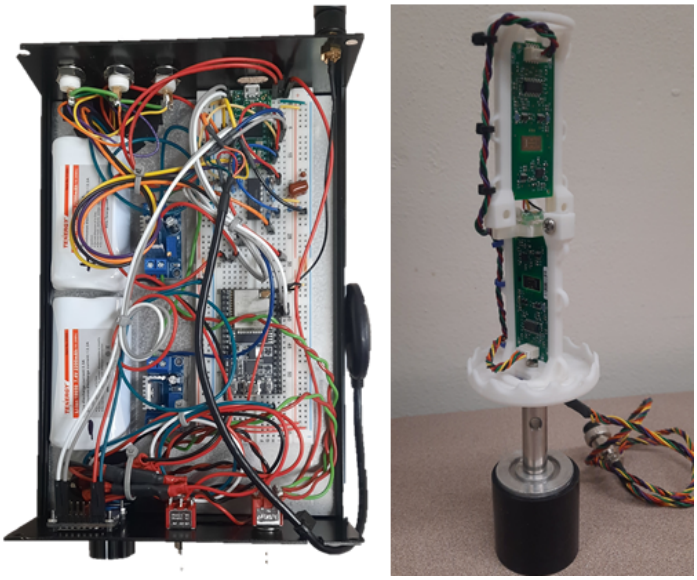

Figure 3.5. Internal Components of the Sensor Node: left image sensor box prototype, right image physical configuration of the MEMS sensors.

Designing a node requires considerations on power consumption and distribution, wiring, housing, shielding, etc. The schematic diagram of the circuit is shown in figure 3.6.

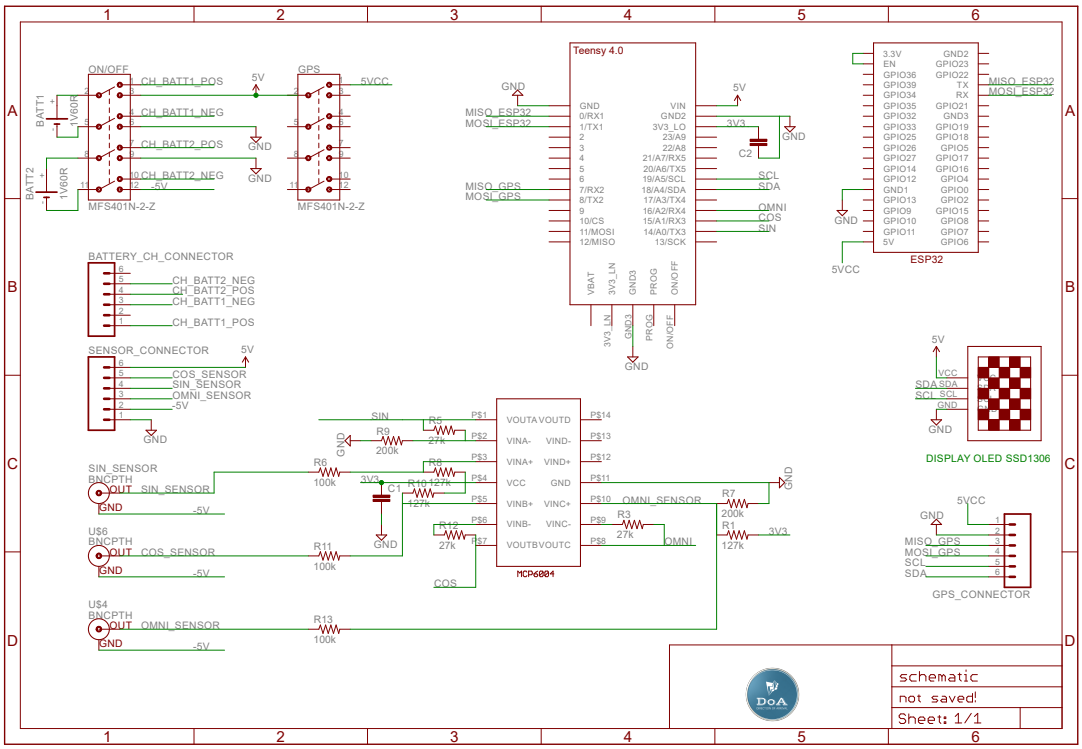

Figure 3.6. Schematic Circuit of the Sensor Node.

The components of every node are the following:

- Cosine sensor: MEMS sensor that provides a voltage variation as a response to the angle of arrival of the sound. The voltage is at a maximum when the angle of arrival is 0°.
- Sine sensor: this sensor will be oriented orthogonal to the cosine sensor so that the provided voltage will be 0 V when the angle of arrival is 0°.
- Omnidirectional sensor: commercially available omnidirectional MEMS microphone which detects the intensity of the sound without angle dependence of the arrival

sound.

- Battery module: Consists of a set of two 1-cell Li-Ion battery with a capacity of 2100 mAh. These batteries provide power to all the components.
- Regulator module: three linear regulators convert the voltage from the batteries to +5 Vdc and to -5 Vdc. Table 3.1 describes the power consumption for every component.
- Signal conditioning circuit: consists of two OPAMP components that convert the read voltage from the MEMS sensors to a 0–5 volts AC that can be measure for the microcontroller. The output voltage from the MEMS sensors ranges from ±2.8 V; this voltage is conditioned to obtain the required voltages by the microcontroller.
- Microcontroller unit: the Teensy 4.0 microcontroller described in 3.2, is the main component of every node. It performs the data acquisition calculations and provides an angle that corresponds to the angle of arrival of the sound.
- ESP32 microcontroller: this module is a single-core Xtensa 32-bit LX7 microcontroller running at 240 MHz with 320 kB of SRAM and 4MB SPI Flash. This unit has the following peripherals: 20 12-bit ADC channels, four SPI interfaces, one UART interface TTL-level, eight PWM channels and two 8-bit DACs. The unit also has wireless communications with protocol Wi-Fi 802.11 b/g/n (up to 150 Mbps) WPA/WPA2/WPA2-Enterprise/SPS. The current consumption of this device is 200 mA in normal operation and 550 μA in sleep conditions [31]. This unit provides Wi-Fi capabilities to the node.
- Antenna Module: an 2.4-GHz monopole antenna is connected to the ESP32 microcontroller to provide the radiofrequency link for the node.
- GPS module: this module provides satellite-based position data, this position data provided in an update rate of 1 Hz.
- Magnetometer module: The magnetometer determines the heading angle of the node with respect to the magnetic north of the Earth, using a 3-axis magnetometer capable of reading the Earth's magnetic field.
- Display: Connected through an I2C bus to the microcontroller, the display shows the current DoA.

3.4.2 Power Distribution

Every sensor node requires two 1-cell Li-Ion batteries to power all the electronic components. These batteries are connected to a 5 V linear regulator which provide power to every component. Table 3.1 shows a calculation of the current consumption of every sensor node.

Table 3.1. Calculation of the Current Consumption for the Sensor Node.

Battery	Component	Current
Battery 1	MEMS Sensors	~1 mA
	Teensy	~100 mA
	Signal Conditioning Circuit	~3 mA
	LCD display	~30 mA
	ESP32	~200 mA
	GPS/Magnetometer	~35 mA
Battery 2	Negative Voltage	~ 1 mA

Based on the information of table 3.1, the current consumption for the Battery 1 is 268 mA, therefore the battery will last at least 4.8 hours. In the case of Battery 2, the current consumption is 1 mA. The endurance of this battery will guarantee 17.6 hours of continuous operation.

Due to the chemistry of the batteries, fully discharging them will reduce their reliability. Therefore, it is recommended that continuous use be limited to a maximum of 5 hours.

The sensor box, shown in figure 3.4, can be mounted on a tripod for normal operation. In this position, the box will be exposed to the weather conditions. Because the housing box is made of anodized aluminum, further considerations should be addressed when exposing the box to the sun. A wrap in aluminum foil was required in the test site to reduce the overheating of the internal components. On hot days, the external temperature was higher than 40° C. A permanent solution of the overheating issue on the sensor box is to include a fan cooling system. Adding a 5-volt fan powered from Battery 2 in the sensor box will consume 200 mA. Therefore, Battery 2 endurance will be reduced from 17.6 hours to 6.4 hours.

3.5 Low-level Firmware Description

To achieve the goal of the node sensor, its tasks are divided between the two microcontrollers in the node module. The Teensy microcontroller returns data acquisition, and the ESP32 executes the communications.

3.5.1 Data Acquisition Microcontroller

The data acquisition microcontroller, shown in figure 3.7, runs embedded software written in C++. This firmware performs the different tasks such as reading the analog values from the sensors, parsing the telemetry provided from the modules and sending the position information and angle of arrival to be displayed in the GUI.

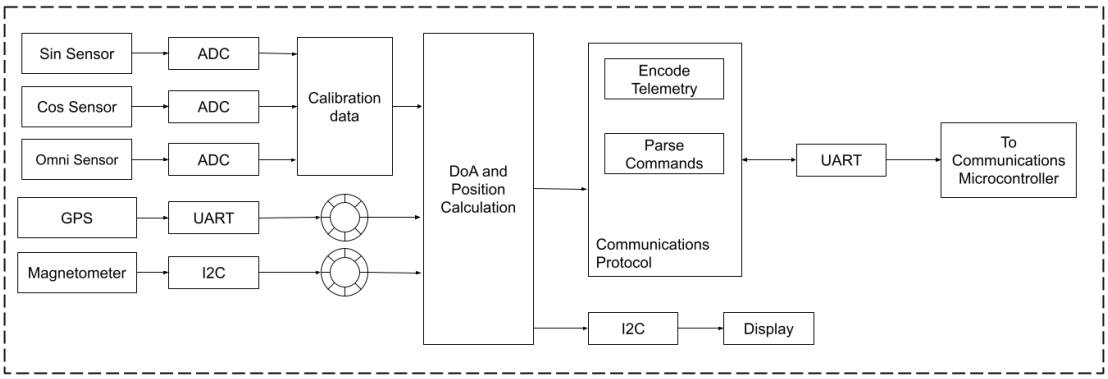

Figure 3.7. Low-level Architecture of the Sensor Microcontroller.

Figure 3.7 shows the placement of the ADC, I2C, and UART within the sensor microcontroller. They can be described as follows:

- Analog to Digital Circuit (ADC): three channels of a 12-bit successive-approximation analog-to-digital converter circuit are used to sample the provided voltage by the three sensors. The ADC peripheral runs at a sample frequency of 13 800 samples per second, measuring a total of 4096 data samples from every sensor. The total sampling time is about 300 ms. This sample time gives a total of three new angle determination per second. This information is stored in the SRAM memory of the Teensy microcontroller in a scale 0–1024 units, scaled 0–3.3 volts.

- Universal Asynchronous Receiver-Transmitter (UART): two ports are used in the Teensy microcontroller. UART0 is set to send the estimated values and receive the required commands from the user. This telemetry port is configured to run at a speed of 115 200 bps with eight bits of data, one stop bit and no parity. At this frequency, the maximum bandwidth available is 10.9 kB/s. The second UART used is UART2, which is used to read the data coming from the GPS module. This serial port is configured to run at a speed of 9600 bps with eight bits of data, one stop bit and no parity. The GPS module sends position at a refresh rate of 1 Hz. This information is stored in a circular buffer using for this task, a direct memory access peripheral (DMA) to allow the CPU to parse this information and determine the position of the node.
- Inter-Integrated Circuit (I2C): The I2C peripheral is used to communicate with the magnetometer module and determine the heading angle of the sensor node with respect to the true north of the Earth. The model WGS-84 of the Earth is used to perform the calculations. The second module connected to the I2C peripheral is the display. The microcontroller displays the estimated angle of arrival of the sound in the display.

The program loaded in the microcontroller consists of two parts, the first part is the initialization of the peripherals used in the Teensy. The second part is the DoA estimation. Figure 3.8 shows the tasks flow used in the data acquisition microcontroller.

- Process commands: Another function of the data acquisition microcontroller is to execute subroutines per every command received from the user. The commands are the following:
  - Adjust the threshold level: The threshold level is a number, in a scale of 512 to 1024 counts, that sets the minimum signal level collected by the firmware to be considered a valid signal, while lower values are considered as noise. This level is important in noisy environments to avoid false positives.
  - Detection mode: There are two detection modes: continuous time, which enables the process to run indefinitely and single shot, for detecting the first strong signal.
  - Set position: The application is always sending the GPS position directly read from the GPS module, but this position is not calibrated. This means that this position might not be the same every time it is read. For this reason, as a part of

the initialization routine, the GUI averages the position and sends the average (fix position) to the node.

– Set developer mode: This commands enables the software to send every recorded variable, such as the analog values, timing values, errors, warnings. This is useful to debug the firmware.

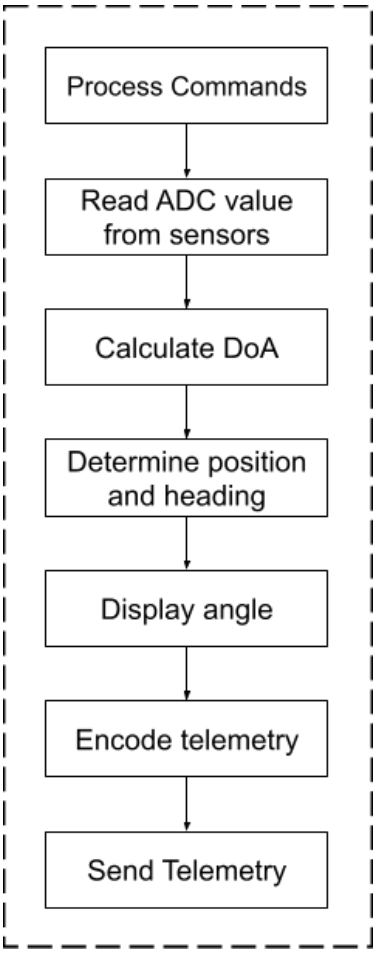

Figure 3.8. DoA Acquisition Flow.

- Read ADC values: The second task in the sequence is to sample the three ADC channels for 300 ms. The module is a 12-bit ADC running at a frequency of 13 800 samples per second. These measurements are stored in the SRAM memory of the

microcontroller, and are used to calculate the angle of arrival of the sound.

- Calculate DoA: This is the first task of main the routine. In this task, the microcontroller reads the three analog channels and applies a threshold comparison if the signal level is enough to set that a new detection has been found. Then, the microcontroller applies calibration parameters and signal processing to compute the angle of arrival of the incident sound. This angle is in the range of -180 – +180°.
- Determine position and heading: In this task, the microcontroller reads the GPS position. If the GPS position has changed since the previous measurement, the microcontroller updates the position variable. The same task is conducted for the heading angle; the microcontroller reads the heading angle from the magnetometer and updates the variable. In section 3.6 these variables will be discussed.
- Display angle: In this task, the microcontroller updates the angle displayed in the display module. This module is controlled using I2C.
- Encode telemetry: Telemetry is the useful information to be passed to the GUI. In this task, the microcontroller updates the variables in the communications protocol that will be sent to the GUI.
- Send telemetry: Now that the required variables of the communications protocol are updated, the microcontroller sends this information through the UART0 module to the communications microcontroller.

3.5.2 Communications Microcontroller

The ESP32 is the bridge between the collected data from the data acquisition microcontroller and the GUI. It establishes a TCP link and broadcasts the telemetry data. Figure 3.9 shows the low-level firmware architecture of this microcontroller.

The communications microcontroller reads the telemetry coming from the data acquisition microcontroller using the following peripherals:

- Universal asynchronous receiver-transmitter (UART): one port of the ESP32 microcontroller is used. In this case, UART0 is set to receive the telemetry data coming from the Teensy. This telemetry port is configured to run at a speed of 115 200 bps with eight bits of data, one stop bit and no parity. At this frequency, the maximum bandwidth available is 10.9 kB. This configuration should match with the same con-

figuration of the Teensy microcontroller to be able to establish a communications link.

- Wi-Fi communications: The standard for IEEE 802.11g [30] describes the requirements to establish a local area network using a radio frequency of 2.4 GHz. The ESP32 microcontroller is programmed to be connected to an existing Wi-Fi network. Then, the microcontroller configures a TCP socket to establish a bidirectional communication with the GUI. This virtual link is used to send the received telemetry from the data acquisition microcontroller to the GUI.

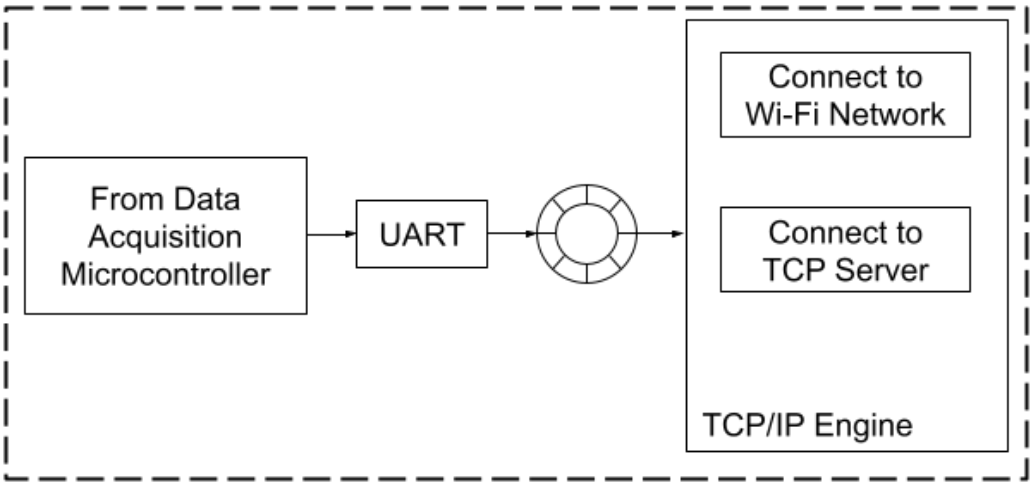

Figure 3.9. Low-level Architecture of the Communications Microcontroller.

For this microcontroller, three tasks will be running at the same time. To do so, the events feature of the ESP32 will be configured. The events feature of the microcontroller triggers an interruption when a flag is set. The flag is set in hardware for the Wi-Fi engine whenever the module acquires/loses connection, receives data, or succesfully sends data. The following tasks are executed by the communications microcontroller:

1. The first task will be to connect to a network. The event system of the ESP32 monitores whether the microcontroller is connected to a Wi-Fi network and has a valid IP address. In case of disconnection, it tries reconnect to the network.
2. The second task is to check that the microcontroller is connected to a TCP server.

When the microcontroller has a valid IP address, this task is launched. The GUI will provide a TCP server, and it will be always listening for sockets to connect. If the server is closed and the microcontroller has a valid IP address, the routine will try to reconnect to the server.

3. The third task is to send the telemetry to the GUI and forward the received commands to the data acquisition microcontroller.

3.6 Communications Protocol

A common communications protocol should be used to communicate the node sensor with the GUI. For this work, the MavLink protocol was chosen. MavLink is an open-source protocol used for drones [32]. Table 3.2 shows the definition of the messages used in both the microcontroller and the GUI.

Table 3.2. Definition of the MavLink Messages.

Message	Field	Description
HEARTBEAT	type	Component type
	system_status	Status of the component: online, warning, emergency
HEARTBEAT	mavlink_version	Protocol version
APP_BOOT	hardwareRevision	Current hardware revision
	firmwareVersion	Current firmware version
APP_BOOT	reset	Previous reset cause
APP_SYSTEM	cpuLoad	CPU load in % (0 – 100%)
	voltage	Input voltage in volts
	temperature	Temperature in Celsius × 100
	current	Drained current × 1000
	uptime	Up time Hours, minutes and seconds in the following format: HHMMSS, i.e. 235959 is 23:59:59
APP_ACTION	componentId	Command number
	componentOrder	Current order or set value

APP_STATUS	bearing	The offset angle to the acoustic front
	threshold	Actual Threshold used
	hms	Hours, minutes and seconds in the HHMMSS-mmm format.
	beta	Calculated angle of arrival
	lat	Calibrated Latitude in °
	lon	Calibrated Longitude in °
	hdg	Heading angle in °
	alt	Altitude in meters
	yaw	Yaw angle in textdegree
	pitch	Pitch angle in °
	roll	Roll angle in °
GPS_RAW	lat	Latitude in °
	lon	Longitude in °
	alt	Altitude in meters
	fix_type	0: Invalid, 1: GPS fix, 2: DGPS fix, 3: PPS fix, 4: RTK, 5: float RTK, 6: deadreckoning
	eph	Horizontal accuracy in meters
	epv	Vertical accuracy in meters
	sats	Number of visible sats used in the solution
	v	GPS ground speed, also known as speed of ground (SOG)
	hdg	Compass heading in °, 0-360 °, also known as course over ground (COG)
	hms	Hours, minutes and seconds in the HHMMSS format.

3.7 Description of the Graphical User Interface

The GUI is responsible to display all the telemetry coming from the sensor node and sends commands. Figure 3.10 shows an image of this GUI.

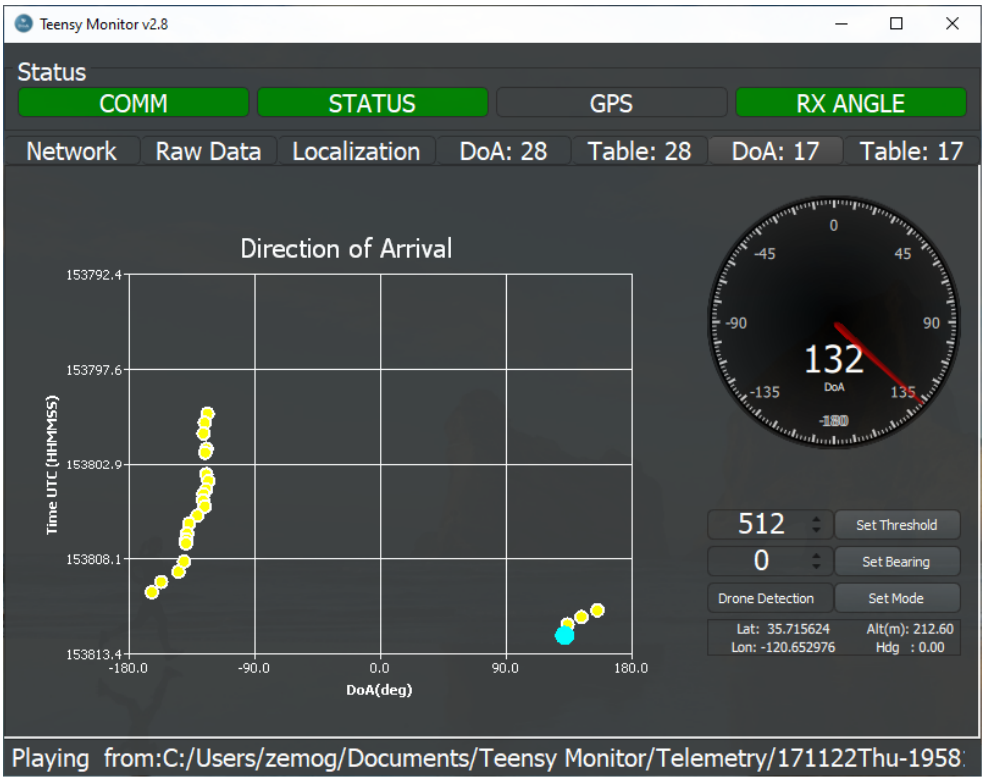

Figure 3.10. Graphical User Interface (GUI).

The main purpose of the software is to provide the user with a robust software that can record the data coming from every node and display in real time different characteristics:

- Create a sensor network: This software can handle a set of different nodes. The software is capable of creating a TCP service to listen and provide service to different types of clients. The server passed the 10 thousand stations test. As a client, the software can work as a mirror from the software server allowing us to run multiple instances of this software in different stations connected to the same local network. The software is capable of communicating with the sensors using a Serial Port, which is extremely useful to debug and test new features in the firmware.
- Display data: The software is capable of displaying the current angle of arrival of every sensor and previous values as well.
- Send commands: Commands like “adjust the threshold,” “set a bearing offset,” “set continuous or single events.”

- Log data: All the received telemetry is stored in a master file. Independent logs are created for every connected node.
- Display position data: Every node is displayed in an Open Street Maps. The current and previous estimated positions of the drone are also displayed.
- Replay telemetry: After a test, every logged data can be replayed in real time or the file can be speeded up.

3.7.1 Software Architecture

The software application was written in C++ using the Qt framework [33] and runs natively in Microsoft Windows 7 and later versions. The purpose of this software application is to receive multiple sensor nodes at once and connect to the Open Street Maps service [34] to display in the map the position of the nodes and the estimated position of the drone.

Figure 3.11, shows the software data flow. Every node represents a thread in the application:

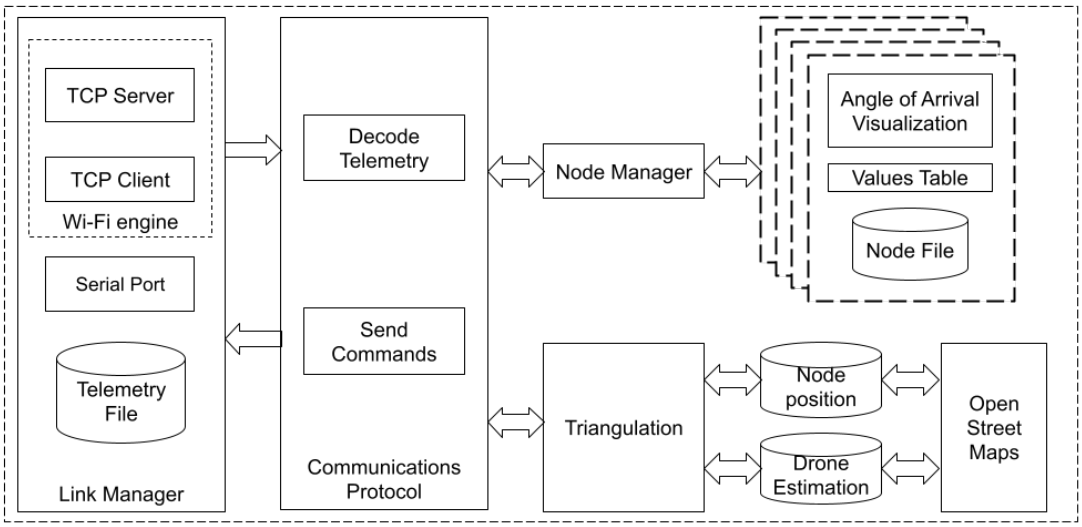

Figure 3.11. GUI Data Flow.

The data flow is described as follows:

- Link manager: The link manager connects the software applications with a data stream available. It handles tasks such as electrical considerations, data timing, data

corruption and logic levels. The data can come from the following sources:

- Serial port: In Figure 3.12, the serial port widget is shown in the top left part. It has the options to select the COM port and the baud rate (data speed). This is a wired option and is meant to work locally with a single sensor node at a time. The software then connects to this port and reads telemetry coming from the sensor.
- Network client: As mentioned before, the software application can run on different computers. When using more than one computer, the software can read the telemetry as a client. Then, the application can interact with the nodes as the master computer. Another option is to set the communications link to listen-only mode. In this configuration, the application can read the telemetry, but cannot send commands. They are ignored by the application. In figure 3.12 top right, the options for network server are shown. The computer should be connected to the same local network, and the IP address of the master node should match with the previously configured IP address in the sensor nodes. The TCP port used by default is 5000, but other tcp ports can be used.

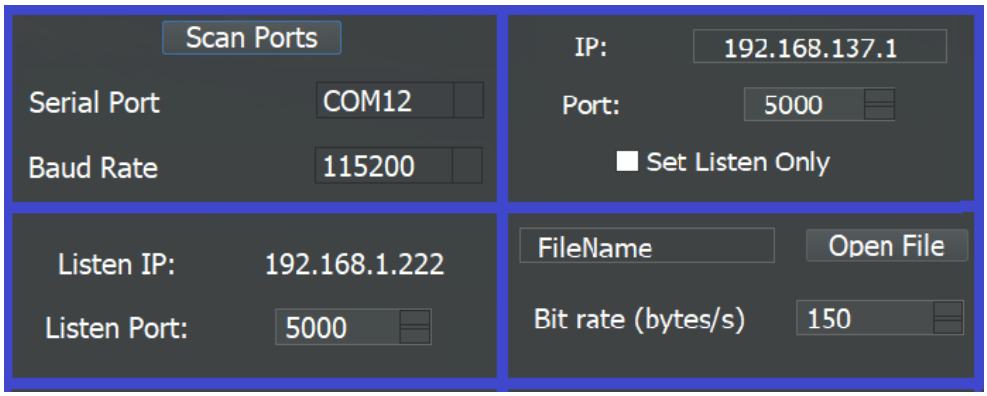

Figure 3.12. GUI Communications Options. Top Left: Serial Port; Top Right: Network Client; Bottom Left: Sensor Server; Bottom Right: Replay File.

- Sensor server: In this configuration, shown in figure 3.12, the software application creates a sensor network. This network is used by the available sensors connected to the same local network. Every node is read independently and identified by the ID of the node. In the options, presented in figure 3.12, the local IP address and the selected tcp port are shown.

  - Replay file: When the application is connected to a valid link such as serial port, sensor server, and network client, a logging routine is launched to make entries in a "tel" file to store all the received telemetry. This log file can later be used to review the sensors behavior by selecting the option shown in figure 3.12 bottom right. In this option, the bit rate can be selected to run the file in real time, sped up or slowed down. This tool is useful for post-processing.
  - Server: When the software is connected through replay file, sensor server and serial port, there is an option to enable a TCP server to allow more computers to connect to the network. For this option the TCP port should be selected.
- Communications protocol: This layer receives the bytes from the link manager. These bytes are parsed using the MavLink protocol engine. The message definition is detailed in section 3.6. Because the MavLink protocol is message oriented, when the parser detects a new message ID, it communicates with the node manager to create a new node. When the parsed message corresponds to an existing node, it sends this message to the node manager. The communications protocol encodes the orders sent by the nodes through the node manager. These orders are converted to a byte stream, and they are sent to the link manager to be transmitted.
- Node manager: The node manager provides service to all connected nodes to the application software. It is responsible for sending the correct messages to the right Node, and it encodes all the commands to be sent to the sensor nodes. At start-up, the node manager is empty but as soon a new message ID is parsed in the communications protocol, a new node handle is created.
- Node: A node represents a physical sensor node. In figure 3.13 the GUI of a node is shown. In the left it is possible to see how the DoA is graphically displayed. The current DoA is displayed in the form of a dial, while the chart shows the DoA trend for the last 20 seconds. In the chart, every dot represents a DoA estimation. In the figure on the right, all the values are displayed in a form of a table. The values shown are latitude, longitude, and heading of the sensor node, DoA estimated, time stamp, threshold value, and offset value. Besides the telemetry file saved, a comma-separated values (CSV) file is created, storing all the node variables aforementioned.

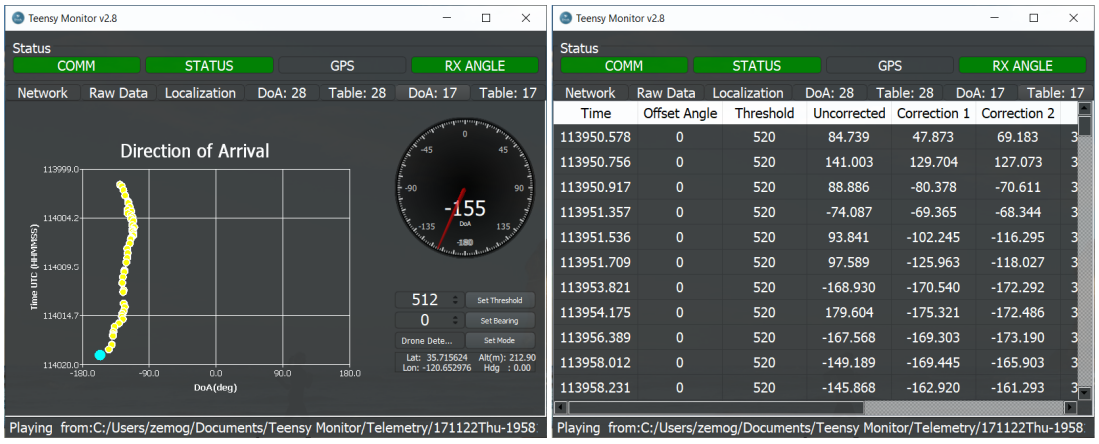

Figure 3.13. Software Node Widget. Left: Graphical display of the DoA in degrees vs time in seconds. Every dot represents a new estimation. Right: Variables displayed in a table.

- Triangulation: One of the purposes to create a software application is to estimate the position of a small rotor UAV based on the DoA values from at least two sensor nodes. The software application integrates a plugin that connects to the maps service provided by Open Street Maps [34]. The Open Street Maps service provides all the map images used in the software application to display the position of the sensor nodes and the estimated position of the UAV.

  Figure 3.14 contains the position of two nodes and the estimated position of the drone. As the software receives telemetry from the nodes, the position of the drone will be updated through time. The current algorithm only uses the position and angle of arrival data from two sensor nodes.

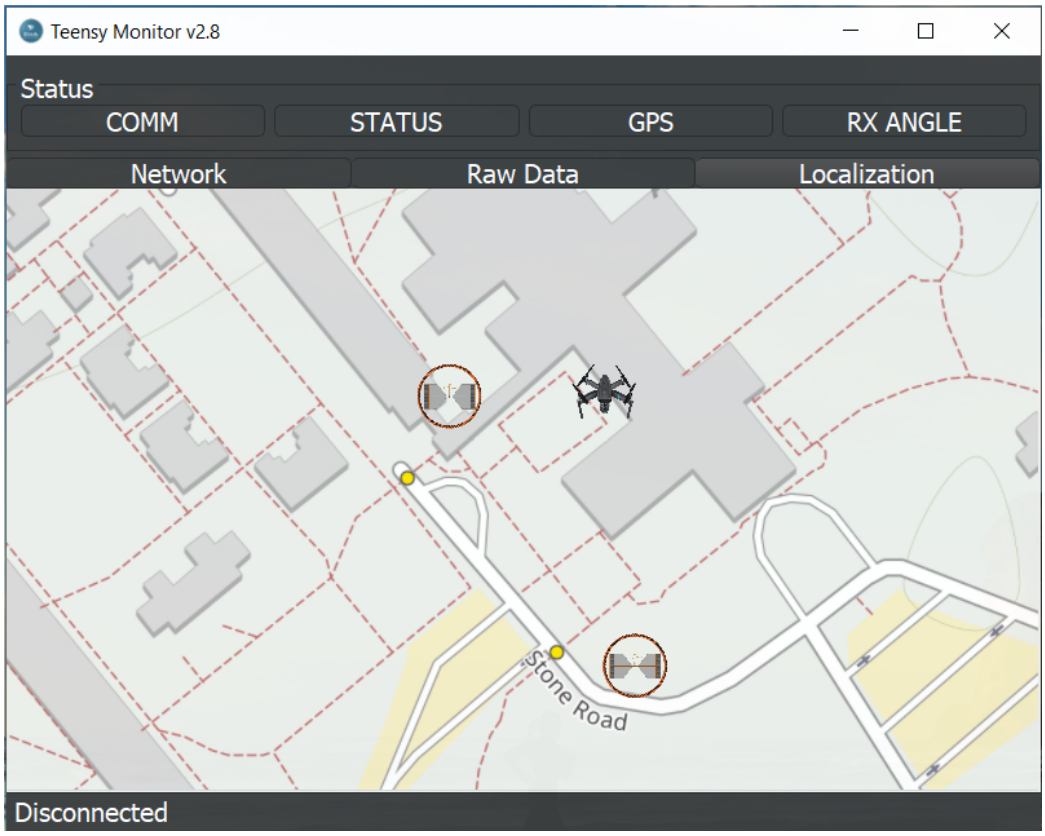

Figure 3.14. Software Initial Map Representation. Source: [34].

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

The proposed architecture (described in chapter 3) is implemented to collect data in the field test. The purpose of the test is to estimate the DoA of the incident sound emitted by a small drone. The estimation of the position of the drone is done in real-time. To do so, it is necessary to set up the sensor network. When the nodes are configured, they need to be placed in a triangular geometry to start collecting and processing data. The test setup and the localization estimator are described in this chapter as well as the result of the field test.

4.1 Test Setup

To set up a test, there are some important parameters to configure in the sensor nodes and in the GUI. These parameters include operational frequencies, IP addresses, TCP port numbers, and baud rates. Settings are listed in Table 4.1. All the sensor nodes connected to the same local area network (LAN) are able to send telemetry to the GUI. The architecture of this network is ID-number oriented, this means that every sensor node has a different ID number, regardless of their IP address.

Table 4.1. Configuration Parameters for the Field Tests.

Node	Parameter	Description
Master GUI	IP address: 192.168.137.1/24	The software should be set as "Sensor Server"
	TCP Port: 5000	Default TCP port
Sensor Node XX	IP address: 192.168.137.XX/24	First Node. The ID of the nodes should be different
	TCP Port: 5000	Default TCP port
Client GUI XX	IP address: 192.168.137.XX/24	The software should be set as "Network Client"
	TCP Port: 5000	Default TCP port

Additional Sensor	IP address: 192.168.137.XX/24	The software might be set as “Serial Port” or “Replay File”. The option “Enable Server” should be selected
	TCP Port: 5000	Default TCP port

4.2 Localization

In the field test, the haversine formula is used to determine the position of the drone. The formula is included in the aviation formulary available in [35]. This estimator calculates the geographic coordinates of a third point based on the knowledge of the position of two points and the angle between each known point and the unknown point. A graphical representation of the haversine formula is shown in figure 4.1. In this figure, it is seen that the three points can form a circumscribed triangle. The position of the third point is calculated from the distance between two points and the angle of the two points with respect to the third point using geometry identities and using cartesian coordinates.

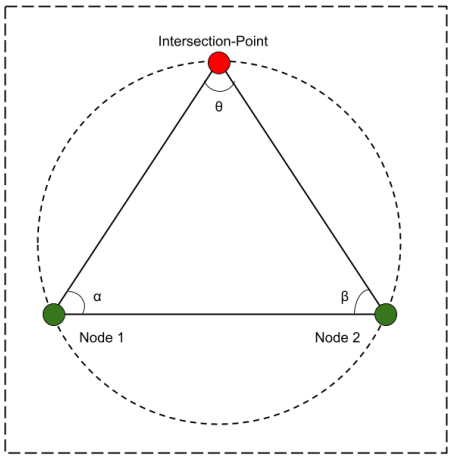

Figure 4.1. Graphical Representation of the Triangulation Estimator. The position of a third point can be estimated by knowing the position of two points and the angle between each point and the third one.

The estimator starts calculating the angular distance between the two points (δ_{12}), shown in Equation 4.1:

$$\delta_{12} = 2\arcsin(\sqrt{\sin^2(\Delta\psi/2) + \cos(\psi_1)\cos(\psi_2)\sin^2(\Delta\lambda/2)}), \tag{4.1}$$

where $\Delta\psi$ and $\Delta\lambda$ represent the latitude and longitude difference between the two points, expressed in degrees. This angular distance is used to determine the initial (θ_a) and final (θ_b) bearings presented in 4.2 and 4.3.

$$\theta_a = \arccos(\sin(\psi_2) - \sin(\psi_1)\cos(\delta_{12}))/(\sin(\delta_{12})\cos(\psi_1)) \tag{4.2}$$

$$\theta_b = \arccos(\sin(\psi_1) - \sin(\psi_2)\cos(\delta_{12}))/(\sin(\delta_{12})\cos(\psi_2)) \tag{4.3}$$

Now, using the heading angles between the two points and the third point ($\theta_{13} = \alpha$ and $\theta_{23} = \beta$), the intermediate angles α_1, α_2, α_3, are calculated. From these angles, the angular distance between point 1 and point 3 (δ_{13}) is determined as follows:

$$\alpha_1 = \theta_a - \theta_{12} \tag{4.4}$$

$$\alpha_2 = \theta_b - \theta_{23} \tag{4.5}$$

$$\alpha_3 = \arccos(-\cos(\alpha_1)\cos(\alpha_2) + \sin(\alpha_1)\sin(\alpha_2)\cos(\delta_{12})) \tag{4.6}$$

$$\delta_{13} = \arctan 2(\sin(\delta_{12})\sin(\alpha_1)\sin(\alpha_2), \cos(\alpha_2) + \cos(\alpha_1)\cos(\alpha_3)) \tag{4.7}$$

Finally, the latitude (ψ_3) and longitude (λ_3) coordinates of the third point are determined from the following expressions:

$$\psi_3 = \arcsin(\sin(\psi_1)\cos(\delta_{13}) + \cos(\psi_1)\sin(\delta_{13})\cos(\theta_{13})) \tag{4.8}$$

$$\lambda_3 = \lambda_1 + \arctan 2(\sin(\theta_{13})\sin(\delta_{13})\cos(\psi_1), \cos(\delta_{13}) - \sin(\psi_1)\sin(\psi_3)) \tag{4.9}$$

4.3 GPS Data Log

Now that the two nodes are assembled and the localization estimator is implemented in the GUI, the tests can be executed. A GPS data logger is the final component needed to test the localization estimator. This system is a simple GPS module connected to a microcontroller that stores the position data from the drone while it is flying. Figure 4.2 shows an image of the data logger and how it is mounted on a drone.

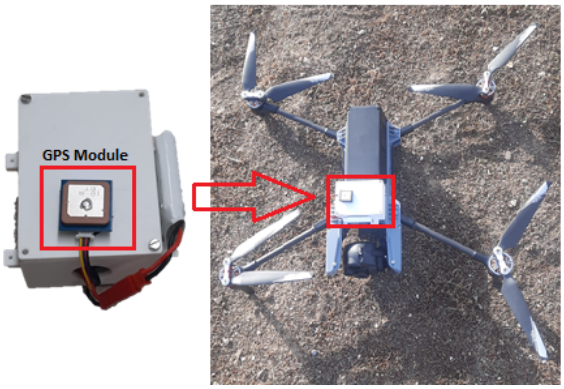

Figure 4.2. GPS Data Logger Used to Store the Position of the Drone. Left: GPS module, right: module mounted on a drone used in a test field.

4.4 Field Test Results

In order to test the functionality of the system, three field tests were performed. Two tests were executed in Paso Robles, CA and one more in Oceanside, CA.

The tests were used to debug the systems. The effective range was the first performance test conducted on the site. Wi-Fi communications were tested at a distance of 100 meters. This was a successful test. There were some environmental conditions challenges, such as overheating of the sensor box that were addressed on site. A communication-loss failure was presented due to this overheating. As a workaround, aluminum foil was used to wrap the sensors box to protect it from the sun exposure.

Battery endurance was the second test performed at the site. The sensor nodes were operated

for at least six hours continuously. At the end of the day, the voltage of the batteries was measured. If the sensor nodes are required to be used for more than six hours continuously, the onboard batteries should be replaced with batteries that can provide more energy.

for the operational scenario test, two sensor nodes were mounted on tripods and placed in the field. The sensor nodes were linearly aligned to allow the drone to fly between these nodes.

Figure 4.3 shows the location of the sensor nodes, a flying drone, and the GUI at the test site. The sensor nodes are highlighted in red, the flying drone is highlighted in green, and the control station is highlighted in purple.

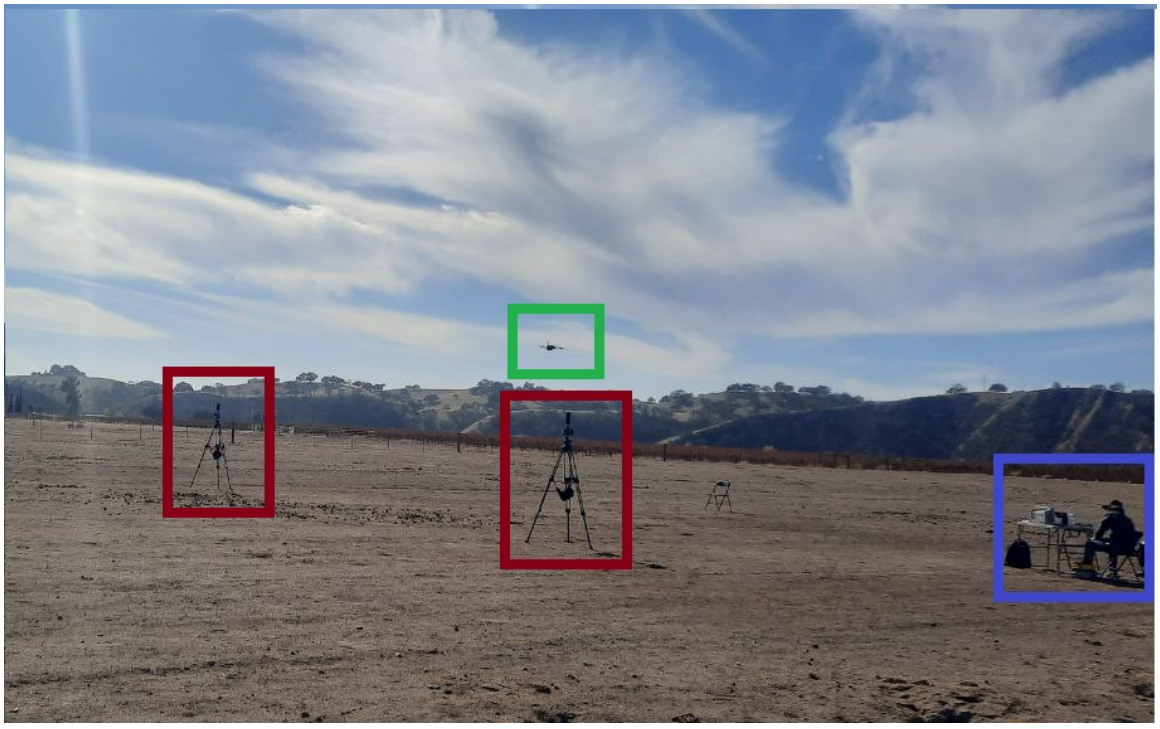

Figure 4.3. Paso Robles Test Setup. Sensor Nodes (red), drone (green) and, Control Station (purple).

From the data obtained in the Paso Robles test, the coordinates of the two nodes are presented in Table 4.2.

Table 4.2. Position Data for the Sensor Nodes.

Node ID	Latitude	Longitude	Altitude	Heading
17	35.71542044	-120.6529758	21.5 m	0°
28	35.71575618	-120.6529599	21.3 m	0°

After passing the communications, power endurance, and component heating, the system is ready to be used to estimate the DoA from a drone. For this task, a drone was set up to execute flight patterns near one sensor node. The result of this test is shown in Figure 4.4. The x-axis represents the angle of arrival in a scale from -180 to 180°. The value of 0° corresponds to the middle of the x-axis. The y-axis shows the UTC time, read from the GPS module.

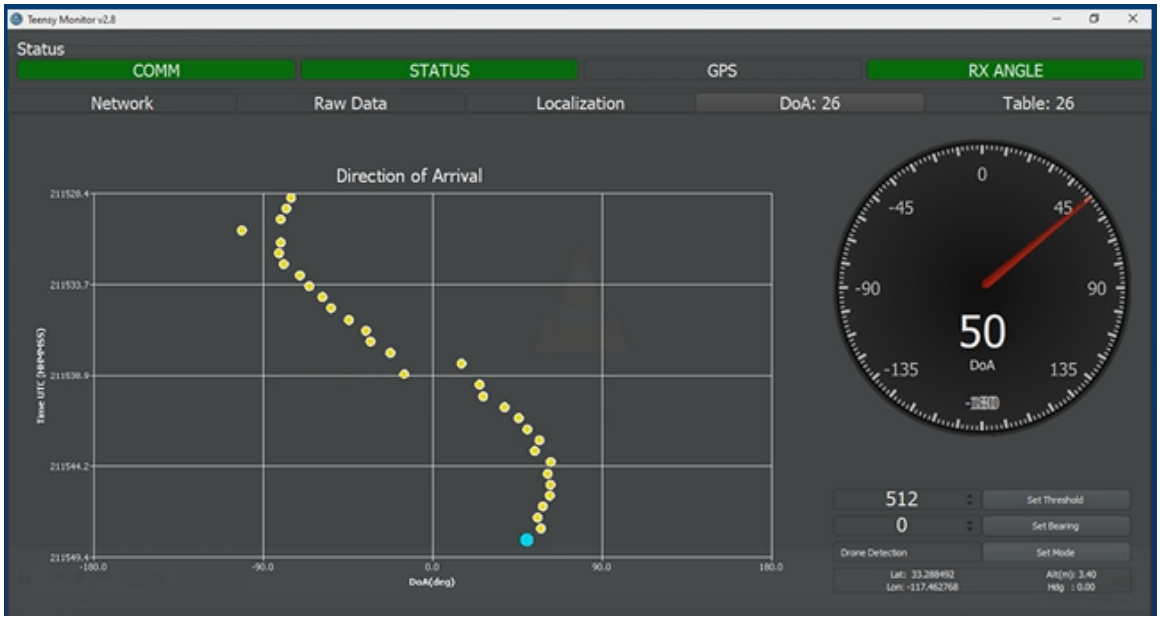

Figure 4.4. Paso Robles Test Setup DoA.

The trace represents a passing of the drone from left to right. Every dot represents a new measurement available from the sensor. Because a sensor node requires about 320 ms to

take a new measurement, the software will receive about two or three new measurements per second. These measurements were stored in a CSV file.

A total of 150 effective minutes of drone flight time was used to test the developed architecture. Multiple flight patterns were chosen to test the reliability of the localization estimator. Because this calculation is based on the triangle geometry, there are some spots where the error is more significant. The position data of the drones was extracted from the GPS data logger, detailed in Section 4.3, to compare the estimated position from calculated in the GUI with the actual drone position. This comparison is showed in Figure 4.5.

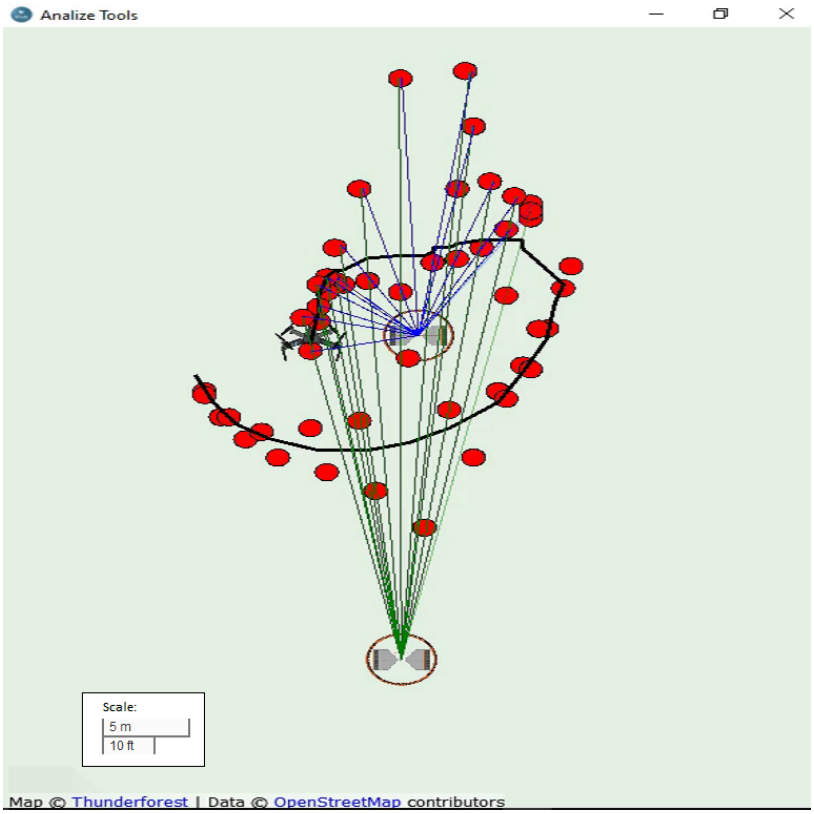

Figure 4.5. Paso Robles Test Setup: Estimation of the Drone Position (red dots) vs the Actual Position of the Drone (solid black line). Node coordinates presented in Table 4.2.

In the map, the position of the two node sensors are represented by a small figure of the MEMS sensor image, as shown in Figure 4.5. Every sensor node has straight lines that represent the DoA sent by the sensor node. A red dot represents the crossing point of two lines from each sensor node. This red dot is an estimated position of the drone. The black line shows the actual path of the drone and the small drone represents the latest recorded position.

Figure 4.5 displays the ability of the localization estimator to detect and track the position of the drone. There are some limitations in the calculation, especially when the drone is aligned with the two sensors. This occurs at 0° and -180° . This limitation is due to the triangle geometry, as the angle between the nodes and the drone gets smaller, the calculation error increases. When the drone moves to other positions, the error in the estimation is reduced. Figure also shows how this error is reduced as the bearing angle increases.

The accuracy of the drone estimation is affected by the position of the node sensors and the collected position of the drone. In the sensor nodes, commercially available GPS receptors were used. The range error for these modules ranges in decimeters, especially affecting for shorter distances. This measurement error affects the accuracy ability of the estimator. To reduce this error, the use of more reliable GPS receptors should be explored.

CHAPTER 5:
Conclusions

5.1 Summary

The MEMS sensor designed and developed in the SRL at NPS is used to determine the DoA using a set of three sensors arranged in a cross-dipole configuration and read by a microcontroller. This microcontroller is wired-connected to a computer to display the angle of arrival of the sound.

In this work, the design of a new system architecture to determine the DoA from more than one sensor simultaneously was developed. An estimated position of small UAV was determined from the data collected from two sensor nodes. This architecture was designed to be adapted to the previous DoA determination. The drone localization estimator required knowledge of the position coordinates of the sensor nodes. This estimator also required that the sensor nodes were placed in different locations. GPS readings and Wi-Fi communications were provided to the sensor nodes to increase the distance between the two nodes and the control station. Using this approach, the implemented architecture was able to collect and broadcast acoustic data from the drone. The collected data was used to estimate the position of the drone in real-time.

Three field tests were conducted to debug the functionality of the new architecture. In these tests, the sensor nodes were set up to estimate the DoA of the drone. Successful individual tests were performed on the site, including battery autonomy, weather conditions endurance, and communications range. The functionality of the GUI was tested. The control station showed in real time the estimated DoA sent by the sensor nodes. In the tests, the position of the drone was estimated and displayed in the GUI. This estimated drone path was compared with the actual drone path recorded with a GPS data logger installed on the drone. The estimator showed a good performance in the 360° of the sensor nodes assembly. Small errors were presented in the drone estimation when it was flying aligned with the two sensor nodes at 0° and 180°.

As a result of this work, the following systems were designed and incorporated to the current

architecture of the SRL lab:

- The design and implementation of a sensor node with Wi-Fi capabilities, GPS acquisition routine, heading position readings from a magnetometer. This design was fully integrated with the previous architecture of the SRL.
- The construction of a GUI used to display the angle of arrival of various sensor nodes at the same time. With capabilities to establish a sensor network and provide the infrastructure to connect more computers to the same network.
- The implementation of a communications protocol.
- The creation of a data log system to store and replay the telemetry collected from the sensor nodes.
- The design and test of a firmware application for two different sets of microcontrollers with routines of collect and broadcast data to the GUI.
- Finally, a triangulation routine was integrated in the GUI. This routine displayed the estimated position of the drone in real time.

Additionally, two more tools were designed to test the efficiency of the previous components:

- A GPS data log system, mounted aboard a drone programmed to record its position data in a CSV file.
- A software analysis tool to display in a map the both the actual and the estimated position of the drone as shown in Figure 4.5.

Overall, this new system architecture successfully proved its ability to detect and track small UAV, providing the SRL with a new test framework for the developed MEMS sensors.

5.2 Future Work

The implemented architecture successfully estimated the position of a drone in real-time within a few centimeters accuracy. To increase the accuracy and robustness of the designed system, the following are suggested:

1. Replace the GPS receptor in the sensor nodes by a military grade GPS. With this replacement, the accuracy in the drone localization estimator will be significantly increased.

2. Explore different triangulation methods to provide the drone position using more than three sensor nodes. This change in the estimator will reduce the error at 0° and 180°. Using more than three sensor nodes will increase the accuracy in the estimation of the drone position.
3. Synchronize the start of the sampling time in the sensor nodes to guarantee that every sensor node is sampling the same position of the drone. This will avoid out-of-phase sensor readings.
4. Include a dedicated analog-to-digital (ADC) module in the sensor nodes. This module can increase the resolution of measurements of the MEMS sensors and reduce the electronic noise providing a higher sensitivity in the overall system.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] R. Austin, *Introduction to Design and Selection of the System*. Chichester, West Sussex, U.K.: John Wiley & Sons, Ltd, 2010, ch. 2, pp. 17–24. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470664797.ch2

[2] B. Taha and A. Shoufan, "Machine learning-based drone detection and classification: State-of-the-art in research," *IEEE Access*, vol. 7, pp. 138 669–138 682, 2019.

[3] D.-H. Shin, D.-H. Jung, D.-C. Kim, J.-W. Ham, and S.-O. Park, "A distributed fmcw radar system based on fiber-optic links for small drone detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 2, pp. 340–347, 2017.

[4] F. Yang, F. Xu, F. Fioranelli, J. Le Kernec, S. Chang, and T. Long, "Practical investigation of a mimo radar system capabilities for small drones detection," *IET radar, sonar & navigation*, vol. 15, no. 7, pp. 760–774, 2021.

[5] R. Barretta and J. Melkert, "UAV visual signature suppression via adaptive materials," in *12th Annual International Symposium on Smart Structures and Materials, San Diego, California*, 2005, pp. 6–10.

[6] "Sonar," *Britannica, The Editors of Encyclopaedia*. Accessed November 2, 2022 [Online]. Available: https://www.britannica.com/technology/sonar

[7] M. Lanzagorta, *Underwater Communication Channels* (Synthesis lectures on communications, # 6). San Rafael, CA, USA: Morgan & Claypool Publishers, 2013, ch. 2, pp. 27–42.

[8] R. A. Rayburn, *Eargle's Microphone Book: From Mono to Stereo to Surround: A Guide to Microphone Design and Application*, 3rd ed. Waltham, MA, USA: Focal Press/Elsevier, 2012.

[9] A. S. Matveev, A. V. Savkin, M. Hoy, and C. Wang, "12 - safe cooperative navigation of multiple wheeled robots in unknown steady environments with obstacles," in *Safe Robot Navigation Among Moving and Steady Obstacles*. Elsevier Inc, 2016, pp. 283–311.

[10] J. N. Yasin, S. A. S. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Unmanned aerial vehicles (UAVs): Collision avoidance systems and approaches," *IEEE Access*, vol. 8, pp. 105 139–105 155, 2020.

[11] A. Sedunov, H. Salloum, D. Haddad, A. Sutin, N. Sedunov, and A. Yakubovskiy, "Field test comparison of various acoustic drone detection methods," *The Journal of the Acoustical Society of America*, vol. 146, no. 4, pp. 2783–2783, 2019.

[12] J. Benesty, *Fundamentals of Signal Enhancement and Array Signal Processing*. Hoboken, NJ, USA: John Wiley & Sons Singapore Pte. Ltd, 2019.

[13] M. Benyamin and G. H. Goldman, *Acoustic Detection and Tracking of a Class I UAS with a Small Tetrahedral Microphone Array*. Adelphi, MD, USA: DEVCOM ARL, 2014.

[14] A. Rahaman, A. Ishfaque, H. Jung, and B. Kim, "Bio-inspired rectangular shaped piezoelectric mems directional microphone," *IEEE Sensors Journal*, vol. 19, no. 1, pp. 88–96, 2019.

[15] F. Alves, R. Rabelo, and G. Karunasiri, "Dual band mems directional acoustic sensor for near resonance operation," *Sensors (Basel, Switzerland)*, vol. 22, no. 15, pp. 5635–, 2022.

[16] R. Austin, *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*, 1st ed. (Aerospace Series). Chichester, West Sussex, U.K.: Wiley, 2010, vol. 54.

[17] "Matrice600," *Matrice 600 Pro*, 2018 [Online]. Available: https://dl.djicdn.com/downloads/m600%20pro/1208EN/Matrice_600_Pro_User_Manual_v1.0_EN_1208.pdf

[18] "Phantom3," *Phantom 3 Standard*, 2018 [Online]. Available: https://dl.djicdn.com/downloads/phantom_3/UserManual/Phantom_3_Standard_User_Manual_v1.4_en.pdf

[19] "Skydio," *Skydio X2D Color/Thermal*, 2015 [Online]. Available: https://pages.skydio.com/rs/784-TUF-591/images/skydio-x2e-datasheet.pdf

[20] "Yuneec," *Typhoon H Plus*, 2018 [Online]. Available: https://us.yuneec.com/wp-content/uploads/2021/01/TYPHOON-H-PLUS-RS_USER-MANUAL_V1.0_20180830.pdf

[21] J. Gundlach, *Designing Unmanned Aircraft Systems: A Comprehensive Approach* (AIAA education series). Manassas, VA, USA: American Institute of Aeronautics and Astronautics, Incorporated, 2014. Available: https://books.google.com/books?id=0123oAEACAAJ

[22] L. E. Kinsler, *Fundamentals of Acoustics*. Monterey, CA, USA: John Wiley & Sons, 1999.

[23] A. V. Oppenheim, *Signals and Systems*, 2nd ed. (Prentice-Hall signal processing series). Upper Saddle River, NJ, USA: Prentice Hall, 1997.

[24] W. Johnson, *Rotorcraft Aeromechanics* (Cambridge aerospace series, 36). New York, NY, USA: Cambridge University Press, 2013.

[25] G. Sinibaldi and L. Marino, "Experimental analysis on the noise of propellers for small UAV," *Applied acoustics*, vol. 74, no. 1, pp. 79–88, 2013.

[26] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, p. 22, 2011.

[27] D. Wilmott, F. Alves, and G. Karunasiri, "Bio-inspired miniature direction finding acoustic sensor," *Nature*, July 21, 2016.

[28] S. Davies, "Bearing accuracies for arctan processing of crossed dipole arrays," in *OCEANS '87*, 1987, pp. 351–356.

[29] "Pjrc store," *Teensy 4.0*, 2022 [Online]. Available: https://www.pjrc.com/store/teensy40.html

[30] "IEEE SA," *IEEE 802.11g*, 2022 [Online]. Available: https://standards.ieee.org/ieee/802.11/5536/

[31] "Expressif," *ESP32 S2*, 2022 [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

[32] "Mavlink," *MavLink v2*, 2009 [Online]. Available: https://mavlink.io/en/

[33] The Qt Company. (2022). Qt Creator. [Online]. Available: https://doc.qt.io/

[34] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[35] E. Williams. (2022). Aviation Formulary v1.47. [Online]. Available: http://www.edwilliams.org/avform147.htm#Intersection

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California