“AUTOMATED RADAR HEADING CALIBRATION WITH COLLABORATING PARTICIPANTS AND MULTI-SENSOR FUSION”

A Thesis

by

JOSH BOYD

BS, Texas A&M University-Corpus Christi , 2011

Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

Texas A&M University-Corpus Christi
Corpus Christi, Texas

December 2021

©Josh Boyd

All Rights Reserved

December 2021

“AUTOMATED RADAR HEADING CALIBRATION WITH COLLABORATING PARTICIPANTS AND MULTI-SENSOR FUSION”

A Thesis

by

JOSH BOYD

This thesis meets the standards for scope and quality of
Texas A&M University-Corpus Christi and is hereby approved.

Scott King, PhD
Chair

Longzhuang Li, PhD
Committee Member

Wenlu Wang, PhD
Committee Member

December 2021

ABSTRACT

As unmanned aerial systems (UAS) become more prolific so will the use of radar systems for tracking UAS in the national airspace system (NAS). The future of Urban Air Mobility (UAM) involves large amounts of UAS operating autonomously and simultaneously in urban environments for the purpose of passenger or cargo transportation. Radar detection of UAS in an urban environment can be hindered by line of sight (LOS) blockage by large buildings thus necessitating many surveillance devices to gain full coverage. Currently, UAM is still in development by the Federal Aviation Administration (FAA) and other airspace partners and many cities do not have the need or resources for full radar coverage. Due to the high cost of individual radar systems and the quantity needed to cover urban areas it is currently not practical to have full radar coverage of an area at all times. Permanent stationary radar systems are generally calibrated once with occasional adjustments and low time constraints. Temporary radar systems must be calibrated and aligned before each mission deployment and often under short time constraints. Temporarily stationed mobile radar platforms will be utilized for specific targeted mission objectives until a more permanent solution is developed and implemented. In the case of disaster response or search and rescue, a temporary radar system needs to be quickly deployed. The key abilities required by a temporary radar system are accurate track position reporting and quick setup and breakdown. One of the bottlenecks to quick setup is heading calibration. Radar antenna alignment is crucial to the performance of the system and its ability to accurately determine the position of a tracked object.

In this paper, we implement and compare multiple methods of radar heading calibration for accuracy and speed including manually with a handheld compass, manually with a web-based heading helper tool, manually with a custom dual Real Time Kinetic (RTK) GPS alignment tool, and automated with a collaborating Radar Cross Section (RCS) device. For RCS devices we use a marine radar reflector and attached RTK GPS when unable to fly and an unmanned aerial vehicle (UAV) also with RTK GPS when able to fly. By leveraging our experience working with UAVs and Radars we show a method to auto-calibrate the positioning sensors by using multi-sensor

fusion and collaborating participants, thus reducing the amount of setup time, and increasing the accuracy of the system.

Index Terms— Radar Heading, Alignment Calibration, UAS, RTK

ACKNOWLEDGEMENTS

I would like to extend my thanks to Mike Sanders, Executive Director of Lone Star UAS Center (LSUASC) of Excellence & Innovation for providing the time, resources and opportunity to develop this project.

I would like to express my appreciation to Dr Scott King, my research adviser, for his valuable and constructive suggestions during the planning and development of this research work. His willingness to give his time so generously has been very much appreciated.

I would like to extend my thanks to Mike Logan, Associate Director of Research at LSUASC via an Intergovernmental Personnel Act detail from NASA Langley Research Center for providing advice and suggestions during the planning and development of this research work.

I would also like to extend my thanks to the engineers at LSUASC, Eric Bird, Luis Hernandez and Riksy Nunez, for assembling the hardware necessary to support this project.

I would also like to extend my thanks to the software engineers at LSUASC, Angelo Diamante and Ryan Kelley, for their help in implementing and testing the systems, and collecting and processing the data.

Finally, I wish to thank my wife, Ratna, for her support and encouragement throughout my study.

DEDICATION

Lone Star Unmanned Aircraft Systems Center of Excellence & Innovation, Corpus Christi, TX.

TABLE OF CONTENTS

Page

ABSTRACT iv

ACKNOWLEDGEMENTS vi

DEDICATION vii

TABLE OF CONTENTS viii

LIST OF FIGURES x

LIST OF TABLES xii

CHAPTER I: INTRODUCTION 1

1.1 UAS vs UAV 3

1.2 Radar 3

1.3 RTK GPS 5

1.4 IMU 6

1.5 Goals and Contributions 6

CHAPTER II: REVIEW OF THE LITERATURE 8

CHAPTER III: SYSTEM DESIGN 12

3.1 Hardware 12

3.2 Software 20

CHAPTER IV: EXPERIMENTS 24

4.1 Alignment Techniques 25

4.1.1 Physical Alignment Tool 25

4.1.2 Compass Alignment 26

4.1.3 Digital Alignment Tool 27

4.1.4 IMU Alignment 27

4.1.5 Collaborating Participant Alignment 28

4.2 Data Collection 29

CHAPTER V: ANALYSIS OF RESULTS 38

Page

5.1 Methods 38
5.2 Alignment Results 41
5.2.1 Manual RTK Alignment 42
5.2.2 Compass Alignment - Android/iPhone 43
5.2.3 Digital Alignment Tool 44
5.2.4 IMU Alignment 44
5.2.5 Collaborating Participant Alignment - Auto RCS 45
5.3 Results 45
CHAPTER VI: SUMMARY AND CONCLUSIONS 48
NOTES 51
REFERENCES 52

LIST OF FIGURES

Page

3.1 The computer receives the position of the radar track and the position of the RCS device then calculates the new heading and sends it back to the radar platform. 13

3.2 Top down view of radar platform orientation setup showing IP addresses, angle offsets and naming designations. 13

3.3 Two Echodyne EchoGuard radars on a radar mast with the electronics box attached to the mast and two Emlid – RTK GNSS/GPS receivers (rovers) attached to the top to calibrate heading. 13

3.4 Two Echodyne EchoGuard radars on a radar mast with the electronics box attached to the mast and IMU/GPS box on top. 13

3.5 The inside of the Radar Platform electronics box contains power, networking and GPS hardware, and a small board computer for running software. 14

3.6 Adafruit BNO055 Absolute Orientation Sensor IMU and separate GPS in box on top of mast as in “Fig. 3.4”. 14

3.7 RCS device version 1. Radar reflector, RTK GPS, and battery on a PVC pipe. 15

3.8 RCS device version 2. Radar reflector, RTK GPS in weather proof box, Raspberry Pi, and battery on a metal pipe. 15

3.9 Software Overview. 20

4.1 The overall process for each experiment. 25

4.2 By using the radar, radar track, and UAV positions we calculate the radar heading offset. 29

4.3 The Goldilocks Zone for filtering potential radar track matches. 29

4.4 Situational awareness tool showing two radar platforms with two radars each tracking the radar reflector as it is driven by ATV down a runway. 29

4.5 Heading calculator web tool. 29

4.6 Digital Alignment Tool. A user can manually enter a heading from an external alignment technique or position the red dot to visually adjust the Radar Platform heading. . 30

4.7 Comparison of each method on the same scale. 32
4.8 Comparison of error in degrees with time to align for Manual RTK method. 33
4.9 Comparison of error in degrees with time to align for Manual Compass iPhone method. 33
4.10 Comparison of error in degrees with time to align for Manual Compass Android method. 34
4.11 Comparison of error in degrees with time to align for Manual Digital Tool method. . . 35
4.12 Comparison of error in degrees with time to align for Auto IMU method. 36
4.13 Comparison of error in degrees with time to align for Auto RCS method. 36
5.1 Time to complete alignment by method. 41
5.2 Difference between estimated truth data and observed check by alignment method. . . 42
5.3 Estimated accuracy based on the NSSDA standard which uses the RMSE formula. Accuracy reported at the 95% confidence level means that 95% of the positions in the dataset will have an error with respect to true ground position that is equal to or smaller than the reported accuracy value. In this graph, lower values are better. 46

LIST OF TABLES

Page

3.1 The radar platform components 16

3.2 The Computer components 16

3.3 The RCS device components 16

3.4 The software components 21

5.1 Estimated Comparison of Calibration Methods 47

CHAPTER I: INTRODUCTION

As of 2020, the Federal Aviation Administration (FAA) now projects that the commercial small UAS (sUAS) sector will have around 828,000 aircraft in 2024 [2]. Recreational sUAS aircraft are currently around 1.4 million and projected to be up to 1.48 million by 2024 [2]. In contrast, as of 2021, there are less than 250,000 commercial and military manned aircraft in the United States National Airspace System (NAS). The quantity of UAS weighing more than 55 lbs, larger UAS (lUAS), is steadily increasing as well. In 2019 the FAA counted 205 lUAS and projects that there will be 407 by 2024. The steady increase in the use of UAS in the NAS necessitates the use of technologies such as radar to enhance safety by detecting, tracking and identifying the aircraft. In some cases it is also necessary to mitigate malicious UAS by using counter UAS (cUAS) technologies that include Acoustic based UAS detection, Passive RF based Detection, Vision based UAS Detection, Radar based UAS Detection, and Data-fusion-based UAS Detection[28]. Generally, it is difficult to detect UAS that are slower and smaller than 1 m^3 due to the radar firmware default settings or custom track settings of observation software filtering out slow or stationary objects and the aircraft having a small Radar Cross Section (RCS) [22]. However, radar technologies are being developed that can accommodate the slower speed and smaller size of sUAS. In a Search and Rescue (SAR) scenario, it is important to be able to quickly set up equipment and quickly calibrate said equipment. The time it takes to set up and calibrate the system is time that could be spent on performing the actual mission. Time critical missions such as SAR accentuate the importance of a fast set up time because the longer a person goes without being rescued, the lower their life expectancy becomes. UAS can involve many components which increases the complexity of the system. Radar Platforms, GPS, Ground Control Stations (GCSs), Remote Pilots in Command (RPIC), Visual Observers (VO), infrared and optical sensors, communications devices, generators, etc can all be involved and all need to be set up to prior to starting a mission. Many of the components can be set up and calibrated ahead of time while others such as the Radar Platform must be calibrated at the beginning of each mission.

Lone Star UAS Center (LSUASC) of Excellence and Innovation employs a custom Radar Plat-

form for tracking Unmanned Aerial Vehicles (UAVs). The LSUASC radar platform is composed of multiple components such as one to four radar units attached to mounting hardware, power sources, networking systems, and electronics systems for collecting and storing data. LSUASC utilizes the Echodyne EchoGuard radar unit which detects and classifies small UAS in dense urban environments. LSUASC deploys one or more Radar Platforms during various missions and requires a setup and configuration process that is accurate and quick to implement. After several rounds of testing and operation in actual missions we have realized the importance of acquiring an accurate heading calibration. We have also identified the heading calibration as an area of improvement to setup time. More details about the radar platform are provided in 3.

One option for calibrating the heading of a ground based radar platform manually configure the heading prior to each use. Manual calibration can be time consuming and less accurate due to human factors. With manual calibration the heading can be estimated by using the GPS location of the radar platform, a map of the area and a point of reference. By using a software visualization tool to line up the radar with the point of reference we can acquire the approximate heading but it is only as accurate as our visual line of sight up to the point of reference in the real world. We can also use a compass to get an approximate heading. We can then make adjustments by visually matching radar tracks to the virtual map of the environment and eventually get a pretty good approximation of the correct heading. This process can happen fairly quickly if everything goes right and a little bit of luck is involved but it can also take upwards of 20 minutes to complete. This is why we propose an auto calibration method to save time and increase accuracy.

The aim of this paper is to reduce the setup time of a mobile radar platform and increase the accuracy of the system by providing a fast method of alignment calibration that is automated by using collaborating participants and multi-sensor fusion. With a mobile radar platform that is accurate and quick to set up, the UAS pilots can start their mission sooner with better situational awareness. In the following subsections of the introduction we will provide some details on the differences between UAS and UAV, what Radar and RTK GPS are and how they are used in relation to UAS, a description of the Lone Star UAS Center's mobile radar platform, details about Inertial Measure-

ment Units (IMUs) in relation to heading, and finally a summary of the goals and contributions for this paper. Later chapters include an overview of existing related technologies and literature, details about the hardware and software involved in the system design, the experiments that were designed and implemented for comparing the chosen heading calibration methods, details about each calibration method, the findings and results, and finally an overall summary and conclusion.

1.1 UAS vs UAV

Unmanned Aerial Systems (UAS) and Unmanned Aerial Vehicles (UAV) are two terms that are often used interchangeably. It is important to define the difference for clarity. A UAV or drone is more narrowly defined as the unmanned aerial vehicle only. A UAS is the entire system of Remote Pilot in Command (RPIC), Visual Observers (VOs), UAV, sensors, transmitters, receivers, GPS modules, software used for operating the UAV, Ground Control Stations (GCS), etc that are needed to make the UAV fly and perform its mission. The sensors can include visual cameras, sonic collision detection, infrared, LiDAR, and Radar to name a few. The GCS is a computer with software for controlling the UAV. The GCS software might include mission planning software, and software for interacting with a UAS Traffic Management (UTM) system via a Provider of Services to UAS (PSU) or UAS Service Supplier (USS). The GCS can also include software for situational awareness and data collection. The UAV can also be called a Remotely Piloted Aircraft (RPA). RPA is generally used when describing more complex UAVs that require in depth training to successfully operate due to the complex remote cockpit. In some cases, the remote cockpit of an RPA can be as complex as that of a commercial airliner. RPA and UAV can still be used interchangeably and both are a sub-component of UAS. The reader should be aware that these acronyms and initialisms are vast and can change or be replaced by others as the UAS industry continues to grow and evolve.

1.2 Radar

The word radar is an acronym that stands for RAdio Detection And Ranging. A radar detection system consists of an antenna that broadcasts and receives radio waves and a processor that is used to determine the features of the waves. The radar broadcasts or transmits radio waves that bounce off of objects and return to the receiver antenna. The radio waves that are captured by the receiver

represent the Radar Cross Section (RCS) of an object and can be processed to determine properties such as position, velocity, and even the type of object. Radar can also report on the heading of a tracked object [25]. RCS is a measure of how reflective an object is and is not affected by distance or strength of the transmitter. RCS signatures can be unique to an object or match well known types of objects which enables classification. Radar systems are capable of operating effectively in adverse weather such as fog, rain, and snow [5]. Radars are also capable of operating in low light conditions where visual detection methods would fail. Larger radar systems are able to track objects as far away as 400km. Some of the disadvantages of Radar include cost, time to lock on, distinguishing closely positioned targets, tracking slow targets, interference, and visual information. The cost of a radar system could be high when using it temporarily in a small area. If the radar is permanently used in a larger area then the overall cost is relatively lower. Radars can take up to 2 seconds to lock onto targets. When multiple targets are close together, radar can have difficulty distinguishing between the individual targets. In some cases, multiple targets might be viewed as a single larger target. Radar has difficultly tracking objects moving at slower speeds, for example, below 1 m/s. Cellular communication base station downlink transmission signals that are in the same channel as the radar are the primary cause of interference with air traffic control (ATC) radar towers [14]. Flocks of birds, trees on a windy day, cars on a nearby road or bridge, boat masts, and even ocean waves can cause false radar tracks and interference. Large buildings interfere with line of sight (LOS) by blocking the radar waves. Radars do not provide visual information such as the color of an object but they can be used to determine the shape of an object.

For a radar to track small Unmanned Aerial Vehicles (sUAVs), the threshold for the RCS of accepted tracks needs to be low. A lower RCS threshold increases the sensitivity of a radar system which attributes to an increase in false radar tracks. Flying at lower altitudes causes difficulty with distinguishing UAVs from ground clutter such as cars, swaying trees, and waves in water bodies. Interference from overlapping radar tracking azimuths can also contribute to the problem of false tracks. Manual sensor calibration is a tedious and expensive process [23]. If the radar system is not calibrated correctly then other potential problems may be compounded. In multi-sensor systems,

incorrect calibration can exacerbate errors and cause safety concerns.

Radars can be used to Detect and Avoid (DAA) non-conforming rogue aircraft. A rogue aircraft is one that is not following its prescribed flight plan or that did not submit a flight plan to the local UAS Traffic Management (UTM) system. Rogue aircraft pose a safety hazard and need to be accurately detected and avoided especially in beyond visual line of sight (BVLOS[1]) scenarios. In every UAS scenario it is critical to the safety of the vehicle and the operator to have good situational awareness. An accurate radar system can provide situational awareness to critical personnel that are involved with a mission.

1.3 RTK GPS

Real-time kinematic (RTK) Global Positioning Satellite (GPS) is a satellite navigation and positioning technique that uses a base station and 1 or more smaller GPS rover modules that link up with the base station. The system is able to auto correct itself to the point of gaining centimeter position accuracy [12][27]. The base station uses a satellite signal's carrier phase wave instead of the data within to calculate a margin of error then transmits the error to each rover. The base station effectively has an exact known position that is fixed and based on the signals from multiple satellites. The carrier phase range between a satellite and the base station receiver is measured to determine the position of the base station with high accuracy. The base station calculates errors and sends the corrections to the rover GPS. Each rover then uses that margin of error to gain a more accurate position than otherwise possible if it were acting alone. This is another example of sensor-fusion with collaborating participants enabling increased accuracy of the overall system. Disadvantages to using RTK include the cost, the required base station, clear LOS to rover, limited radio range, data link drop outs, and speed of rover movement. An RTK system can cost several thousand dollars depending on the brand and how many rovers are required. When setting up the base station, finding a good location that provides a view to wherever the rovers will be is important. The base station can occasionally take several minutes to gain a fixed position thus delaying the mission. The rovers operate best with a clear LOS to the base station which is not ideal in BVLOS scenarios. The long range (LoRa) radios used to communicate between the base station and rover have a limited

range of up to 8km in the case of the EMLID but can be more depending on the RTK device and environment. Data packets can be lost or dropped in some cases causing gaps in data. If a UAV is moving too fast then the onboard rover could have trouble gaining a fixed centimeter accurate reading causing variations in the accuracy of the reported data.

1.4 IMU

An Inertial Measurement Unit (IMU) that uses an accelerometer(s), gyroscope(s), and magnetometer(s) is considered a six degrees of freedom (6Dof) sensor. It is 6Dof because it provides surge (translation forward or backward), heave (translation up or down), and sway (translation left or right) in addition to the pitch (rotation around the transverse axis), roll (rotation around the longitudinal axis), and yaw (rotation around the normal axis) of 3Dof sensors. It is common for an IMU to contain one accelerometer, gyro, and magnetometer per axis of roll, pitch, and yaw. A factory calibrated IMU will have an initial calibration that is unique to the specific sensor. A factory calibrated IMU still needs to be calibrated before each use but it should calibrate faster and have lower estimation drift rate. The accuracy of the magnetometers can fluctuate depending on the planetary magnetic field and ferromagnetic materials in the vicinity of the sensor. With IMUs, the sensor needs to be calibrated for accurate yaw (heading) and generally it is manually calibrated prior to or during each use by physically moving the sensor in a prescribed pattern along each axis in 3D space. The IMU on a UAV also needs to be calibrated in the same way. The IMU on a UAV generally re-calibrates itself as the UAV moves through the environment but it can accumulate errors as a build up of estimation drift as time goes on. Self calibrating an IMU is a challenging task due to the accumulation of errors over time [9] [8]. A stationary IMU, such as on a ground based radar platform, does not continuously move so estimation drift errors are minimized. A stationary IMU cannot internally update itself which means that it depends on a good and accurate initial calibration.

1.5 Goals and Contributions

The Lone Star UAS Center (LSUASC) of Excellence and Innovation requires an accurate radar alignment solution that has a fast setup time. To achieve that goal, we have created software and hardware systems to integrate related positional data into a custom web based situational awareness

application. We also store all of the positional data in a custom relational database. We implement multiple radar alignment methods for comparison to determine the most accurate, timely, and cost effective solution. We compare the accuracy of each alignment method by using the Geospatial Positioning Accuracy Standard provided by the Federal Geographic Data Committee (FGDC). We collect and compare the errors of each alignment method using the FGDC's National Standard for Spatial Data Accuracy (NSSDA). We record the time to complete each alignment method as another comparison. The alignment methods that we will compare include manually with a hand-held compass, manually with a web-based heading helper tool, manually with a custom dual RTK GPS alignment tool, and automated with a collaborating RCS device. The proposed method is the antenna heading auto-calibration with collaborating RCS participant that relies on RTK GPS and multi-sensor fusion. Automating the heading alignment by using an RCS device, RTK GPS, and radar has the potential to be highly accurate and fast.

CHAPTER II: REVIEW OF THE LITERATURE

The necessity of acquiring an accurate antenna alignment for a radar system is well known in the world of radar usage. Radar is used in many systems to detect information about the surrounding environment including cars, UAVs, boats, airplanes, and monitoring experiments to name a few. Each of these systems can utilize multiple sensors that combine data to provide information about the environment. Each of the sensors needs to be properly calibrated to be effective in providing information about the local environment. LSUASC uses multiple technologies to implement a radar system for tracking UAVs and storing the related data. In the case of radar alignment, we test multiple methods that use multiple different technologies that are similar but not directly related to previous research. We discuss some of the related research below which includes sensor fusion, radar alignment, radars, RTK GPS, and time critical events.

The effective alignment (either spatially, geometrically or temporally) of multiple heterogeneous sensor streams, and utilization of the diversity offered by multimodal sensing is referred to as sensor data fusion [24][15]. Data fusion can involve combining data from a set of similar sensors, a set of unique sensors, or a combination of both similar and unique sensors. Sensor data fusion not only relates to radar tracking of UAVs but also to autonomous vehicles, surveillance, smart guiding glasses, and hand gesture recognition [24]. For example, the latest cars fuse proximity information from visual cameras, small wavelength radars, lidar, and ultrasound sensors. By fusing multiple sensors an enhanced perception of the surrounding area is available for making informed decisions to help with things like parking assistance and collision avoidance.

Advanced Driver Assistance Systems (ADAS) assist drivers with parking and driving transportation vehicles. The accuracy of an ADAS depends on sensor mounting position and alignment of different odometry sensors such as gyroscopes or wheel-based odometry and radar systems [13]. For example, a radar alignment error of 1° induces a lateral offset of 1.75m of observed objects in a distance of 100m. This error is enough to make an oncoming car in the distance appear to be in a different lane than it actually is thus degrading any internal mapping of the environment and causing safety concerns. The same applies when tracking UAVs via radar for the purposes of SA

and DAA. If the radar is misaligned then the mission operations manager will not have an accurate view of the mission. If the radar is misaligned then the DAA software could project that a collision will not occur when in fact it will occur. A misaligned radar will have larger and larger errors the further out the target is.

Intelligent Transportation Systems (ITS) assist with traffic management [23]. Manual calibration of sensors is tedious and expensive. Fusing cameras and radars allows auto-calibration without explicit calibration targets. Auto calibration of multiple redundant sensors is crucial to the overall accuracy of a system [23]. It can be challenging to combine multiple sensors due to the complex associations between the data. Normally, when calibrating a camera system, a calibration pattern is used [11]. In 1992, a method was developed for auto-calibrating cameras based on their rotational movement [17]. This method requires 2d feature matches between multiple views and does not require a predefined calibration pattern. Multi-sensor fusion is a bit more complex. When aligning cameras and radars within the same system it becomes apparent that radars do not provide edges while cameras do. Also, cameras are not as good at providing accurate position and velocity as compared to radars. By fusing cameras and radars it is possible to auto-calibrate a system without the use of explicit calibration targets [23]. It is also possible to reduce the time to complete a beam alignment by using an off the shelf IEEE 802.11ad WiFi router [19]. The WiFi router is configured as a base station and is used to send the radar network synchronization data without modifying the existing 802.11 network protocols. By combining radar and synchronization and continuously calibrating a radar based Intelligent Transportation System it is possible to reduce the beam training time.

Radar is more effective than visual sensors for navigating in degraded environments [10]. Degraded environments include low light (dark), dust, fog, smoke, GPS-Denied, or any other type of sensor obscurants [21]. Radar is able to penetrate most degraded environments to assist a vehicle with navigation. Simultaneous Localization and Mapping (SLAM) systems utilize radar to be effective in visually degraded environments. The radar in a SLAM system is aligned between the detected features of two frames by estimating the rotation with the eigenvectors of the point clouds

[10]. The calculated relative transformation is added to the internal pose graph and assists with loop closure. By using the rotational translation detection a moving UAV can use radar SLAM to recognize that it has been somewhere before and close the loop and update its internal pose representation of the world around it. Loop closing refers to determining if a vehicle has previously visited a location and using that information to make adjustments to the internal pose of the vehicle.

High frequency radar is used to map coastal surface ocean currents [4]. Misaligned radar antennas can cause fluctuations in the resulting data. The classical radar alignment technique when mapping surface ocean currents uses the transponder on a boat. Using a transponder on a boat is more costly and error prone. It has been shown that a bistatic auto-calibration technique can be used in place of a transponder on a boat [4]. A bistatic solution involves using the signals from other radar stations to self-calibrate the system. In a bistatic configuration the transmitter and receiver are located in different areas. This is in contrast to a monostatic configuration where the transmitter and receiver are co-located near each other. This radar autocalibration method increases accuracy and lowers cost of mapping surface ocean currents.

The centimeter accuracy of Real Time Kinetic (RTK) GPS makes it ideal for fine tuned analysis. RTK has been used to research the gait of humans[27], and explore the efficacy of automated cinematography [12]. When analyzing the gait of humans the Walking Speed, Step Length and Step Frequency data can be collected by using RTK rovers positioned at various points on the human. Using RTK allows researchers to circumvent the shortcomings of a small laboratory where the test subject might be walking on a treadmill to save space. With RTK the human gait data can more easily include multiple terrain types and longer distances. When combined with other methods such as the use of accelerometers, gyroscopes and video cameras the sensor fusion can create an enhanced analysis capability. Automated cinematography with RTK involves using UAVs with cameras or other automated camera devices that are programmed to track RTK devices. The UAV orients the camera toward the RTK device while remaining stationary or flying a prescribed pattern such as a pleasing orbit or fly by. With the addition of other sensors such as an IMU on each of the subjects the pose of each subject can be known. This means that the camera can know which direction a

subject is gazing in and prioritize camera angles from the appropriate side of the scene.

In a search and rescue scenario, getting the first responders equipment up and running as fast as possible can mean the difference between life and death. Search time is the best predictor for survival meaning that the longer a person is not found the lower their chances for survival [1]. There are many different SAR environments and the average survival time varies depending on the environment. For example, victims of urban earthquakes have a median survival of 5.75 days [16] while people lost in a forest survive about 2.8 days on average [1]. There are 7 factors that affect Urban Search and Rescue (USAR) scenarios. The 7 factors include best practices and lessons learned, rescue technology, community involvement, information systems, technologies integration, crisis management, and available budget [26]. Rescue technology includes enhanced casualty assessment, monitoring and extrication tools. UAVs can help with casualty assessment of hard to reach victims, monitoring the area and searching for victims and dangers via infrared and visual sensors, and assisting with extrication planning. The first responders are using UAVs to assist in search and rescue or disaster analysis then a radar system that is used for situational awareness also needs to be set up and calibrated as quickly as possible.

CHAPTER III: SYSTEM DESIGN

As mentioned, one bottleneck with setting up mobile radar systems, UASs, and technology in general is the time it takes for calibration of the sensors. The accuracy of the sensors depends on proper calibration. The architecture involved in a system that uses radar, UASs, data collection, DAA, etc can have large amounts of moving parts which leads to increased potential points of failure. Since there can be many points of failure in a complex system, it is important for each part of the system to be calibrated and working optimally. The architecture of our system also includes many different hardware and software components working together to capture and store telemetry data and display that data to a user.

A summary of the system that we will use for the alignment experiments involves many hardware components including Echodyne EchoGuard radars and Emlid – RTK GNSS/GPS receivers with centimeter accuracy. We also use the Adafruit BNO055 Absolute Orientation Sensor IMU for the IMU alignment tests. We have implemented a web API server that is used for collecting data, a database server that is used for storing data, and a web server that is used for hosting a web application for viewing the positional data in real-time. The system also includes multiple software components to tie everything together including software to collect and transmit radar tracks and RTK data, software to calculate the alignment error, web API software for accessing the database and software for visualizing each of the components and data points in real-time. The operations that utilize this system are limited to occurring in GPS-Enabled environments. As such, this system does not address operations in GPS-Denied[2] environments such as indoors, underground caverns or extraterrestrial bodies such as asteroids or the moon. In the following sections we go into more detail for each of the hardware components and then the software components.

3.1 Hardware

The hardware can be broken down into four high level components as shown in “Fig. 3.1”; The Radar Platform, a computer, a Radar Cross Section (RCS) device, and the network that connects everything together. The LSUASC Radar Platform consists of the required hardware for tracking

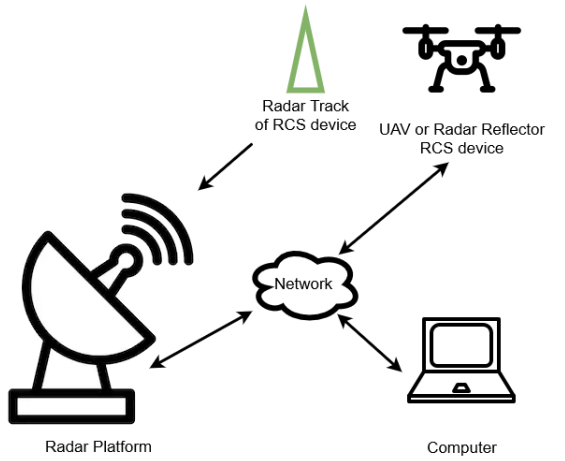

Figure 3.1: The computer receives the position of the radar track and the position of the RCS device then calculates the new heading and sends it back to the radar platform.

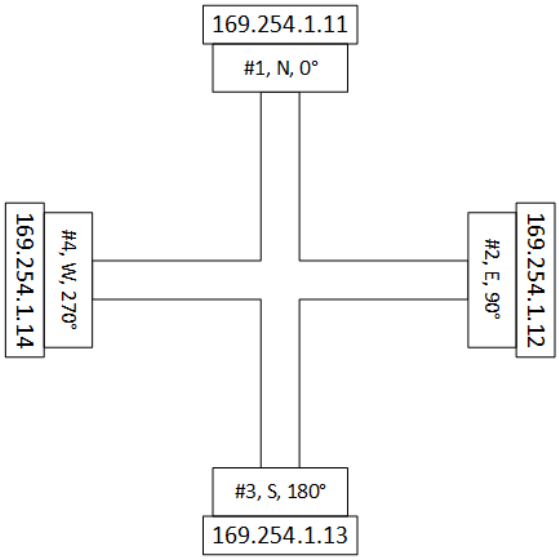

Figure 3.2: Top down view of radar platform orientation setup showing IP addresses, angle offsets and naming designations.

System Overview.

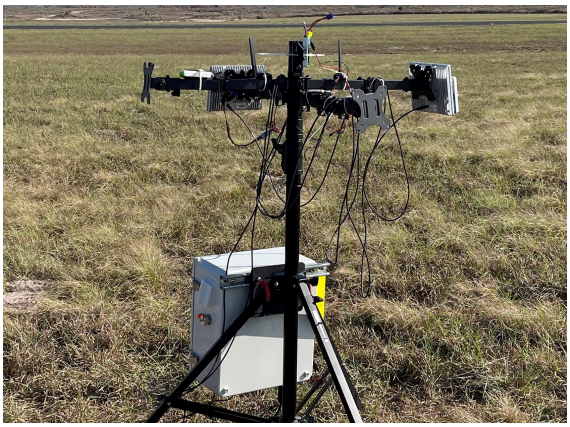

Figure 3.3: Two Echodyne EchoGuard radars on a radar mast with the electronics box attached to the mast and two Emlid – RTK GNSS/GPS receivers (rovers) attached to the top to calibrate heading.

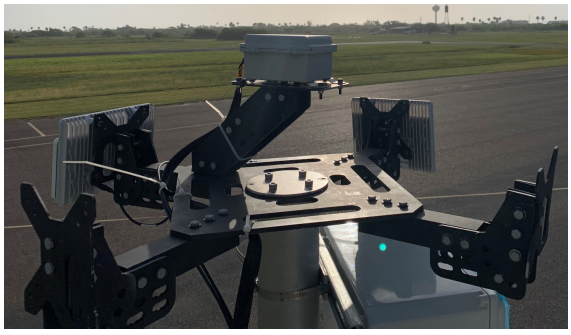

Figure 3.4: Two Echodyne EchoGuard radars on a radar mast with the electronics box attached to the mast and IMU/GPS box on top.

Radar Platforms in various configurations.

objects and sending the track data to the network. The computer represents one or more computers that encompass software for interacting with the various parts of the system. For example, the computer can connect to the radar platform via the network and the computer can connect to the RCS device via the network. The RCS device can either be a radar reflector on a pole carried by a human or a UAV; both with attached position and networking hardware. The RCS device is also

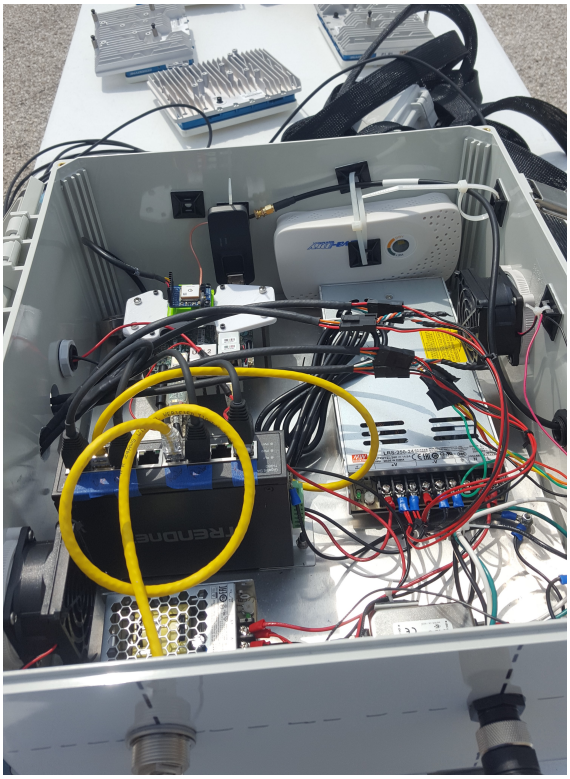

Figure 3.5: The inside of the Radar Platform electronics box contains power, networking and GPS hardware, and a small board computer for running software.

Figure 3.6: Adafruit BNO055 Absolute Orientation Sensor IMU and separate GPS in box on top of mast as in "Fig. 3.4".

Hardware Components.

a collaborating participant because data is shared directly with the networked computer(s) via our web API. In contrast a non-collaborating participant would be a device that we do not have direct control over. Next we go into more detail for each of the four high level hardware components of the system.

An individual Radar Platform consists of 1-4 radar units mounted on a mast to raise the radars several meters above ground level and provide up to 360-degree coverage. See table "Table. 3.1" for a list of radar platform components. The individual radar units are named R1, R2, R3, and R4 (see "Fig. 3.2"). The radar platform has gone through several iterations that include a wooden frame, truck mounted telescoping mast, and the current metal mast (see "Fig. 3.3" and "Fig. 3.4" for the most recent versions). The individual radar units on the radar platform are separated by 90 degrees such that full 360 degree horizontal coverage is possible. The radar platform has a single heading reference, R1, that each of the other mounted radars is offset from. For example, if the

Figure 3.7: RCS device version 1. Radar reflector, RTK GPS, and battery on a PVC pipe.

Figure 3.8: RCS device version 2. Radar reflector, RTK GPS in weather proof box, Raspberry Pi, and battery on a metal pipe.

RCS device collaborating participant versions.

overall platform heading is at 0° with respect to north then the offsets for R1, R2, R3 and R4 are as follows: 0, 90, 180, 270 degrees respectively. If the platform heading is 169° then the offsets for R1, R2, R3 and R4 are as follows 0+169, 90+169, 180+169, 270+169 degrees respectively. The offsets are clamped between 0 and 360 degrees by wrapping around or subtracting 360 until it is less than 360. The individual radar units include internal IMUs that measure pitch and roll using accelerometers, gyroscopes and magnetometers. For the radar platform we will rely on the internal IMU of each radar unit for pitch and roll of each individual radar. This means that when converting radar tracks from radar space to world space coordinates we take into account the orientation of the radar with heading as the offset from R1, and pitch and roll from the IMU within the radar unit

№	Radar Platform
1.	1-4 radars in the configuration shown in “Fig. 3.2”
2.	Each radar contains an IMU for local pitch and roll. Yaw ignored.
3.	Upboard Squared small board computer for processing radar tracks.
4.	RTK GPS for positioning.
5.	Wireless LTE dongle for remote communication.
6.	Access to elevation API for accurate radar altitude.
7.	Mast and mounting bracket for structure.
8.	Weather proof box for electronics and networking hardware.

Table 3.1: The radar platform components

№	Computer
1.	Custom software to operate and monitor radar platform.
2.	Custom software to monitor data collection and visualization.
3.	Human operator.

Table 3.2: The Computer components

№	RCS device
1.	RTK GPS for positioning.
2.	Raspberry Pi for interfacing RTK data to the API.
3.	Wireless dongle for network communication.
4.	Human operator for navigation and control.

Table 3.3: The RCS device components

that reported the track. There is a version of the radar platform mast that has a top mounted box containing an IMU for the IMU experiments (see “Fig. 3.4” and “Fig. 3.6”). The top mounted IMU can be used to mostly automatically provide a magnetic north alignment. After a person calibrates the IMU the heading can be read and applied to the radar configuration in software. We will not use an IMU for the heading of the radar platform for the final solution but we will collect IMU alignment data for comparison with the proposed solution. The latest version of the Radar Platform mast

does not include a top mounted IMU (see “Fig. 3.3”). The Radar Platform includes an electronics box (see “Fig. 3.5” that contains networking hardware such as a wireless LTE dongle for sending and receiving data across the network, and a switch for interfacing between the radars and small board computer (SBC). The SBC is an UPboard Squared running Ubilinux operating system and our custom software. The SBC is used for capturing the radar track positions, converting track positions from radar space to real word positions using the WGS84 datum[3], and sending them to LSUASC’s internal systems over the wireless cellular network for live visualization and storage. The Wireless LTE dongle is directly connected to the SBC. The radars are connected to the switch so that the SBC can read their data via UDP sockets. The data is automatically uploaded to a database via wireless LTE dongle and web API. There is also a GPS module inside the box that is connected to the SBC and normally used for position. However, for these experiments we are not using this GPS. Instead we use an external RTK GPS for higher accuracy. There are two power converters as some of the hardware components require 5v while others require 12v. The power converters split the same power source, from a wall outlet when testing or a generator in the field, into either 5v or 12v and the related components are wired to the appropriate power converter. The electronics box also has several intake and exhaust fans to provide airflow for cooling. There is also a dehumidifier to try and reduce moisture. Our test environment is hot and humid so the fans and dehumidifier were added as mitigation but they did not seem to have a large impact. On the outside of the box there are ports for a power cable, the radar data and power lines, a fuse box to protect against power surges, and a red light to indicate when the box is powered. There are also mounting hooks on the outside of the electronics box for hanging it off of the ground since we occasionally set it up in tall grass.

In the diagram the computer represents all of the computers and servers involved in the system except those in the other high level components such as the SBCs in the radar platform and RCS device. See table “Table. 3.2” for a list of computer components. The computers include laptops for use in development and in the field, servers that host a web API for collecting data, a database for storing data, and a web server for hosting a website. The website has multiple tools for things

like Situational Awareness (SA), configuring the radar platform, calculating heading giving GPS points, file conversion tools, etc. The computer is used to configure the radar platform software and RCS device software. For the radar we use a web based radar configurator tool. For the RCS device we physically connect via Ethernet cable and SSH into the RaspberryPi. The computer is also used to access a web based custom visualization tool to view the position of each sensor in a three dimensional virtual environment in real time. The visualization tool is our web based situational awareness (SA) application that runs on a separate server. The computer is also able to view the data in the database during operations by remotely connecting to the database server. The computer is also able to monitor the web API server and messaging server. More details about the software used in each component will be given in section 3.2.

The preferred collaborating participant RCS device is a UAV with RTK GPS and a human operator that initiates autonomous flight of the UAV in front of the radar platform. See table "Table. 3.3" for a list of RCS device components. For example, after the radar platform has been set up, the operator would ideally be able to launch the UAV with the press of a button that would put the radar platform system into calibration mode, autonomously fly a pattern in front of the radar then return to the starting location. As an alternative to an autonomous UAV, we have constructed an RCS device that we like to call "The Mace of Reflection" as its shape resembles a large mace. The Mace of Reflection RCS device consists of a marine radar reflector attached to a 10ft piece of aluminum conduit. The aluminum conduit has been cut to have 4 slots near the top so that the radar reflector can slide into them. The radar reflector is secured with a conduit cap and hose clamp. The marine reflector is a typical radar reflector purchased from a local boat shop. Underneath the marine reflector we attach our battery powered RTK GPS rover. The RTK GPS rover communicates with the RTK base station via LoRa to receive positional corrections. Also, below the rover is a battery powered Raspberry Pi with attached wireless LTE dongle. The RaspberryPi is directly wired to the RTK Rover. The RCS device uses RTK GPS for positional information and reports the position data to Raspberry Pi. The Rasberry Pi collects the position data from the a serial port interface on the RTK then forwards it to the LSUASC network via the wireless LTE dongle and LSUASC web

API. The LSUASC web API is the gateway to the LSUASC database where the data is stored. The first version of the RCS device was created with a PVC pipe and did not have the ability to send RTK data in realtime to the radar box (see “Fig. 3.7”). The version one proof of concept required manual collection of the RTK data using the provided mobile app. The latest version of the RCS device has the attached Raspberry Pi and Wireless LTE dongle and is able to send the RTK data to the Lone Star network for processing (see “Fig. 3.8”). The conduit also has 2 small holes drilled though it for connecting the RTK box with zip ties. The other components were connected with duct tape but additional holes can be drilled in the future if necessary.

The network includes networking hardware but it represents all of the communications connections between all of the servers, computers, SBCs, and applications. For example, the computer uses the network to connect to the radar platform, the database, or the RCS device. This system has the capability of cross communication between the radar platform, RCS device, and a ground control station (GCS) computer via the LSUASC network. Some of the communications involve web API REST requests while others involve passing protobuf encoded messages to a messaging server. The messaging server then broadcasts the messages to all topic listeners. For example, the radar platform software sends data to the web API for DB storage and also sends telemetry to the messaging server. The SA web application client for each user listens to the messaging server for telemetry then displays it to the user.

For the hardware systems used in this research project, the radar platform already existed, the network infrastructure already existed, and the computer existed. The radar platform electronics device was custom built and is maintained by the LSUASC engineering team and it represents a unique design. The networking infrastructure is built and maintained by the Texas A&M IT department. The RCS device was custom made from scratch for this project and represents a unique design. The RCS device was built by a collaboration between the LSUASC engineering and software teams.

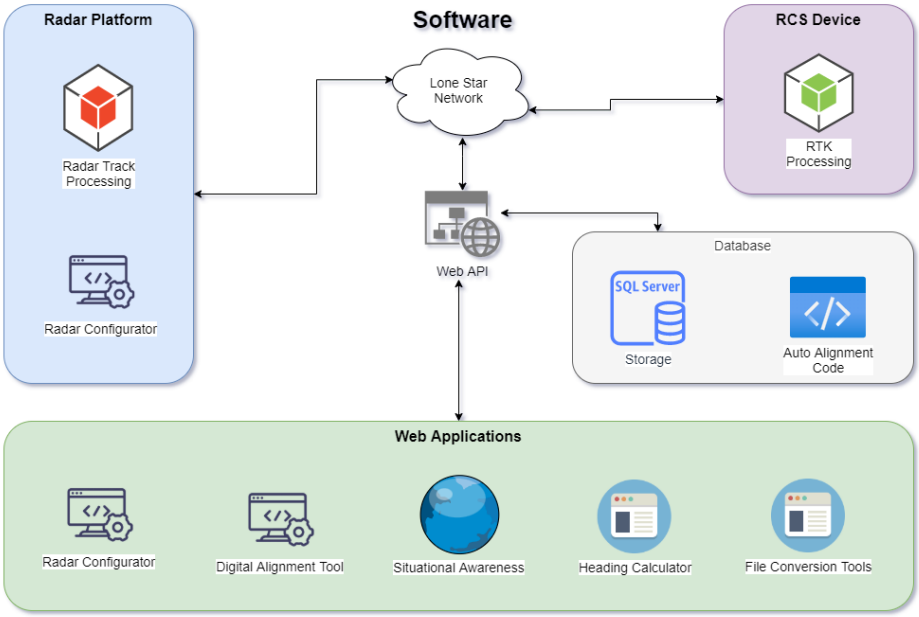

Figure 3.9: Software Overview.

3.2 Software

The software for this system includes multiple applications for collecting, storing, visualizing and processing data (see "Fig. 3.9"). An SQL relational database stores all of the data that we collect and provides the ability to run SQL queries for analysis of the data. Some data is processes in real time while other data is processed after the data collection is completed. A web API is used as the entry point to the database and provides each of the other software components with the ability to store data such as logs for troubleshooting. For example, the Radar Platform and RCS Device both send data to the web api for storage. Message oriented middleware software is used for passing radar and RTK telemetry directly to the LSUASC SA application for real-time data visualization. The messages that are sent are trimmed down and strictly what is necessary for visualization. Being able to see the positions of the radar and RTK telemetry in real-time allows us to easily verify that things are working as expected. A summary list of the software components can be seen in "Table. 3.4"

The software for collecting radar track data runs on a small board computer in the radar electronics box. The application is written in Java and listens to the radars on multiple sockets for various types of data such as tracks, status, and configuration. The radar data collection software

№	Software
1.	A custom database for storing the data.
2.	A custom web API for collecting the data.
3.	A custom program for collecting and forwarding the radar track data to the web API.
4.	A messaging server for routing messages between various applications.
5.	A custom program for collecting and forwarding the RTK data to the web API.
6.	A custom stored procedure for calculating the Auto RCS alignment.
7.	A web based situational awareness application.
8.	A web based radar configurator tool.
9.	A web based digital alignment tool.
10.	A web based heading calculator.
11.	A web based file conversion tool that supports multiple conversion methods.

Table 3.4: The software components

converts the raw radar track data from radar space into real world latitude, longitude and altitude coordinates by applying matrix transformations to the track positions in relation to the GPS location of the radar unit and its altitude reference[4]. The radar data collection software then sends the raw track data and the calculated real world positions to the LSUASC web API for database storage and also to the messaging server for real time viewing in the Situational Awareness (SA) application. The application also collects the local GPS position to send with the various data streams. The radar platform also runs a radar configurator application that is paired with another component on our web server and is written in NodeJs. The radar configurator is used to send information about the radar platform and receive and apply configuration changes such as the alignment value. The radar configurator is also used to start and stop the radar platform software from tracking. The radar platform software is custom made but it existed prior to this project. For this research we modified the software to include two auto alignment modes, one for IMU alignment and one for RCS alignment. The IMU that we used reports calibration levels from 1 to 4 with 1 being the best and 4 being the worst. When the radar platform is set to Auto IMU alignment mode the software checks for the calibration level to be below 2 to complete the alignment. When the radar platform

is in Auto RCS alignment mode it periodically calls a web API that runs a stored procedure (SP) in the database. The parameters for the SP include the radar platform UUID, a timestamp, and the current radar platform heading. The SP queries the database to match up radar tracks and RTK data then calculates the heading alignment offset. The SP only looks at radar tracks that were previously stored by the specific calling radar platform. The SP only looks at RTK tracks that are within a certain threshold of distance and time to the radar tracks. The algorithm for matching the radar and RTK tracks is naive and only filters tracks based on a calculated goldilocks zone around the radar platform and then the closest distance between the radar track and RTK track. The goldilocks zone is described in more detail in 4.2. The SP only returns a new heading for the radar platform if the result is not the same as the passed in alignment and the difference between the new and old alignments is greater than 1°. The algorithm considers the alignment stabilized when corrections are less than 1°.

We developed a custom digital alignment tool that runs in a web page. The tool is currently built into the web application side of the radar configurator tool. The tool displays the radar platform location and a red dot that can be moved around by the user. The tool is used to visually set the radar alignment by moving the red dot around until it aligns with the landmark of choice. While moving the red dot, the user will see that the heading alignment value changes. In order to find the correct estimate of where to put the red dot, a person must physically view the radar platform from a distance of at least 10ft and visually align the radar platform with a landmark in the distance. For example, a person visually lines the radar platform up with the corner of a distant building, then moves the red dot to the satellite image of the buildings corner. Once complete, the user saves the radar configuration to apply the update. This tool is currently optional such that the heading alignment can be manually adjusted by typing in a value. For the manual alignment methods the text box that is updated by the digital alignment tool is the same text box in the radar configurator that is used for applying the alignment change.

We also developed custom software for this project that is used for collecting and processing the RTK data. The RTK data collection program runs on a Raspberry PI that is physically connected to

the RTK rover and a wireless LTE dongle and is written in NodeJs. The software works by listening to the serial interface of the RTK rover and forwarding the RTK position data to the LSUASC web API. The RTK data is stored in the database for later review. Software was developed for this project to handle data analysis by comparing the radar track data with the RTK data.

We also created several file conversion tools for various conversions such as LLH[5] to CSV, Radar to CSV, and CSV to KML[6] to assist with data analysis and visualization. In some cases we converted the raw radar and RTK data into CSV format prior to loading it into custom parsing applications for generating graphs such as those in this document. In other cases we converted radar and RTK data into KML files to be able view the entire scenario in Google Earth. When generating the KML files we reduced the number of points and labeled them with their timestamps and colored them by data data type.

There are many software components involved in the overall system. Some of the components such as the radar configurator are used in some shape or form during all alignment methods. The radar configurator is used to set the initial alignment configuration, which radars should track, and start and stop radar tracking. Other software components such as the digital alignment tool, heading calculator, Auto IMU mode, and Auto RCS mode and only used during a specific alignment method. The web API and database were updated to support data collection for each of the alignment methods. The majority of the API endpoints are POST calls for inserting data into database tables. The one API call that is a GET is for the Auto RCS proposed method and, as previously mentioned, it calls the custom SP for calculating a new heading based on radar track and RTK GPS data. Finally, our situational awareness (SA) tool previously existed and required minor changes. We added the ability to visualize RTK GPS track data to the SA tool for this project. The SA application, digital alignment tool, heading calibration tool, file conversion tools, and Web API are all written in C ASP.Net Core. The SA application uses several third party technologies including CesiumJS, Vue.js, jQuery, and Bootstrap. In the next chapter we will discuss the process of each experiment and give more insight into the usage of some of the previously described software.

CHAPTER IV: EXPERIMENTS

The experiments involve performing several alignment techniques, collecting the resulting data and comparing the results with truth values. The physical alignment method uses ADS-B[7] as the truth position and compares with radar tracks. However, the other methods use an alignment with the physical method as the truth alignment. The data includes positions of all involved devices, time to align and accuracy of alignment. The overall process for each experiment can be broken down into five main parts which are "Initial Setup", "Start Timing", "Perform Alignment", "Update Config", and "Stop Timing" (see "Fig. 4.1"). The Initial Setup of each alignment method varies between methods but all methods include setting up the radar platform. Setting up the radar platform involves physically setting up the mast, attaching the radars to the mast, connecting power and data cables from each radar to the electronics box, physically aligning it in the general desired direction, and connecting power to the electronics box. We do not include the "Initial Setup" in the time to align. We use the physical alignment method (two RTK GPS) as the estimated truth value for all but the physical alignment method. For all methods other than the physical and digital alignment methods, the final step of the Initial Setup is to collect and record the alignment truth based on the physical alignment tool method. For the digital alignment method we get the truth after we "Stop Timing" since it involves physically adjusting the alignment. Once the Initial Setup is complete and we are ready to start the selected alignment method we "Start Timing" the alignment method. Then we proceed to the "Perform Alignment" stage which is different for each alignment method and discussed in detail below. Once the alignment is completed we then perform the "Update Config" stage signalling that it is time to "Stop Timing" the process. Updating the radar configuration with the alignment value attained from the alignment method is handled differently between the auto and manual methods. For the manual alignment methods, the user manually types the alignment value into the radar configurator tool then clicks save. Once the user clicks save, we "Stop Timing." For the auto alignment methods, the user waits for each method to stabilize and then we "Stop Timing." For the auto alignment methods, stabilizing means that the alignment value was selected and applied to the configuration. For the Auto IMU method stabilized means that the IMU gained a calibration

value greater than or equal to two. For the Auto RCS method, stabilized means that the SP is no longer sending updates.

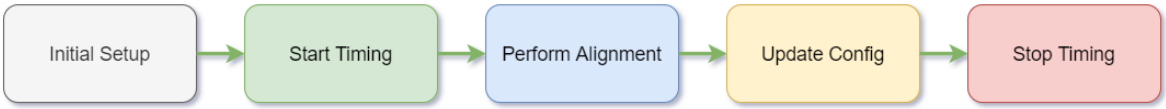

Figure 4.1: The overall process for each experiment.

4.1 Alignment Techniques

The alignment techniques that we implement, test and compare include "Physical Alignment Tool", "Compass Alignment", "Digital Alignment Tool", "IMU Alignment", and "Collaborating Participant Alignment". For the "Physical Alignment Tool", we compare the locations of radar tracks with ADS-B telemetry to determine an estimate of the accuracy. For the other alignment techniques, we will use the estimated alignment of the "Physical Alignment Tool" (taken without radar tracks and ADS-B) as the truth value to determine how accurate the other alignment techniques are. We discuss each alignment method in detail.

4.1.1 Physical Alignment Tool

We created a custom Physical Alignment Tool that mimics a commercially available tool similar to commercially available antenna alignment tools. The commercially available tools might perform better than our custom tool but we already had the components to create a custom version. The tool consists of two mounting brackets on the radar platform mast for attaching two RTK base stations running in rover mode. There is also another RTK base station running in base station mode for providing corrections to the base stations running in rover mode. We manually collect the latitude and longitude readings from each RTK rover and enter them into a custom web tool to calculate the heading of the radar platform. The custom web tool works on mobile devices for ease of use in the field and stores the results of the alignment in our database. The tool also stores the start and end time of the alignment and other meta data to assist with linking it to the collected radar tracks. Once the custom web tool calculates the heading, we manually enter it into the platform configuration by using the heading helper in our radar configurator tool. It is possible to have the

tool apply the new heading directly to the radar platform and that could be a feature that we add in the future. The physical RTK alignment tool is highly accurate due to the centimeter accuracy of RTK GPS. However, there is still the potential for some error due to the nature of GPS and imperfections in the radar platform mast. The alignment accuracy can be affected by human error such as incorrectly entering the latitude and longitude readings into the heading helper tool. The accuracy is also affected by whether or not the RTK GPS modules acquire a fixed (centimeter) vs float (meter) solution. Due to the possible errors we say that this truth value is an estimate of the actual truth. In order to measure the accuracy of this method we tracked the radar signatures of UAVs in flight and compared them with the ADS-B telemetry. The time to perform this method includes attaching the two RTK rovers to the radar mast, entering the latitude and longitude of each RTK rover into the calculator, and entering the resulting alignment into the radar configuration. This solution works fairly well, is relatively easy to implement, and is more accurate than our tests with an IMU and compass. The main issues with this solution are that it takes longer to set up and the RTK calibration occasionally takes a while to gain a fixed resolution.

4.1.2 Compass Alignment

A simple approach to aligning a radar antenna is to manually use a compass and visually acquire the heading of the antenna. Using this method will achieve an approximate result that can be adjusted over time by viewing live tracks and manually adjusting the heading so that the tracks align with landmarks in a situational awareness (SA) program. A margin of error is expected and is compounded the further away the track is from the radar. In order to gather data for determining the accuracy of this method we included three people in the process. The first person was the data collector and recorded each value. The second person aligned the radar platform with a physical RTK alignment tool as described in 4.1.1. The third person performed the compass alignment without knowing the RTK alignment. The RTK alignment or truth value was unknown to the person performing the alignment and we varied which person did each job. The time to align was recorded as the time it took to get the compass reading and apply it to the radar platform configuration. In our tests we found that this method was good enough in most cases and did not require any additional

development. We also noticed that the results were different between device types so we decided to split the compass experiment into Android and iPhone. While the initial setup time is fast, due to continuous heading adjustments, the overall setup time can be increased and the accuracy is suspect.

4.1.3 Digital Alignment Tool

As part of our radar configurator tool, we have created a digital alignment (Heading Helper) tool that consists of the location of the radar platform on a map and an adjustable line to align the radar platform with a distant object (see fig 4.6). This tool was created prior to this research project and was not modified. The tool involves visually aligning the radar platform with a distant object then also aligning the radar with the object on the map to calculate the heading. A user should stand five to ten feet behind the southern radar (R3) such that when looking at the southern (R3) and northern (R1) radars they are lined up with a landmark that is visible in satellite imagery. Then the user will go into the digital alignment tool and move the red alignment dot onto the landmark. If there is a building nearby, then using a corner of the building as a landmark helps with accuracy since it is easy to see in the distance with the user's eye sight and in the satellite imagery. Accuracy was calculated by first collecting the RTK alignment, then performing this alignment, and later comparing the data. The time to align was recorded as the time it took to visually align the radar with a distant landmark, open the web page and digitally align with the same landmark, and click the button to apply the new heading to the radar configuration. This method appeared to be adequate when there did exist an object in the distance to align with. As mentioned, it helps if the object is tall enough to see in the distance and large enough to show up in satellite imagery. This method fails when there are no discernible features in the environment to align with. Since this is a manual method, human error is also a concern. If the radar platform is in an area without any visible landmarks then an alternate method must be used.

4.1.4 IMU Alignment

By using an Inertial Measurement Unit (IMU) we can electronically and mostly automatically acquire the heading of the radar platform and immediately use it in the software. The IMU includes a margin of error and needs to be re-calibrated prior to each use. Re-calibrating an IMU usually in-

volves turning it along each axis and moving it in a figure eight pattern. Once the IMU is calibrated the alignment process can be automated. An IMU generally works better the more it is moving so an internal Kalman filter can self heal the measurements over time. The accuracy for this method was calculated by first collecting an alignment with the physical RTK method, then performing the IMU alignment, and finally comparing the differences in the measurements. The time to complete this method included calibration of the IMU and waiting for the IMU to have valid calibration status of 2 or higher where 0 is uncalibrated and 3 is fully calibrated. For a radar platform that needs a quick heading calibration before each mission, an IMU appears to be a good solution. However, in our tests with several different IMUs of the same model, we found that it was not accurate enough and in most cases the setup time was increased.

4.1.5 Collaborating Participant Alignment

We desire to use a UAV with RTK GPS as a collaborating participant but for this paper we implemented a manually carried RCS device. We use a radar reflector as the collaborating participant Radar Cross Section (RCS) device. It is attached to the end of a 10ft pole of electrical conduit. The pole has an attached battery powered RTK GPS module for providing the position of the radar reflector RCS device. This collaborating participant is used to create a radar target and GPS location. The GPS data of the RCS device is recorded at the same time that the radar target is recorded. The GPS location of the RCS device is compared with the radar track position for accuracy. The offset angle in degrees from the radar track to the RCS device with respect to the radar platform is calculated by using the law of cosines (see “Fig. 4.2”). The heading alignment is automatically calculated and applied to the radar configuration. The Initial Setup for this method includes making sure that the RCS device is powered on and operational. Starting the RCS device is not included in the alignment time as we believe the final solution (UAV as RCS device) would have a negligible setup time. The time to align for the auto RCS technique starts and includes the time to walk or jog the RCS device about 30 meters out away from the radar platform, then 10 to 20 meters parallel to the radar, and back. The time is stopped once the alignment is stabilized which means that it reached the correct threshold and was written to the radar configuration. A UAV will have the

added benefit of being able to move quickly and further in three dimensions. In this experiment we are only looking at two dimensional position with latitude and longitude.

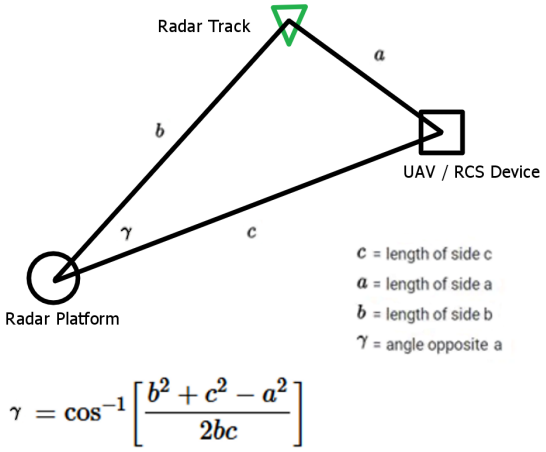

Figure 4.2: By using the radar, radar track, and UAV positions we calculate the radar heading offset.

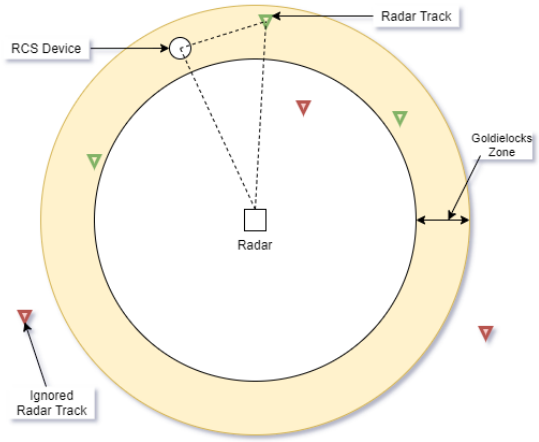

Figure 4.3: The Goldilocks Zone for filtering potential radar track matches.

Figure 4.4: Situational awareness tool showing two radar platforms with two radars each tracking the radar reflector as it is driven by ATV down a runway.

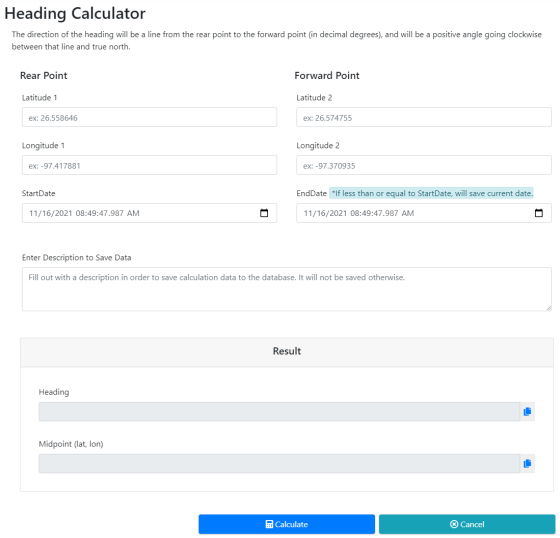

Figure 4.5: Heading calculator web tool.

4.2 Data Collection

As the RCS device moves in front of a radar, the GPS positions of the radar platform and RCS device are sent to our database via web API. The radar platform software calls an API at a rate of 1hz that calculates the offset and returns the new heading of the radar platform by using the GPS location of the radar platform, GPS location of the UAV and perceived GPS location of the radar track that is

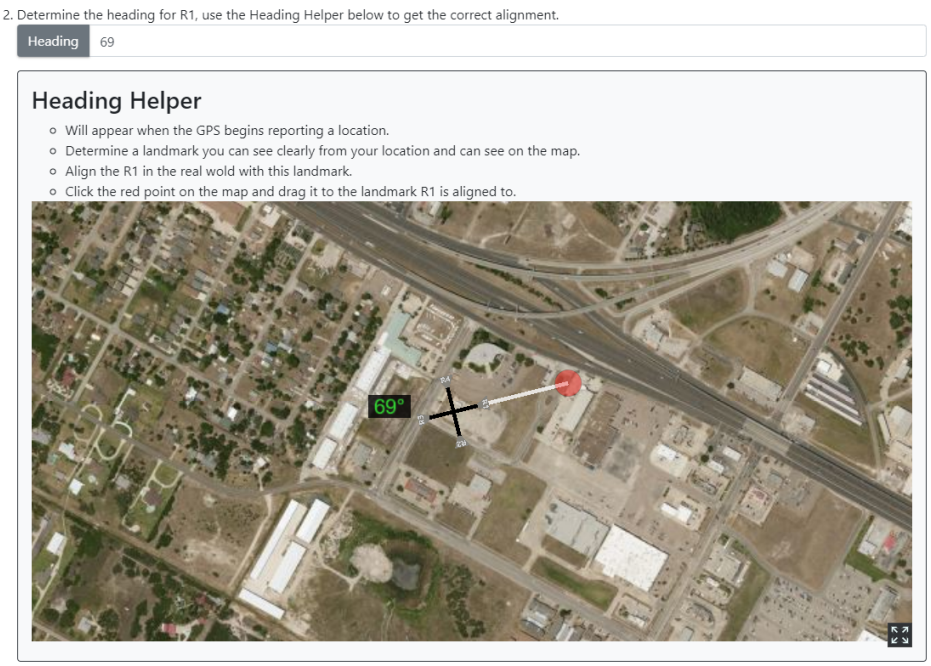

Figure 4.6: Digital Alignment Tool. A user can manually enter a heading from an external alignment technique or position the red dot to visually adjust the Radar Platform heading.

estimated to be the UAV (see "Fig. 4.2"). Due to the possibility of false positive radar tracks we use a simple distance function to find the closest track to the UAV GPS truth position to use as the RCS device track. We are assuming that the initial heading of the radar platform is relatively close to the truth value that we are seeking so the closest track is the correct track most of the time. A human operator is able to get the radar platform aligned to a rough estimate of the correct heading within +/- 10-20 degrees. Otherwise, if the radar is initially 180° off from a correct heading and there are false tracks near the RTK GPS device then those tracks would be incorrectly selected as the correct track to use for the adjustment. The limitation of requiring the radar platform to be aligned in the correct general direction will be addressed in future versions. We also implement what we are calling the goldilocks zone around the radar platform (see "Fig. 4.3"). We will exclude radar tracks that are not located in the goldilocks zone from being potential candidates of the actual radar track. The goldilocks zone is defined as the distance from the radar platform to the RTK GPS of the collaborating participant plus or minus 1.69 meters in the horizontal plane. If the RTK GPS of the collaborating participant moves closer to the radar then the radius of the goldilocks zone radius will decrease. Likewise, if the RTK GPS of the collaborating participant moves away from the radar

then the goldilocks zone radius will increase. If the only radar track that exists in the data is outside of the established threshold of the goldilocks zone then it will be ignored. To summarize, we take the tracks in the goldilocks zone then from those take the one that is closest to the RTK GPS device. The heading of the radar platform will be automatically adjusted by the calculated offset between the radar track and the UAV GPS coordinates. We are effectively rotating the calibration of the radar platform heading until the radar track and UAV matches locations. Since we are assuming the GPS location of the radar platform and UAV to be truth positions then we can expect the orientation adjustment to be correct. We will be saving the data to a database for later review and analysis. We will also visually observe the process in the real world and in a virtual world for accuracy.

Since we do not have a system that runs on a UAV yet we perform the Auto RCS experiments with the radar cross section (RCS) device on a stick. Otherwise known as "The mace of reflection," the manually carried RCS device serves as a stand in for using a UAV. It consists of a 10-15ft lightweight pole with a detectable RCS and GPS at one end along with a power supply and wireless transmitter. With this device we can have someone walk around and move the pole in front of the radar at an appropriate distance to get the data that we need. We also use this as another comparison for the time to set up and reliability analysis. It remains to be seen whether or not it will be more efficient than flying a UAV in front of the radar platform.

For each of the six alignment methods, we performed the experiment twenty times. The experiments and data collection were conducted over the course of several months. For the manual experiments, we collected the time and alignment values on paper then transferred them to excel spreadsheets to calculate the errors. For the auto experiments, the data was automatically collected and processed by software and stored in the database. We extracted all of the data into CSV files then created the graphs using Python with the pandas and matplotlib libraries.

In "Fig. 4.7" we see a comparison of all of data that was collected on the same scale. Looking at all of the data on the same scale is helpful and makes it apparent that some methods are better or worse than others at a glance. However, since the data varies across a larger range, we want to look at each individual set of data on a local scale and we will compare the data in more depth

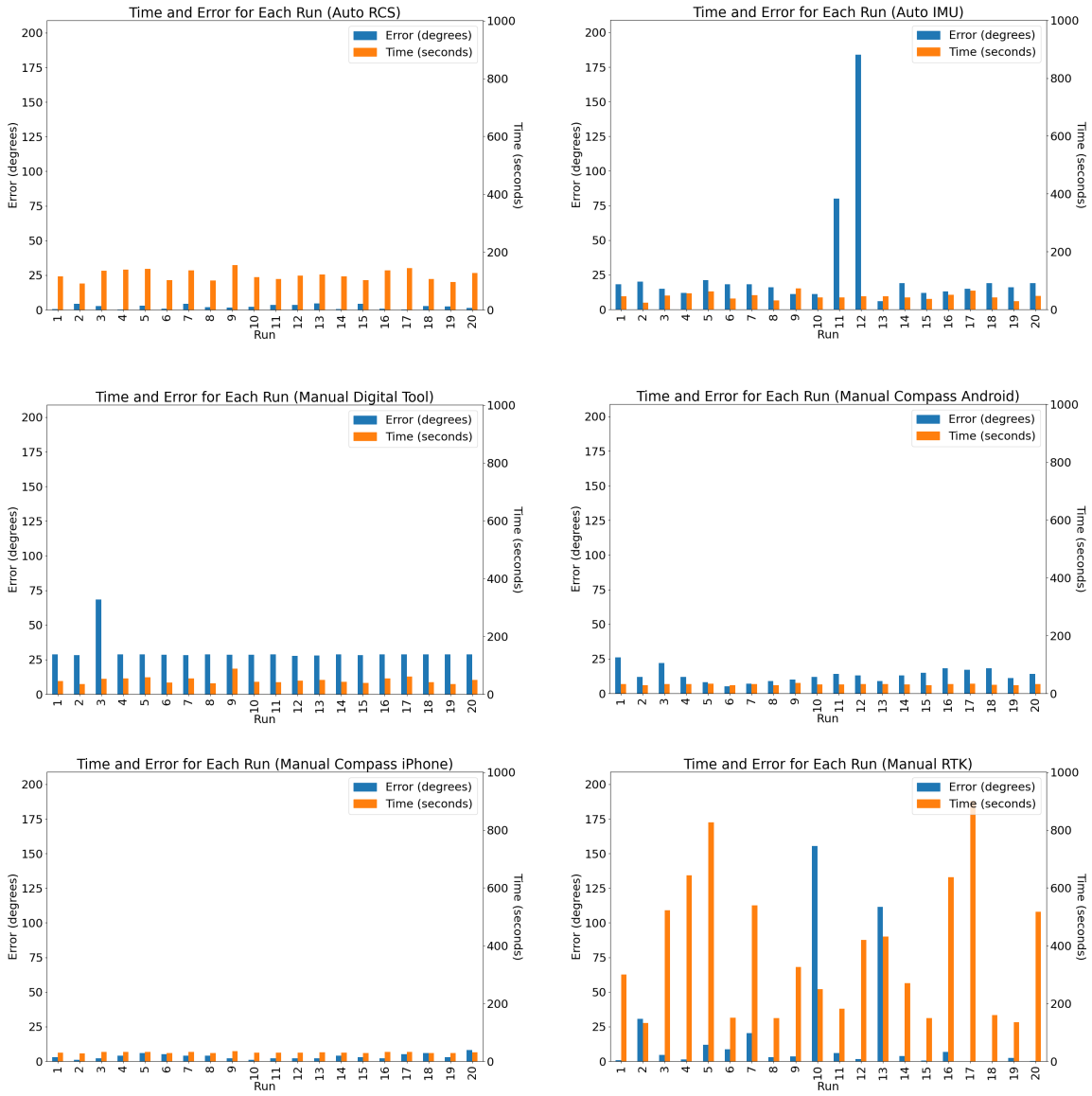

Figure 4.7: Comparison of each method on the same scale.

Figure 4.8: Comparison of error in degrees with time to align for Manual RTK method.

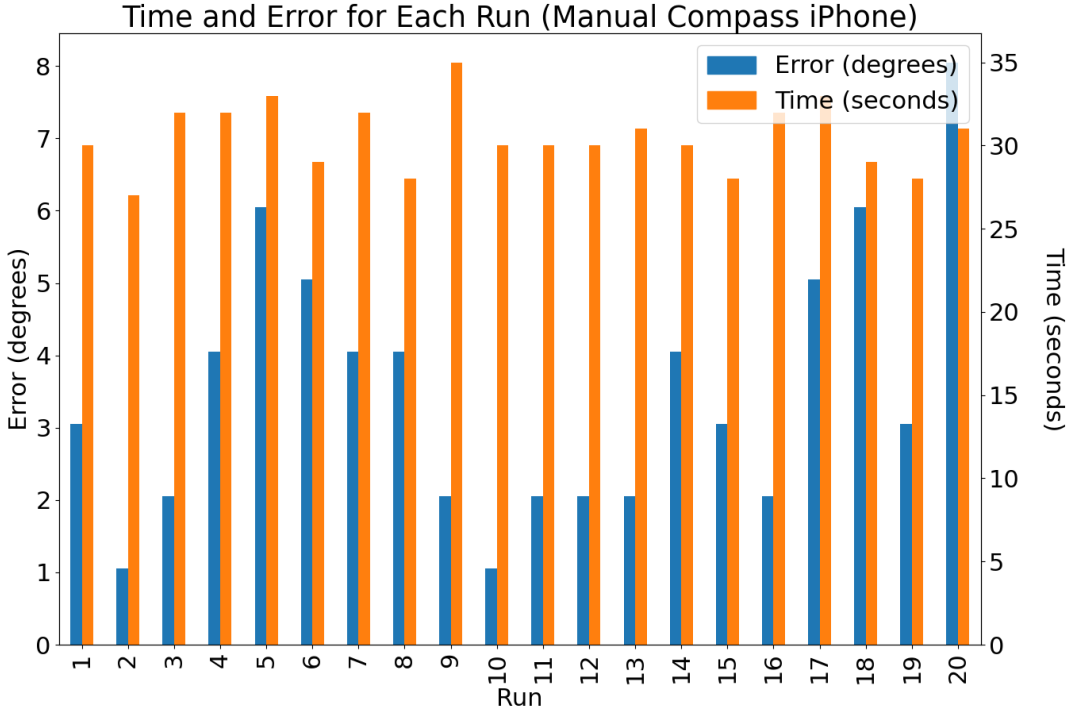

Figure 4.9: Comparison of error in degrees with time to align for Manual Compass iPhone method.

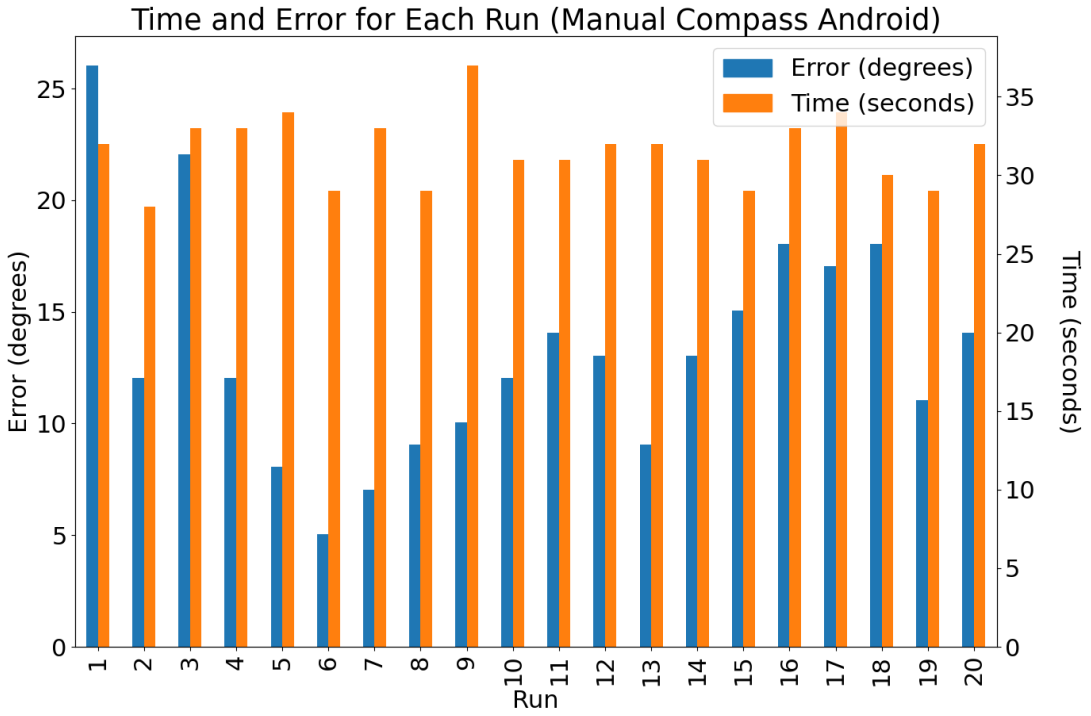

Figure 4.10: Comparison of error in degrees with time to align for Manual Compass Android method.

during the analysis phase. In “Fig. 4.8” we see a bar graph representation of the data that was collected for the Manual RTK alignment method. The graph shows the error in degrees in blue and the time to complete the alignment in orange superimposed over each other. Keep in mind that the scale for each metric is different. For the Manual RTK alignment method we can see that the range of errors was roughly between 1 and 155. If we exclude the largest outliers then the range becomes roughly 1 to 30 degrees. In “Fig. 4.9” we see a bar graph representation of the data that was collected for the Manual Compass iPhone alignment method. We can see that the range of errors was between 1 to 8 degrees when using the specific iPhone devices that we tested with. In “Fig. 4.10” we see a bar graph representation of the data that was collected for the Manual Compass Android alignment method. We can see that the error range was roughly 5 to 27 degrees when using the specific Android phones that we tested. In “Fig. 4.11” we see a bar graph representation of the data that was collected for the Manual Digital alignment method. We can see that the error range was roughly 28 to 68 degrees when using the digital alignment tool. In “Fig. 4.12” we see a bar

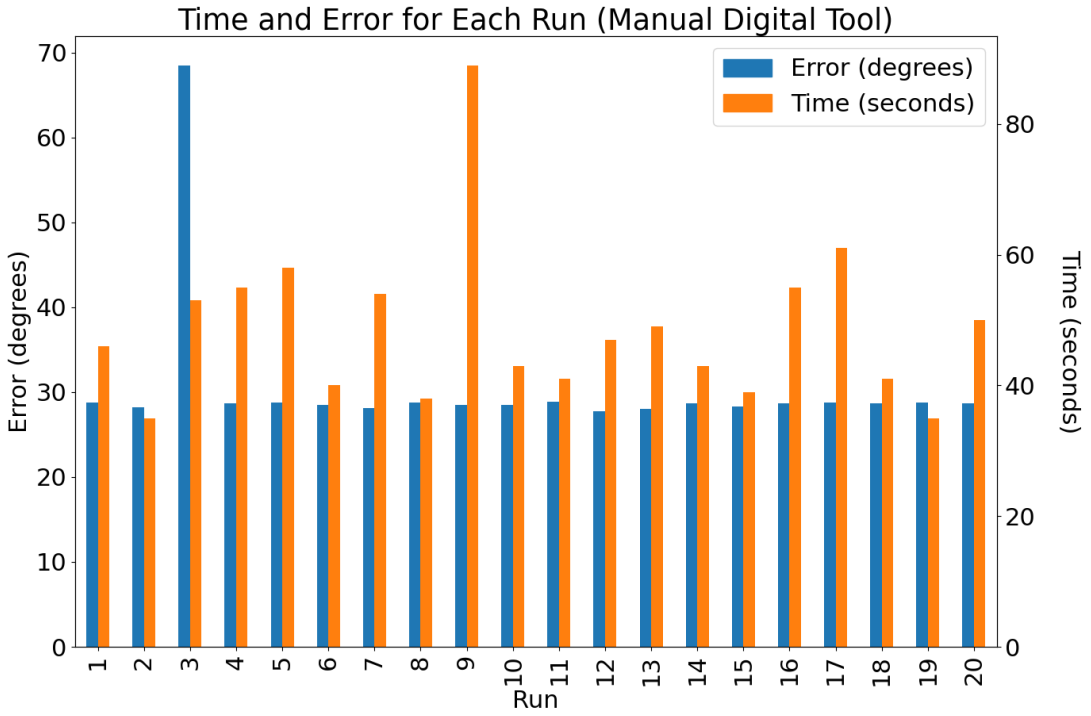

Figure 4.11: Comparison of error in degrees with time to align for Manual Digital Tool method.

graph representation of the data that was collected for the Auto IMU alignment method. We can see that the error range was roughly 10 to 180 degrees when using the specific IMU that we tested. Finally, in “Fig. 4.13” we see a bar graph representation of the data that was collected for the Auto RCS alignment method. We can see that the error range was roughly .2 to 4.5 degrees when using the proposed Auto RCS method.

We tested five different alignment methods. Of the five methods, the ”Compass” alignment method was broken down into two sets of data; one for Android and one for iPhone. The basis of our experiment was to measure the time it took to perform an alignment method and the error of the method. The time to complete an alignment method excluded the time to set up the radar platform and was slightly different for each method but all timers ended when the new alignment was saved to the radar configuration. The manual RTK method compared radar tracks versus ADS-B tracks of live UAVs to get an error. All other methods were compared to the RTK method to get an error offset. We performed each alignment method twenty times for a total of 120 runs of the experiment

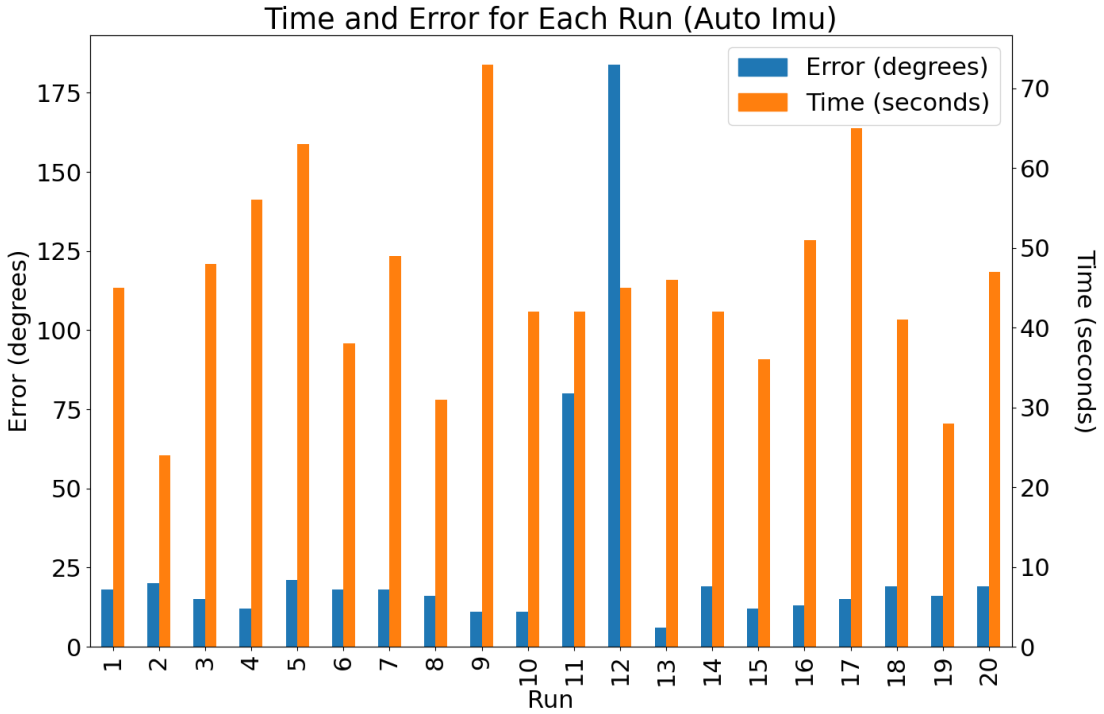

Figure 4.12: Comparison of error in degrees with time to align for Auto IMU method.

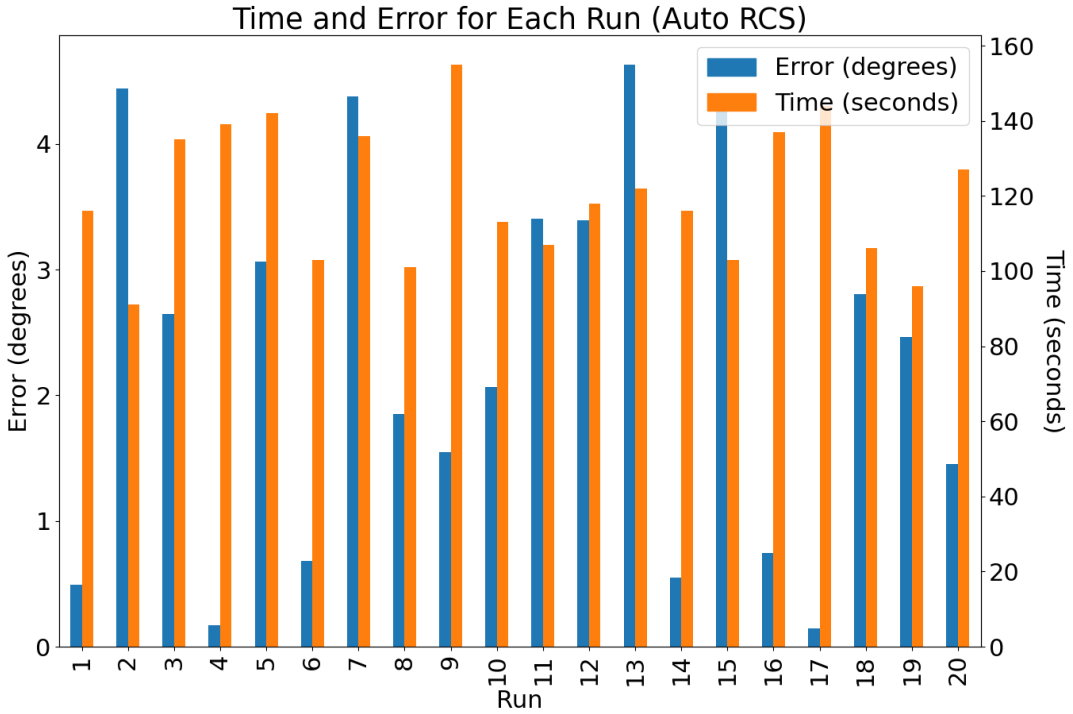

Figure 4.13: Comparison of error in degrees with time to align for Auto RCS method.

which resulted in 120 elapsed time and 120 accuracy data points. We discuss the observations of the data that was collected further in CHAPTER V: ANALYSIS OF RESULTS.

CHAPTER V: ANALYSIS OF RESULTS

The proposed method involves using a collaborating participant RCS device to auto calibrate the radar platform orientation. The custom built RCS device proved functional and provided the necessary data. It is more complex to develop than the manual methods but the final solution generally involves less setup time in the field. A drawback of the "Mace of reflection" RCS device is that the electronics are currently too bulky to put on a small UAV. A potential drawback of the proposed method, once implemented on a UAV, is the time it takes for a pilot to do their pre-flight checklist and get the UAV into the air. The variable time it takes a pilot to complete the UAV setup could potentially nullify the time savings benefit of the proposed method. Ideally, the final result would involve a very limited pre-flight checklist and an automated launch, flight, and land sequence. Also, as mentioned earlier, if the time to setup is not reduced then the positive impact on a search and rescue scenario or other time sensitive operation is minimal. As in all things it seems like there is some give and take between each method. The one thing that stands out is that the proposed method has the potential to more reliably increase the accuracy of the system as compared to the other tested methods. In table 5.1 we show a summary comparison of the alignment methods that we tested and compared with the proposed method. The results show that proposed method warrants further development and testing.

5.1 Methods

Given the three positions in space (latitude, longitude and altitude) of the radar platform, radar track and RCS device (UAV or RCS on a stick) we flatten it to a two dimensional math problem by ignoring the altitude and only focusing on the latitude and longitude. By limiting the equation to a two dimensional math problem we can use simple trigonometry to calculate the angle between the radar track and the UAV with respect to the radar platform. We can use the distance formula to find the distances between each point and then with the law of cosines we can solve for the unknown angle to get our result. The law of cosines equation

$$\gamma = cos^{-1}\left[\frac{b^2 + c^2 - a^2}{2bc}\right] \tag{5.1}$$

where a is the distance between radar track and UAV, b is the distance between radar track and radar platform, c is the distance between UAV and radar platform, and γ is the angle between b and c. The law of cosines will give us the offset between the perceived location of the radar track and the estimated true RTK GPS value of the collaborating participant (see "Fig. 4.2"). Next we need to determine the direction to apply the offset. In order to determine the sign that we should apply to the offset we use the equation

$$IIF\left(\left(\left[tan^{-1}\left(\frac{(T_{lat}-P_{lat})}{(T_{lon}-P_{lon})}\right)\right]-\left[tan^{-1}\left(\frac{(R_{lat}-P_{lat})}{(R_{lon}-P_{lon})}\right)\right]\right)<0,-1,1\right) \tag{5.2}$$

where T is the radar track location, P is the radar platform location, and R is the RTK location. IIf is defined as (expression, truePart, falsePart). So if the result of the expression is less than zero it results in true which returns -1 as the sign to use. If the expression resolves to false then 1 is returned as the sign to use. The resulting sign is applied to the previously calculated offset. Once we have the offset correction we apply it to the orientation of the radar platform to adjust it such that the radar track will be horizontally aligned with the UAV. At this point we have auto aligned the radar platform by using the data provided by the collaborating participant. It is possible that we will need to make some adjustments to the formula to take into account the earths geoid. We will need to monitor for unexpected behavior as well. All things being correct, we anticipated that this solution would be fairly simple to achieve with our existing supporting infrastructure. We also anticipated that it should increase the accuracy of future positioning calculations and reduce the time it takes to set up the radar platform. As it turns out, the proposed method was the most accurate but the second most time consuming to complete.

The Federal Geographic Data Committee (FGDC) manages standards for development, use, sharing, and dissemination of geographic data [6]. The FGDC involves many members as stated in this quote from their National Standard for Spatial Data Accuracy (NSSDA) document [6]:

> The FGDC is composed of representatives from the Departments of Agriculture, Commerce, Defense, Energy, Housing and Urban Development, the Interior, State, and Transportation; the Environmental Protection Agency; the Federal Emergency Man-

agement Agency; the Library of Congress; the National Aeronautics and Space Administration; the National Archives and Records Administration; and the Tennessee Valley Authority. Additional Federal agencies participate on FGDC subcommittees and working groups. The Department of the Interior chairs the committee.

The FGDC provides a National Standard for Spatial Data Accuracy (NSSDA) for estimating positional accuracy that uses root-mean-square error (RMSE). In the standard, formulas and examples are provided for calculating RMSE for horizontal and vertical accuracy. For this paper we are only concerned with the formula for calculating horizontal accuracy. If we assume that the RMSE of each dimension in a horizontal plane is:

$$RMSE_x = sqrt\left[\Sigma(x_{data,i} - x_{check,i})^2/n\right] \tag{5.3}$$

$$RMSE_y = sqrt\left[\Sigma(y_{data,i} - y_{check,i})^2/n\right] \tag{5.4}$$

Where $(x_{check,i}, y_{check,i})$ are the coordinates of the ith check point in the dataset and $(x_{data,i} - x_{check,i})$ is the difference between the estimated truth data and the observed check data and n is the number of observed check points then the horizontal RMSE is:

$$RMSE_r = sqrt\left[\Sigma\left((x_{data,i} - x_{check,i})^2 + (y_{data,i} - y_{check,i})^2\right)/n\right] \tag{5.5}$$

$$RMSE_r = sqrt\left[RMSE_x^2 + RMSE_y^2\right] \tag{5.6}$$

The RMSE method is better suited to the proposed method since the typical percent error formula fails when using latitude and longitude or angular values. For example, if we assume that the typical percent error formula is given as $PE = (|acceptedvalue–experimentalvalue|/acceptedvalue)*100\%$, and we are using two different sets of measured latitude/longitude positions with the same difference from the estimated truth value then they will have different errors which is incorrect. The same applies with angle offsets and it is a bit easier to explain. Assume we have a truth of 110° and the measured offset error is 2° then we have a truth of 171° and the measured offset error is also 2°. In this case one would assume that the percent error should be the same but the typical percent

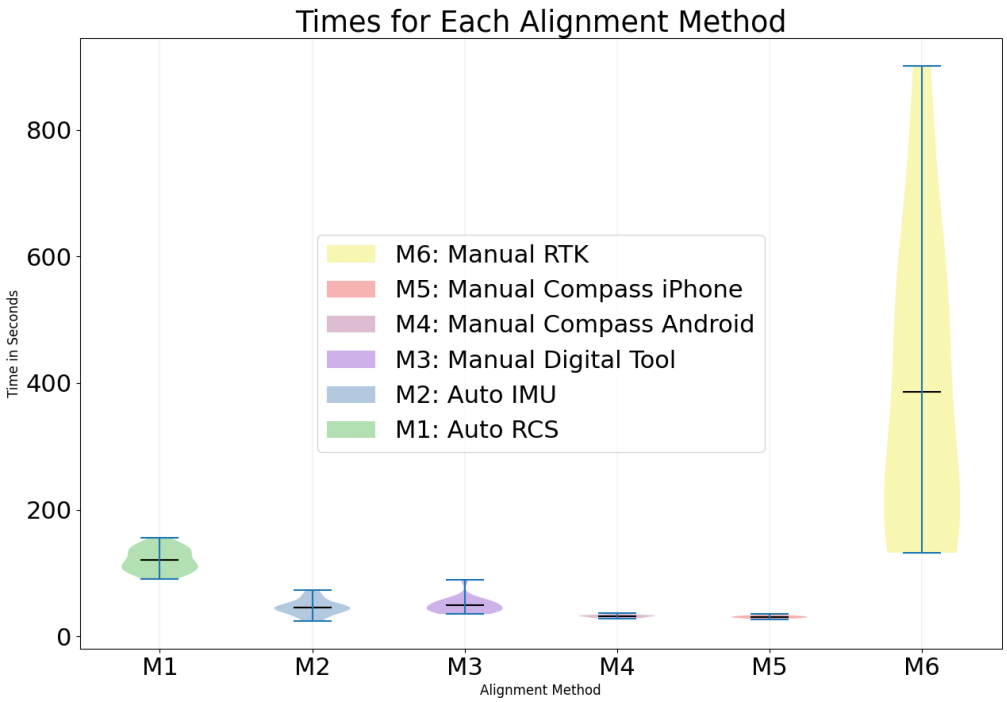

Figure 5.1: Time to complete alignment by method.

error formula will result in two different answers which is incorrect for this type of data. We need to use the NSSDA RMSE formula to calculate accuracy but we should be aware that the resulting number has a different meaning than percent error. The NSSDA RMSE result can be interpreted as 95% of the positions in the measured data will have an error at or below the value.

5.2 Alignment Results

In figure 5.1 we compare the alignment times of each method. We can see that the mobile phone compass methods took the least amount of time to perform while the manual physical alignment method (Manual RTK) took the most time. The proposed method took the second most time to perform. In figure 5.2 we can see that the alignment errors of the Manual RTK and Auto IMU methods had larger amounts of outliers while the Auto RCS (proposed method) and iPhone compass had the smallest amount of outliers. The proposed method actually had the lowest amount of alignment error outliers between all alignment methods. The results are surprising to say the least and we will talk about each result in more detail below.

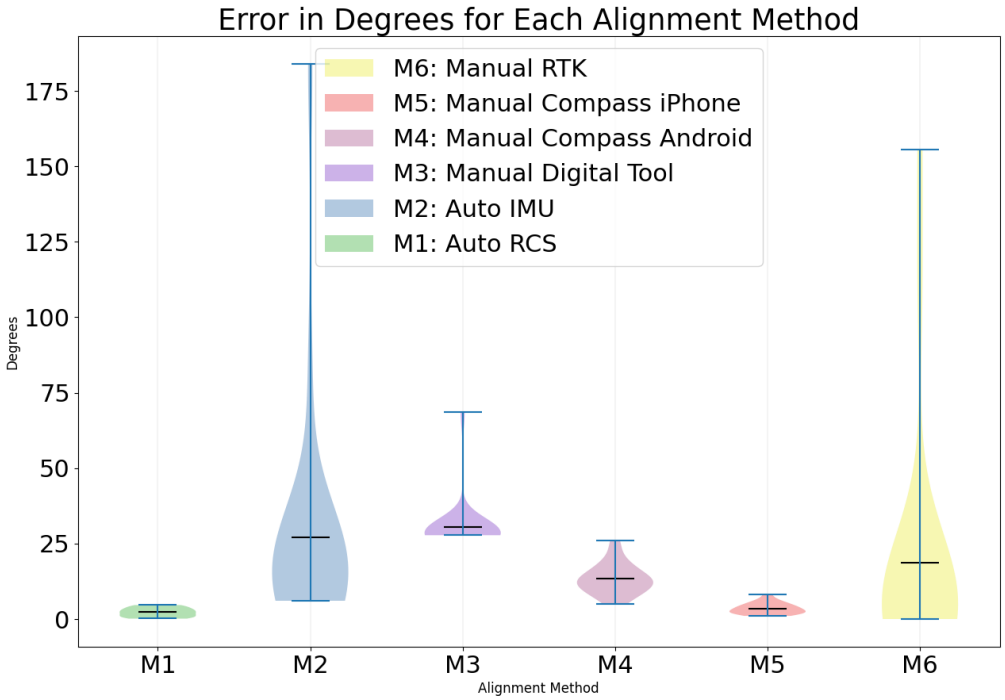

Figure 5.2: Difference between estimated truth data and observed check by alignment method.

5.2.1 Manual RTK Alignment

The Manual RTK alignment method took the longest to complete on average. The shortest time to complete this method was still longer than the worse time of most of the other methods. As part of the setup process we attached the RTK rovers onto the radar mast with screws to ensure that they were in the same spot each time and to prevent them from moving. The process of attaching the RTK rovers to the radar platform mast was included in the alignment time. Occasionally, the RTK units would take several minutes to get a fix for centimeter accuracy. It was also time consuming to collect the GPS coordinates from the RTK units and then manually enter them into the heading calculator. It was trivial time wise to enter the result into the radar configurator tool but we do note that the auto methods save time here as well. We could definitely improve the setup time by making our version of the tool closer to what the industry uses for antenna alignment. We can also lower the setup time by directly interfacing with the RTK units to collect the latitude and longitude values. We were surprised by the variance in error and might be able to attribute it to human error. Since,

for these experiments, we had to manually collect the data from both of the RTK devices, manually enter it into our heading calculator tool, then manually enter it into the radar configuration the opportunity for mistakes was increased. In a later version of this method we would like to directly integrate the RTK devices with the radar platform to reduce the possibility of human error. When we look at "Fig. 4.8", we can see that the majority of the errors are pretty low and the two largest errors are extreme in relation. We also see that this is one of two methods, including the proposed method, that had some errors with 0 degrees. This method warrants further testing to try and identify the cause(s) of the outliers in error. It remains to be seen if the time can be significantly reduced since much of it is waiting for the RTK to gain a fixed solution. It might be worth it to further analyse the time to align by breaking it down into the individual steps so that we can see which part of the alignment is the most time consuming.

5.2.2 Compass Alignment - Android/iPhone

The Manual Compass Alignment provided some surprising results. We tested with multiple iPhone and Android phones using multiple compass apps on each device. What we found was that the phone compass alignments were pretty quick to complete and fairly accurate compared to some of the other methods. We found that each phone had a different error and in some cases different apps on the same phone had different errors. We found that within the set of mobile phone devices that we used to perform the alignment tests that the iPhone devices tended to perform better than Android. This finding is out of the scope of this paper and a larger sampling of various mobile devices would be needed for a more accurate conclusion. However, due to the differences between the selected mobile devices, we decided to split the compass alignment into two different methods for analysis. Mobile phones do have an internal IMU with a small magnetometer for measuring the earths magnetic field [18][7]. A traditional compass needs to be placed on a flat surface or plane to work so that the needle has the least resistance. On the other hand, a mobile phone uses the internal accelerometer in the IMU to detect the pitch and roll of the device and combines that information with the magnetometer in the IMU to calculate a heading regardless of the device orientation. For our tests we made sure that the phone was flat but it turns out that having it flat is unnecessary and

mostly just helps with the human factors and visual interpretation of the results. For a quick and easy alignment solution, using a phone compass can get the radar platform calibrated to roughly the correct location but accuracy is still not where we would like to see it. In order to improve the accuracy of using a phone compass we need improved hardware from phone manufacturers and/or improved apps from app developers.

5.2.3 Digital Alignment Tool

The Manual Digital Alignment Tool was another manual method. This method uses a custom website interface for visually aligning the radar platform with a distant landmark. We expected this method to be low on time when compared to the other methods and it was faster than the average. However, we also expected it to be fairly accurate but it was consistently the worst. We currently think there might be a bug in the software such that the visual representation of the world on the map is not properly aligned or the math for calculating the heading has an issue. The low variance in errors seen in "Fig. 4.11" leads credence to the conclusion that something is wrong in the software. This method was developed prior to this research and we did not modify it all during this research project. We would like to investigate further to determine the cause of the consistent inaccuracy. This method is more complex than some of the other manual methods so it requires more development time. We still feel that it can be an effective method once the bugs are identified and worked out. Once that happens we will test again to see how it compares to the other methods.

5.2.4 IMU Alignment

The Auto IMU Alignment used to take longer to complete on average than most of the other methods due to the physical calibration requirements and our previous radar platform configuration. The data that we collected reflects the latest iteration of this method which has improved over time to reduce how long it takes to complete. We were not able to improve the accuracy with this method when using the specific IMU device that we selected. The main problem with this method was the consistency and accuracy of the alignments. As "Fig. 4.12" shows, the errors are never low enough to be close to useful. This data coincides with our previous experience with the specific IMU. There is a lot of variance in the accuracy of the alignments as compared to the other methods

(see “Fig. 5.2”), which leads us to believe that the particular IMU that we tested is unreliable when stationary. The IMU has the ability to export and load calibration configurations. This feature did not work very well with the selected IMU so we excluded it from our experiments. From the experience and knowledge that we have gained from testing this alignment method, we currently believe that using an IMU for stationary yaw alignment is not the best option. With that in mind we have plans to test out other IMUs to see if the accuracy and reliability are improved.

5.2.5 Collaborating Participant Alignment - Auto RCS

The Collaborating Participant Alignment or Auto RCS method is our proposed method and it involves using a collaborating RCS device to auto-calibrate the radar platform alignment. For this paper we were not able to use a UAV as the RCS device but we were able to use a custom built system, dubbed the “Mace of Reflection,” to perform the tests as described in 3.1 and 4.1.5. Due to the nature of this system we expected and recorded some longer alignment times. This is because we were either walking or running with the device when performing the data collection for the experiments. This method was actually the most accurate alignment method, of the methods tested, with the lowest variance in accuracy (see “Fig. 5.2”). With further refinement and the use of a UAV as the RCS device we can likely get the alignment time down. The accuracy of the proposed method is pretty good compared with the other methods but it is still not good enough. As “Fig. 5.2” shows, there were about 6 out of 20 measurements with an error at or below 1 degree. We would like to have 90% or higher of the errors below 1 degree. With further refinement of our algorithm for determining the correct radar track, we can increase the accuracy to have a higher percentage of results at or below 1 degree of error.

5.3 Results

The results of our experiments show that the proposed method of auto calibration with a collaborating RCS device (Auto RCS) is the most accurate but not the quickest to complete. We can see in “Fig. 5.3” that when removing outliers the Auto IMU and Manual RTK methods have a drastically improved accuracy on the high end. We also see that the proposed method still has the best accuracy at a 95% confidence level meaning that 95% of the time the accuracy is the best of the tested align-

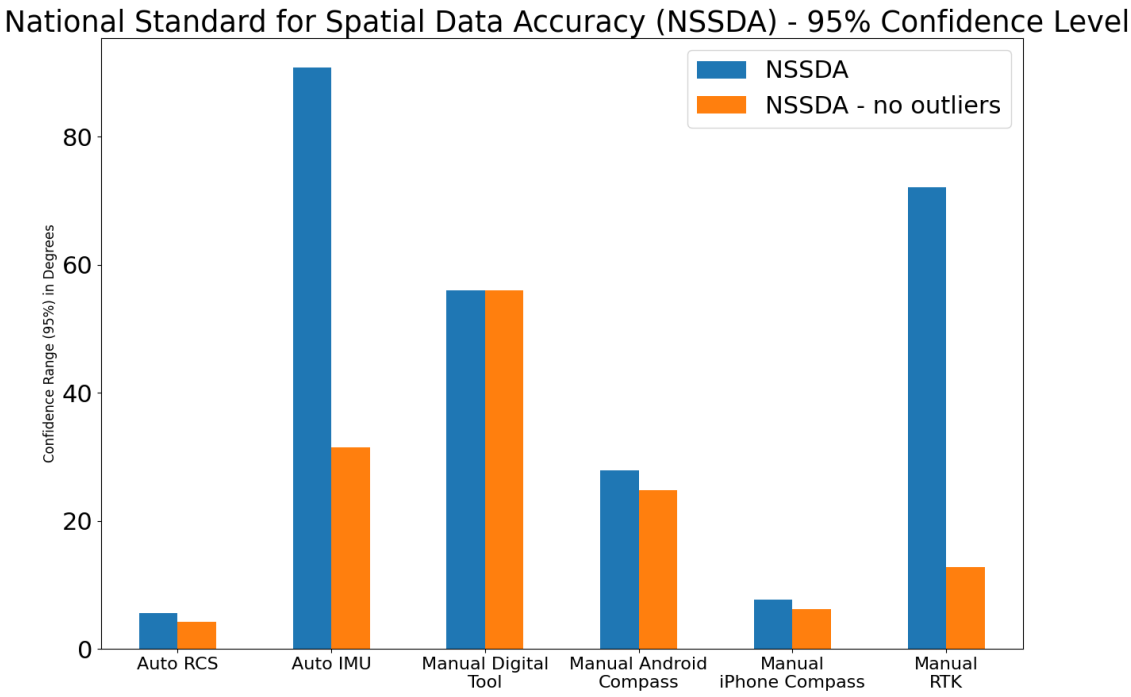

Figure 5.3: Estimated accuracy based on the NSSDA standard which uses the RMSE formula. Accuracy reported at the 95% confidence level means that 95% of the positions in the dataset will have an error with respect to true ground position that is equal to or smaller than the reported accuracy value. In this graph, lower values are better.

ment methods. The specific IMU that we tested had the largest errors and we are aware that this might be due to the specific IMU that was tested. Using a cellphone compass app was the fastest method but the error rate was still higher than desired. The manual RTK alignment method took the longest to complete and had large outliers in the error rate. The manual digital tool had the highest baseline of errors and this is likely due to a bug in the software. There were variations of time and error within each of the alignment methods. In the case of the mobile devices and IMUs we think that more in depth testing of the specific types of devices would provide a better understanding of the pros and cons of each device type. The time to complete the proposed method can be reduced by using an autonomous UAV. Also, for the proposed method, further testing and refinement of the algorithm for picking radar tracks is required to lower the error deltas due to false radar tracks.

Table 5.1: Estimated Comparison of Calibration Methods

Heading Calibration Methods		
Method	***Time***	***Error****
Manual Compass[a] Android	$28 - 37s$	$5 - 26°$
Manual Compass[a] iPhone	$27 - 35s$	$1 - 8°$
Manual Digital Tool[b]	$35 - 65s$	$28 - 69°$
Manual RTK[c]	$132 - 901s$	$0 - 155°$
Auto IMU[d]	$24 - 73s$	$6 - 188°$
Auto RCS[f]	$91 - 155s$	$0 - 5°$

[a]Manual entry with physical compass.

[b]Manual entry with digital heading helper tool.

[c]Manual entry with RTK GPS alignment tool.

[d]Auto with IMU mounted on radar mast.

[f]Auto with radar reflector as RCS (proposed method).

*Estimated error in range of degrees.

CHAPTER VI: SUMMARY AND CONCLUSIONS

We have compared the time to perform multiple alignment methods and calculated the accuracy of each method using the FDGC RMSE formula. The alignment methods that we compared included manually with a handheld compass across multiple device types, manually with a web-based heading helper tool, manually with a custom dual RTK GPS alignment tool, automated with an IMU, and automated with a collaborating RCS device. We found that using a phone compass was fast but not accurate enough. Different phones and different apps have different compass errors so they are not consistent but they could improve as the internal hardware and associated software improves. We found that our web-based heading helper tool was not as accurate as we anticipated so we will need to do more testing and make improvements to see if there are bugs in the system and if it is still viable. We found that using the custom manual RTK alignment tool took the longest to align and is fairly accurate. With further refinement to reduce the setup time and address the accuracy outliers, the physical RTK alignment tool could be improved. We found that the IMU we tested with did not have consistent accuracy but it was fast. The main issues with the IMU are having it mounted to the radar mast when it needs to be physically calibrated before each use and the inaccuracies in the specific device. More exploration into using an IMU could reveal better results. We showed that the proposed method of using a collaborating RCS device is more accurate than the other methods but not the fastest. The reason it was slower was directly related to walking or jogging the RCS device in front of the radar. In future iterations of this work we would like to make the system more compact to enable installing it on a small UAV. Then we would test it again with the goal of reducing the alignment time to less than one minute.

It is clear that multi-sensor fusion with collaborating participants increases the accuracy of the radar ground station heading. Fusing the RTK GPS from the RCS device with the radar track data provides the ability to automatically acquire an accurate heading. This in turn can be used to increase the accuracy of the internal UAV sensors and provide RPICs with the information needed for DAA from non-conforming agents. It is also clear that there is a need to test a wider variety of devices within each of the alignment methods to further improve the data collected for analysis. We

tested one RTK device, one type of radar, one type of IMU, several different Android and iPhone phones, and several custom software solutions. There are still many other devices and alignment methods to explore. Future work could include research that focuses on one alignment method and the myriad of devices related to that alignment method.

While our Auto RCS was not the fastest method we are confident that using an autonomous UAV, instead of a RCS device carried by a human, has the potential to reduce the alignment time. By using auto calibration of the radar ground station heading the time to setup and calibrate the radar system can be reduced. A reduced setup time allows operations such as search and rescue drone operators to quickly start their mission in an environment where DAA radar capabilities are necessary. We have also shown the potential for the radar track conversions to real world coordinates to be more accurate given the accuracy of the RTK GPS readings. This system is also working towards assisting in automated DAA decisions by using multi-sensor fusion to accurately detect nonparticipant UAVs. With more detailed information about moving entities in the environment we can help the UAV and the pilot make better decisions about how to navigate.

In the future we should consider auto-calibration systems that do not use explicit participating collaborators as calibration targets. For example, we could try a machine learning (ML) approach that uses the collected data and a model for recognizing the offset as a human would. The ML approach could potentially be used to identify and filter out false radar tracks in addition to matching the RTK tracks with the correct radar tracks of the RCS device. We should investigate using visual cameras and neural networks to auto-calibrate the system and compare the accuracy and setup time with the methods attempted here. Combining visual information with radar tracks has been shown to be feasible and warrants more research. We should also consider auto-calibration methods that do not involve GPS because UAVs occasionally need to fly in GPS-Denied environments. Imagine UAVs flying in a large warehouse, underground mine, or even on the moon. It seems that our current dependence on GPS for positioning and alignment accuracy will be a problem if we were to use UAVs to explore extraterrestrial bodies such as planets and asteroids where GPS might not exist. Developing navigation systems for autonomous vehicles where GPS is available is definitely less

difficult than in GPS-Denied environments. Perhaps we should also develop systems for quickly creating a GPS-Enabled environment on extraterrestrial bodies or in currently accessible terrestrial GPS-Denied environments. In the mean time it is beneficial to explore the techniques of sensor alignment in GPS-Enabled environments as it has the side effect of helping to increase the insights into sensor alignment, navigation and DAA within a GPS-Denied environment.

The FAA has projected a large increase in the number of UAVs that will soon be flying in the NAS. UAVs or drones are definitely here to stay which leads to the need for DAA solutions that help to keep the airways safe. Radar is capable of providing information for DAA but since it is expensive the coverage is not likely to reach 100% any time soon. This means that mobile radar platforms are still going to be the go to radar solution for DAA. While a stationary radar platform needs to be calibrated a small finite amount of times, a mobile radar platform needs to be calibrated each time it is set up for a mission. Since the time to set up and accuracy are important factors to a mobile radar platform, it is important to explore different methods for achieving higher accuracy and lower setup time. One such method is to combine the use of RTK GPS and an external RCS device to achieve an accurate radar alignment.

NOTES

[1] BVLOS flights involve a UAV and a RPIC that does not have a visual line of sight to the aircraft. The RPIC remotely operates the UAV by using the on board sensor data viewed at a local GCS.
[2] GPS-Denied environments are those where GPS is unavailable or non-existent. UAVs may have to fly in GPS-Denied environments and radar can assist with navigation and DAA.
[3] World Geodetic System 1984 (WGS84) is a standard coordinate system based on an ellipsoid model and it is maintained by the United States National Geospatial-Intelligence Agency. WGS84 is the reference that GPS is based on.
[4] Altitude references include mean sea level (MSL), above ground level (AGL). When converting to MSL or AGL to height, a datum such as WGS84 is used.
[5] LLH a simple text protocol for Latitude, Longitude, and Height as well as solution status. See [3].
[6] KML (Keyhole Markup Language) is an XML standard language for visualizing geographic data. It was submitted to the Open Geospatial Consortium (OGC) by Google in 2008. See [20].
[7] Automatic Dependent Surveillance-Broadcast (ADS-B) is the preferred method for air traffic control in the NAS. An aircraft determines its position via satellite or other means then periodically broadcasts it via ADS-B for others to track.

REFERENCES

[1] Annette L et al Adams. Search is a time-critical event: when search and rescue missions may become futile. *Wilderness environmental medicine*, 18:95–101, 2 2007.

[2] Federal Aviation Administration. Fy 2020–2040 forecast individual sections - unmanned aircraft systems. *FAA Aerospace Forecasts*, pages 51–63, 2020. The FAA Aerospace Forecast 2020-2040 is developed to support budget and planning needs of the FAA.

[3] Copyright (c) 2007□2013, T. Takasu, All rights reserved. *RTKLIB ver. 2.4.2 Manual*, 2013. http://www.rtklib.com/prog/manual_2.4.2.pdf.

[4] Dylan Dumas and Charles-Antoine Guérin. Self-calibration and antenna grouping for bistatic oceanographic high-frequency radars, 2020.

[5] Martins Ezuma, Chethan Kumar Anjinappa, Mark Funderburk, and Ismail Guvenc. Radar cross section based statistical recognition of uavs at microwave frequencies, 2021.

[6] Federal Geographic Data Committee. *Geospatial Positioning Accuracy Standards Part 3: National Standard for Spatial Data Accuracy*, 1998.

[7] Mellisa Gallant. How does your phone know which way is north? an explainer., Feb 2019.

[8] Wei Gao, Ya Zhang, and Jianguo Wang. Research on initial alignment and self-calibration of rotary strapdown inertial navigation systems. *Sensors*, 15(2):3154–3171, 2015.

[9] Karol Hausman, James A. Preiss, Gaurav S. Sukhatme, and Stephan Weiss. Observability-aware trajectory optimization for self-calibration with application to uavs. *CoRR*, abs/1604.07905, 2016.

[10] Ziyang Hong, Yvan Petillot, Andrew Wallace, and Sen Wang. Radar slam: A robust slam system for all weather conditions, 2021.

[11] Qiang Ji and Songtao Dai. Self-calibration of a rotating camera with a translational offset. *IEEE Transactions on Robotics and Automation*, 20(1):1–14, 2004.

[12] Niels Joubert, Jane L. E, Dan B. Goldman, Floraine Berthouzoz, Mike Roberts, James A. Landay, and Pat Hanrahan. Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles. *CoRR*, abs/1610.01691, 2016.

[13] D. Kellner, M. Barjenbruch, K. Dietmayer, J. Klappstein, and J. Dickmann. Joint radar alignment and odometry calibration. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 366–374, 2015.

[14] Neelakantan Nurani Krishnan, Ratnesh Kumbhkar, Narayan B. Mandayam, Ivan Seskar, and Sastry Kompella. How close can I be? - A comprehensive analysis of cellular interference on ATC radar. *CoRR*, abs/1704.07491, 2017.

[15] R. C. Luo, Chih-Chen Yih, and Kuo L. Su. Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE sensors journal*, 2(2):107–119, 2002.

[16] Anthony Macintyre, Joseph Barbera, and Edward Smith. Surviving collapsed structure entrapment after earthquakes: A ”time-to-rescue” analysis. *Prehospital and disaster medicine*, 21:4–17; discussion 18, 02 2006.

[17] Stephen J. Maybank and Olivier D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[18] 2014 Michael Setzer nbsp;|nbsp; Jan 26. How compass apps can tell direction, Jan 2014.

[19] Guillem Reus Muns, Kumar Vijay Mishra, Carlos Bocanegra Guerra, Yonnina C. Eldar, and Kaushik R. Chowdhury. Beam alignment and tracking for autonomous vehicular communication using ieee 802.11ad-based radar, 2017.

[20] Open Geospatial Consortium Inc. *OGC®KML*, 2008. https://www.ogc.org/standards/kml/.

[21] Christos Papachristos, Shehryar Khattak, and Kostas Alexis. Autonomous exploration of visually-degraded environments using aerial robots. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 775–780, 2017.

[22] M. E. Rovkin, V. A. Khlusov, N. D. Malyutin, A. V. Hristenko, A. S. Novikov, D. M. Nosov, M. V. Osipov, M. O. Konovalenko, A. O. Marchenko, and V. E. Ilchenkoy. Radar detection of small-size uavs. In *2018 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, pages 371–374, 2018.

[23] Christoph Schöller, Maximilian Schnettler, Annkathrin Krämmer, Gereon Hinz, Maida Bakovic, Müge Güzet, and Alois C. Knoll. Targetless rotational auto-calibration of radar and camera for intelligent transportation systems. *CoRR*, abs/1904.08743, 2019.

[24] Varuna De Silva, Jamie Roche, and Ahmet Kondoz. Robust fusion of lidar and wide-angle camera data for autonomous mobile robots, 2018.

[25] Nihal Singh, Dibakar Sil, and Ankit Sharma. Parallelized instantaneous velocity and heading estimation of objects using single imaging radar, 2020.

[26] Milt Statheropoulos, Agapios Agapiou, George Pallis, K. Mikedi, Sofia Karma, Pierina Julia Vamvakari, Miranda Dandoulaki, Fivos Andritsos, and Charles Thomas. Factors that affect rescue time in urban search and rescue (usar) operations. *Natural Hazards*, 75, 01 2015.

[27] Philippe Terrier and Yves Schutz. How useful is satellite positioning system (gps) to track gait parameters? a review. *Journal of neuroengineering and rehabilitation*, 2:28, 10 2005.

[28] Jian Wang, Yongxin Liu, and Houbing Song. Counter-unmanned aircraft system(s) (c-uas): State of the art, challenges, and future trends. *IEEE Aerospace and Electronic Systems Magazine*, 36(3):4–29, Mar 2021.