Generalised state estimators for robotic platforms through the use of improved sensor characterisation and variance modelling

Harry A.G. Pointon

A thesis submitted in partial fulfilment of the requirements of

Liverpool John Moores University for the degree of

Doctor of Philosophy

December 2019

Declaration

I, Harry A.G. Pointon, declare that this thesis titled, 'Generalised state estimators for robotic platforms through the use of improved sensor characterisation and variance modelling.' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Abstract

The aim of this research is to develop an improved representation of the sensor variance in a state estimator and assess its viability in conjunction with generalised system models. This would enable the use of a single state estimation system across many different platforms.

A key challenge in the safe deployment of Unmanned Aerial Vehicle (UAV) systems is localisation. In built up environments traditional Global Navigation Satilite System (GNSS) systems become unreliable, and other sensing systems are often limited in application. Deploying UAV platforms in complex or safety critical operations often requires a legal exemption, with a demonstration of robust, practical operation of the equipment proposed. To this end, a generalised state estimator would allow repeated use of the same, experimentally validated systems.

This research presents a methodology to characterise the principle input sensor, in this case, an Ultra Wideband (UWB) system through the use of the Robotic Total Station (RTS). The project continues, by demonstrating the implementation of a sensor variance model in the commonly used Extended Kalman Filter (EKF) framework, in both ground and aerial platforms. The work concludes, with a demonstration of a generalised state estimator in use for both a ground and aerial platform, and shows a more stable, noise tolerant output, assessed using the RTS system.

Acknowledgements

I would like to thank my family and friends for their help and support over the past 8 years, what I have achieved would not have been possible without their understanding, patience, and good humour. I owe them a great deal.

I would also like to thank the staff of LJMU for their patience and knowledge provided in equal quantities. Especially my supervisors, Dr Frederic Bezombes, Dr Christian Matthews and Dr Badr Abdullah.

Finally I feel it is important to recognise the late Prof. David Burton, without whom the drone research at Liverpool John Moores would not have been possible, and none of us would have been brought together.

Contents

List of Figures

List of Tables

Abbreviations

3D - 3 Dimensional

2D - 2 Dimensional

3-DOF - 3 Degrees of Freedom

6-DOF - 6 Degrees of Freedom

AKF - Adaptive Kalman Filter

BVLOS - Beyond Visual Line of Sight

BLDC - Brushless Direct Current

CKF - Cubrature Kalman Filter

CAA - Civil Aviation Authority

DCM - Direction Cosine Matrix

DC - Direct Current

EKF - Extended Kalman Filter

ESC - Electronic Speed Controller

ESKF - Error State Kalman Filter

EMC - Electro Magnetic Containment

FCU - Flight Control Unit

FAA - Federal Aviation Administration

GNSS - Global Navigation Satellite System

IMU - Inertial Measurement Unit

LiPo - Lithium Polymer

MBEKF - Model Based Extended Kalman Filter

MEMS - Micro-electro Mechanical System

NCLOS - Non-Clear Line of Sight

PID - Proportional Integral Differential controller

RTS - Robotic Total Station

ROS - Robot Operating System

sUAS - small Unmanned Aerial System

SLAM - Simultainious Localisation And Mapping

UWB - Ultra Wideband

UAV - Unmanned Aerial Vehicle

UAS - Unmanned Aerial System

UKF - Unscented Kalman Filter

WLAN - Wireless Local Area Network

Chapter 1

Introduction

1.1 Unmanned Aerial Vehicles

An Unmanned Aerial Vehicle (UAV) is a tool. As with many tools, their design, manufacture, and use has evolved over time; with modifications made to enhance performance, make use of new materials and construction techniques, and improve safety [1, 2]. In the past five to ten years, UAVs have seen an increase in their application due to a number of factors; new technologies allowing for better performance, increased longevity, lighter sensor payloads and cheaper construction have contributed to this increase [3]. One of the factors in this increase in UAV deployment and adoption may be attributed to smart phone development. The improvements made in early smart phone systems were quickly adopted by researchers working in UAV platforms for their compact size, and sensor dense nature [4]. This effect has also been credited to an extent to the automotive market [5]. Now, with regulatory bodies giving freedom to lighter platform sizes, UAVs are being produced at ever lower weights.

With these developments, established industries have taken up the UAV tool for new applications. The applicability, feasibility and longevity of these new applications may be up for question, with many of the new uses for UAVs also achievable through the use of long sticks and adhesive. However, irrespective of practicality, UAV use has increased. The aviation regulator in the United Kingdom, the Civil Aviation Authority (CAA) defined in a report released in July of 2018 that an anticipated increase in UAV platforms required an increase in research of UAV systems and regulation [6]. The American Federal Aviation Administration (FAA) document, released in 2019, provides data showing an increase in "non-model small Unmanned Aerial System (sUAS) " of 44% since 2018 [7]. In this case "non-model" sUAS refers to commercial UAVplatforms, where model sUAS is the designation used by the FAA to distinguish between recreational and commercial platforms [7]. With this increase, the legal and safety aspects of widespread UAV adoption have become increasingly relevant.

Of particular interest is Beyond Visual Line of Sight (BVLOS) UAV operation. Under current UK law, as administered by the CAA , BVLOS flight requires a safety case be produced and permission granted on a case by case basis. Within this, the permission for autonomous or high authority automated flight requires a set of specifications be met. Among these according to chapter 3, paragraph 3.23 of CAP 722:

> ”As such, UAS developers will need to ensure that any data related to autonomous control has a sufficient level of integrity such that the ability to comply with basic safety requirements is maintained. This will require the development of appropriately robust communication and data validation systems.”

This requirement includes navigational data used for the autonomous system. As will be discussed later, for all of the benefits of a UAV system to be fully reaped the crewing impact on a team must be minimal. Therefore, it would be unacceptable for a pilot, and at least one spotter be absented from the main team. To minimise crew impact the UAV system must require minimal interaction from the main crew body, to accomplish this an autonomous system would be preferable. In the case of UAV platforms, navigation data is usually subject to filtering and post processing [8, 9, 10, 11]. Therefore, it may be inferred from CAP 722 that there is a requirement to validate the tools used in this filtering and post processing. In many cases, this filtering or post processing is accomplished through the use of "state estimators".

In the coming chapters a technical description and overview of state estimation will be given; however, at this stage it is prudent to introduce the concept in broad terms to allow for a better understanding of the choices made and why this research is interesting.

In the practical world there may not be a complete model of a system. Assumptions must always be made, and factors discounted due to their influence being too small, their effects unknown, or presence unaccounted for. The same may be said for any observation made. There will always be a degree of uncertainty about any direct measurement made of a state, and any indirect observation is governed by the same principle of incomplete models; meaning the act of inferring by the information contained within the observation requires its own incomplete model. These concepts do not preclude the use of mathematical prediction, as their influence is often small, but rather limit its scope in some way or another.

As an example of this, consider a thrown ball, observed by a second party. The observer may construct an idea of the position and velocity of the ball based upon an observation, however this will be subject to the same uncertainties mentioned previously. Based on the observers' knowledge of physics, it may be assumed that their prediction of the position of the ball in half a second would be reasonably accurate, given its momentum and speed. However, as the time between the initial observation and the prediction increases the accuracy of the estimate will decrease due to unaccounted for, or unmeasured factors. These factors could include wind speed variation, angular velocity of the ball, or incomplete modelling of the system. To counteract this drift, regular observations may be made to correct for errors in the prediction. In broad terms, this is state estimation as used in the localisation of UAVs.

As shown in this example, there are two key elements in a state estimator, the observation, and the system model. In many projects aimed at improving the performance of a localisation system, the approaches broadly fall into two categories; investigating the formulation of the state estimator, or improving either the system model or observation source. In terms of the observation, a simple method to improve the performance is better sensing technology, however this broadly increases the cost of the system. Another approach would be to improve the representation of the system in the system model. This approach however reduces the generality of the state estimator, meaning it is applicable in less varied circumstances. As will be described in more depth later in the thesis, through the use of alternate ground truth techniques, better sensor characterisation is possible. This allows for a third approach, and shapes the direction of this project. Can a more accurate representation of the observation uncertainty improve a state estimation system? Does this also maintain sufficient generality to allow application to a varied set of environments and circumstances?

Work has been conducted into how the validation or state estimation systems for UAV platforms and similar systems used elsewhere in aviation [12][5][13]. It has been demonstrated that this is possible, however a hurdle to the widespread use of these techniques is the required cost in terms of time and resources, as much of this work is conducted through experimental procedure [12][5][13]. For every new state estimator, the process must be repeated. Therefore, there is an advantage to a state estimator with a generality which allows for use in multiple cases without modification and the associated re-validation.

1.2 Current Applications

Before defining the specific direction and focus for this project, it is logical to give an overview of the relevant current applications of the technology. As outlined later in Section 1.5, each future chapter of this thesis will also contain a specified state of the art section, relevant to the area at hand. This section will deal with giving a general overview of the use of UAVs relevant to this project.

Firstly, in the field of conservation, much work has been conducted through the use of fixed wing and rotary wing UAVs for data collection in a variety of circumstances [14, 15, 16, 17]. The work identified by Paneque-Galvez et al. using small unmanned vehicles for environmental data collection was conducted as early as 1983 by Tomlins et al [18, 19, 20]. An example of the advantages of UAV deployment is discussed by Koh and Wich; in this publication they cite the reduced cost and increased frequency of observation (high temporal resolution) in comparison to satellite data collection methods [21]. In the works conducted by Koh and Wich, Paneque-Galvez et al, and Baena et al, the UAV platforms typically used consumer grade digital cameras to capture the required data. To contextualise this data, the image locations are also normally recorded through the use of onboard GNSS [15, 20, 21]. Although the works listed here demonstrate clear development and use cases in the field, minimal note is taken as to the accuracy of the positioning of the UAVs or the uncertainty associated with the geo-referencing of the images captured.

Another application topic for UAVs in an established field is built environment and civil engineering. In this field, UAVs have been equipped with tools such as Lidar, radar, and conventional tools such as RGB and multispectral cameras [22, 23, 24]. Study indicates that the use of UAV platforms would aid in the monitoring of a wide range of systems including gas pipelines, power lines, industrial facilities, and structures [25, 26, 27, 28]. Each of these areas of study share the common thread of reduced deployment cost, allowing for increased inspection frequency.

Finally, in the field of emergency response and Search and Rescue (SAR) applications UAVs have been found to be particularly useful [29]. Through the use of remote sensing techniques, the process of searching may be made more efficient, and the process of reporting findings may be made more reliable [30]. Existing technology allows for remote vehicle deployment quickly and efficiently, however the most appropriate applications of this technology is still under investigated [30]. As described in a report by Her Majesty's Inspectorate of Constabulary and Fire and Rescue Services (HMICFRS), of the "surveyed forces in March 2017, we found that 28 of the 43 forces in England and Wales" with 9 more forces considering their introduction. The report states that of the 32 forces surveyed none were able to produce analysis on the efficiency or effectiveness of drones. However, the report claims a decreased requirement of National Police Air Service (NPAS) support as an impact of UAV introduction. In the UK in 2016/2017 the NPAS logged 16,369 operational hours. With 26,856 calls from the police forces in England and Wales. The HMICFRS report in 2016/17 stated the NPAS cost per flying hour was £2,820. This figure is highly variable, depending upon how it is estimated, for example the estimate from the same source in 2009 was £1,335. Through the incorporation of UAV platforms as shown in other fields, savings may be made, and the demands of crewed aviation platforms reduced.

As has been shown in this section, a key requirement in the future development of UAV use is an improved localisation system. Not only is the localisation system required to improve the operational safety of the system as shown here, but as demonstrated by the works highlighted in Section 1.2, to relate and fully utilise any data collected from onboard sensing systems.

1.3 Novelty of Research

The novelty of the work presented in this thesis is falls primarily into two categories. First, the research presented here outlines a new methodology for assessment of sensor and navigation system performance. Through the use of a RTS, positional measurements of an accuracy available only through the use of a visual motion capture system such as VICON or similar system may be captured. However, the RTS allows for greater range, and longer range operation in outdoors environments. This novelty includes the synchronisation of this data with the experimental data using methods not previously described in the literature, through the use of networking of all sensors. Secondly, this data is then used to construct a model of the sensor variance, informed by and expanding upon the existing literature. This data is then incorporated into existing state estimation algorithms in a novel manner, by replacing the static sensor variance with the constructed model. This allows for a more robust, general formulation, with fewer drawbacks seen in alternatives, such as the Adaptive or Cuberature Kalman Filters.

1.4 Aims and Objectives

The aim of this project and research is to demonstrate an alternate method for improving state estimator performance, through the use of an improved sensor variance characterisation. The goal is to determine the effectiveness of employing a general system model, and retaining low cost sensors, while still improving estimator performance. To accomplish the aims stated here, the objectives of this project are to:

- Investigate of the use of more detailed sensor characterisation in state estimators.
- Study the effects of an alternate representation of this sensor characteristic model.
- Compare the performance of this estimator with and without a sensor characteristic model.
- Formulate a platform specific and platform agnostic state estimator.
- Explore the use of a platform agnostic state estimator, with this sensor characteristic model.

To achieve this, a single observation source is to be identified and studied. As stated, the aim of this project is not to improve the performance of a single sensor, but rather to demonstrate a methodology to better represent sensor variance, and determine the usefulness of this approach. Therefore, any methodology shown here should be applicable to other sensor systems.

1.5 Thesis Structure

This section of the introduction chapter will outline the structure of the thesis and breakdown the major points within the project at each stage.

The thesis begins with an introduction to UAV platforms, and outlines the current uses and research in the field. It highlights the novelty in this project, outlining the aims and objectives for the research. Finally outlining the structure of the thesis.

Chapter one will deal with the initial introduction of the mechanics behind a rigid body system in 6 degrees of freedom. It will then move onto a definition of the pertinent terms and metrics that may be used to assess navigation sensor systems. Finally, these topics will be used to first assess the performance of an Inertial Measurement Unit (IMU), then implement this device in a fashion to allow for an attempt to be made to estimate the position and orientation of of a platform with this sensor alone.

Chapter two will build upon the work in chapter one to estimate reliably the position and orientation of a platform. A review of the existing sensors and technologies available and in use in both academic and industrial applications will be carried out. This information will then be used to determine the most appropriate sensor system to augment the estimation of the platform position and orientation. With the sensor to be used chosen, a methodology to fuse the sensor data with that of theIMU will be determined, based upon the literature. The chapter will then lay out the experimental practice to be used to test and assess the system defined previously. This project aims, as defined previously, to construct a platform agnostic state estimator, with a monitorable "health", through the use of improved sensor characterisation. As an initial stage, for clarity and simplicity, this chapter will utilise a ground rover, with three degrees of freedom considered. The reasoning for this is described in depth in this chapter.

With the method used to estimate the states (orientation and position) of the platform established, and a ground truth system defined, the project will begin to investigate the sensor variance of the observation source in more detail. The third chapter then describes how this investigation will seek to build a more full picture of the behaviour of the sensor, and construct a model to represent this as a function of a variable that is practical to monitor or estimate, within the scope of this project. This model will then be incorporated into the state estimation system formulated in the previous chapter. As in chapter two, this system will be deployed through the use of a ground rover. To determine the resilience of the state estimator to increased noise levels the observation sensor will be placed in a non-ideal configuration.

With the work conducted in the previous chapters, the fourth chapter of the thesis will develop the formulated state estimator and combine the lessons learned with the six degree of freedom equations defined in chapter one. This chapter will describe the implementation of the state estimator utilising the sensor behaviour model for an UAV platform. This requires an increase in degrees of freedom from three to six, as the platform is less constrained. The development, construction, and tuning of the UAV testing platform will be described. Due to increased sensor noise levels seen in this stage of the testing the analysis will become impractical. It was decided that a final section of the project would be to investigate this increased sensor noise, and define a new experimental methodology to test the six degree of freedom state estimator with the included sensor behaviour model.

The fifth chapter of this thesis will present the investigation into the increased noise levels of the observation sensor, testing a variety of possible causes identified in the literature shown in the previous chapters. This chapter will then define a new experimental methodology and setup procedure to reduce or remove the occurrence of the identified factors. With this complete the chapter will move on to the testing of the six degree of freedom state estimator formulated in the previous chapter through the use of a ground rover, and aerial platforms. The results will then be described and an analysis carried out to assess the functionality of this state estimation formulation.

The final chapter of the thesis describes the work conducted during the project, outlines the principle lessons learned, and the findings gathered throughout the project. The chapter continues on to describe future work which may be carried out with this project as a basis, suggesting topics of further study and possible modifications which may yield interesting results, but were beyond the scope of this study.

1.6 Works published in the conduct of this research

- **Understanding Evidence Dispersal Caused by the Effects of Using Unmanned Aerial Vehicles in Active Indoor Crime Scenes**, 2018, ICFS 2018 : 20th International Conference on Forensic Sciences, Elizabeth Parrott, Harry Pointon, Frederic Bezombes, Heather Panter

- **Uncertainty Characterisation of Mobile Robot Localisation Techniques using Optical Surveying Grade Instruments**, 2018, Sensors, MDPI, Benjamin J. McLoughlin, Harry A. G. Pointon, John P. McLoughlin, Andy Shaw and Frederic A. Bezombes

- **Towards a Model Based Sensor Measurement Variance Input for Extended Kalman Filter State Estimation**, 2019, Drones, MDPI, Harry A. G. Pointon, Benjamin J. McLoughlin, Christian Matthews and Frederic A. Bezombes

- **Mapping of Ultra-Wide Band Positional Variance for Indoor Environments**, 2019, TAROS 2019: Towards Autonomous Robotic Systems, Harry A. G. Pointon, Frederic A. Bezombes

- **Requirements and Limitations of Thermal Drones for Effective Search and Rescue in Marine and Coastal Areas**, 2019, Drones, MDPI, Claire Burke 1, Paul R. McWhirter, Josh Veitch-Michaelis, Owen McAree, Harry A.G. Pointon, Serge Wich and Steve Longmore

Chapter 2

Rigid Body Motion

This chapter serves as an introduction to rigid body motion in 3D and 6 Degrees of Freedom (6-DOF). From the introduction of this work we found that one of the main challenges in the use of UAVs and other autonomous systems in cluttered BVLOS environments is the estimation of vehicle motion and position. As previously defined, the aim of this project is to investigate the implementation of a system capable of operating under a variety of conditions and with different platforms. Therefore, an holistic system model is preferred. In this case, for objects moving in 3D the assumption of rigid body mechanics and freedom in 6 degrees is appropriate, assuming no significant components are joined flexibly to the main structure.

When discussing sensor measurement, the manner of the measurement is a useful attribute to mention. If a sensor is affixed rigidly to the platform in question, any forces applied to the sensor are taken as applied to the platform as a whole. Examples of this are IMU in a strapdown configuration [31]. As opposed to a sensor taking measurements of the environment relative to the agent. An example of this could be optical flow cameras, or certain laser line scanning based Simultainious Localisation And Mapping (SLAM) algorithms

[32].

One of the first stages in the selection and implementation of sensors in motion estimation is the noise characterisation [33][34]. This chapter provides an introduction to this and demonstrates the process on two IMUs. The findings from this are then compared to those from the second half of the chapter, when the IMUs are implemented.

In order to gain a good understanding of the complexity of the issue an initial experiment was undertaken using two types of IMUs. Firstly a consumer grade, raw output MPU6050 is deployed as an input for a 6-DOF rigid body motion estimator. This takes into account neither sensor noise or uncertainty. Secondly this same system is used in conjunction with a BNO055 IMU. The BNO055 comes preconfigured with sensor filtering and, sensor fusion between both an accelerometer-gyroscope pair and a magnetometer, giving significantly more stable readings. A comparison of the 2 systems is then made, and the conclusions drawn serve as the foundation for the future motion estimation work laid out in this thesis. The IMUs used in this section were chosen for their representation of the different approaches in IMU integration. The MPU6050 is a low cost chip, with raw data output. The BNO055 is 10 times the price, while still relatively low cost. The BNO055 contains sensor fusion aimed at reducing noise and increasing robustness of operation. For future reference this section will also include the kinematics of a 6-DOF rigid body; this may be found in Section 2.4.2.

2.1 Inertial Measurement Units

IMUs are sensor packages, usually comprised of an accelerometer and gyroscope [35]. IMU packages vary depending upon manufacturer in terms of the number of accelerometers and gyroscopes, and often contain barometers and magnetometers [36] [37]. Gyroscopes come in a number of forms, however in this case the prodominant form is that of the micro-electro mechanical system (MEMS) type [38] [39]. To clarify and simplify future reference to MEMS IMUs will simply refer to them as IMUs due to the ubiquity of this type and when the operating principle is different a note will be made.

It is a well known feature of IMU systems that the measurement bias of the accelerometer and gyroscope instrumentation is directly connected to the temperature of the unit [35] [39]. As such many IMUs contain thermal controller systems or at minimum a means of temperature monitoring. This temperature reading may be used with manufacturer specifications to alter the bias accordingly [40].

2.2 Noise Characterisation

As previously described, sensor readings in general, and IMUs especially are expected to contain noise. This section describes the types of noise inherent in these systems and, presents methods used to characterise and represent this noise. One method that may be employed is to look at the noise of the sensor in a statistical manner [41]. By calculating the standard deviation, mean and, therefore variance of the sensor data, an idea of the expected results can be established. The calculation of the standard deviation requires a sample that is representative of the general "population" where population refers here to all possible data output from the sensor. Conventionally the sample size may be calculated based on either the standard deviation of the population or the population size itself. As the aim of this experiment is to define the standard deviation of the population, this method of calculating the required sample size is not applicable, and since the population size is technically boundless, therefore infinite, meaning that this method is also not applicable.

For the purposes of this experiment the literature on the subject informs that the expected noise profile of an IMU of this type should be "Gaussian" [42] [43] [44]. Therefore to define an adequate sample, the data was collected until a clear normal distribution is established at minimum, and the data collection was continued for as long as is practical.

2.2.1 Noise Characterisation Methods

Much work has been conducted in both the development of methods to characterise sensors and the application of these methods. First it is logical to define the terminology used herein. Precision, is defined as the repeatability of a reading, without reference to its target [44]. This may be seen below in Figure 2.1, where 5 groups of 100 samples may be said to have been precisely placed. In this demonstration, the figures represent a 2-dimensional measurement source, with no units. The position of the groups relative to one another is irrelevant, but rather the standard deviation of each group viewed individually is the factor for assessment when considering precision. Accuracy is the metric whereby samples are assessed in relation to their closeness to a specific point not within the sample [44]. This may be seen in Figure 2.2; it may be observed that a degree of precision is inherent in accuracy, as a tight placement of points is also a feature of an accurate group, if the centroid of those points is near to the desired point. Therefore an accurately placed set of points would also be a precisely placed set of points, similar to that described in Figure 2.3.

A method of defining the accuracy of a sample is by taking the "mean" of the error of the sample, while another often employed means of defining the

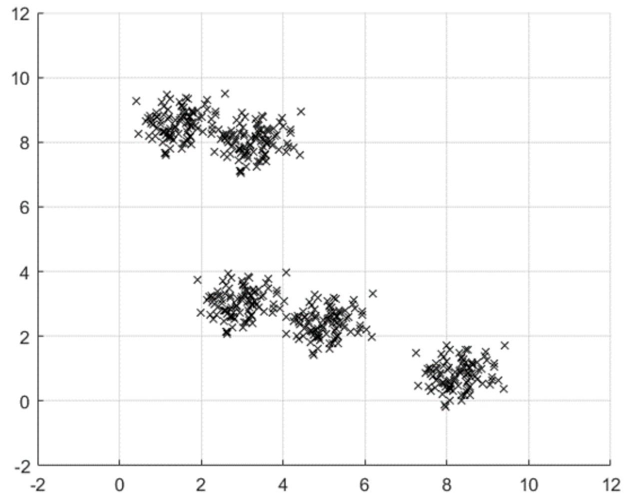


Figure 2.1: Demonstration of Precision, using dimensionless axis.

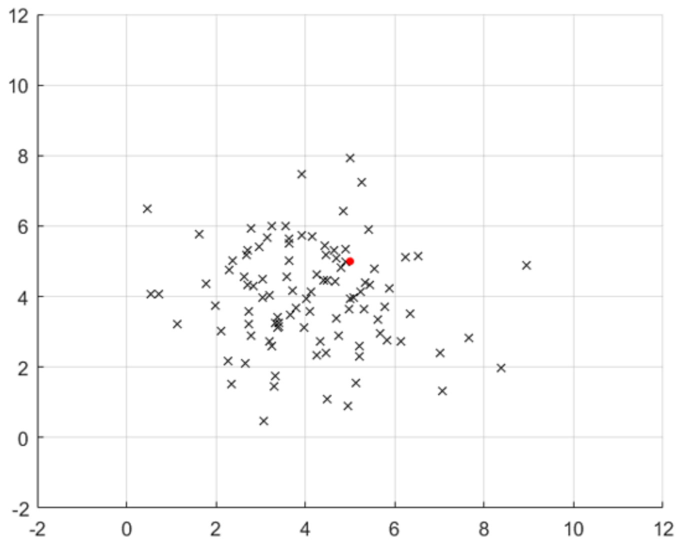


Figure 2.2: Demonstration of Accuracy, using dimensionless axis.

precision is the "standard deviation" of a sample denoted in this thesis as μ and σ respectively. For clarity the standard deviation may be calculated as shown in Equation (2.1) and the mean as shown in Equation (2.2), where N represents the sample size, and x_i the current measurement.

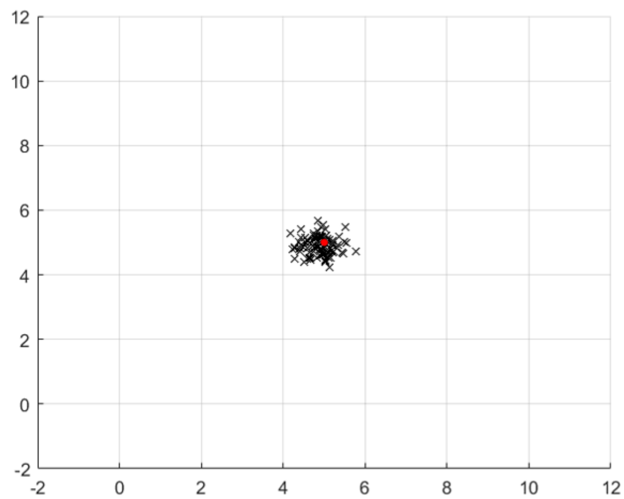


Figure 2.3: Demonstration of Accuracy and Precision, using dimensionless axis.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu)^2} \tag{2.1}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N}x_i \tag{2.2}$$

Noise in sensors is usually the product of various factors. In IMUs for example, it is the sum of multiple factors of various significance, which may be split into two forms, deterministic and stochastic [33] [45]. Deterministic errors such as those caused by manufacturing errors, misalignments and scale factor errors may be characterised as "static" [33] [45] . Stochastic error on the other hand is defined as a random noise in the measurement output, and must therefore be characterised statistically. This may be accomplished through a number of methods depending upon purpose [33] [45]. One common method is to compute the "Allan Variance" (AV) [34]. The AV is a method of characterising the noise in a signal in the time domain [46], by separating the total samples N taken over time τ_0 into overlapping clusters of time interval τ where the increment n between clusters is less than the cluster size [34].

The averages s_i of each segment is then used to calculate the AV as shown in Equation (2.3).

$$\sigma^2(\tau) = \frac{1}{(2s-1)} \sum_{i=1}^{s-1} (s_{i+1}(\tau) - s_i(\tau))^2 \tag{2.3}$$

It should be noted that methods such as finding the Power Spectrum Density of the signal to find the "colouring" of its noise are often mentioned in these characterisation systems. However, as has been found, the noise of the IMU is often close enough to "white" to require no such techniques and is therefore assumed Gaussian White Noise (GWN) [43] [42]. The main area of interest in this case is the offset or bias of the sensor output and the standard deviation and/or variance of the noise.

2.2.2 Experimental setup

The BNO055 data was collected through the use of the Adafruit Unified Sensor Drivers and BNO055 Driver libraries, running on a NVidia jetson TX1 connected to the sensor via an I2C connection. The MPU6050 was connected to the same NVidia Jetson TX1 through I2C, again using the Adafruit libraries. Neither sensor test was run while the other sensor was connected. The MPU6050 may be run in a variety of modes to alter the scale of the measurement. For this set of tests the scale sensitivity was left at the default 8192 LSB/g, and 65.5 LSB/($^{\circ}s^{-1}$) for the accelerometer and gyroscope respectively. In this case LSB stands for Least Significant Bit. The BNO055 was run in the standard mode and the fused data output was taken as the data to be used in the rigid body algorithm.

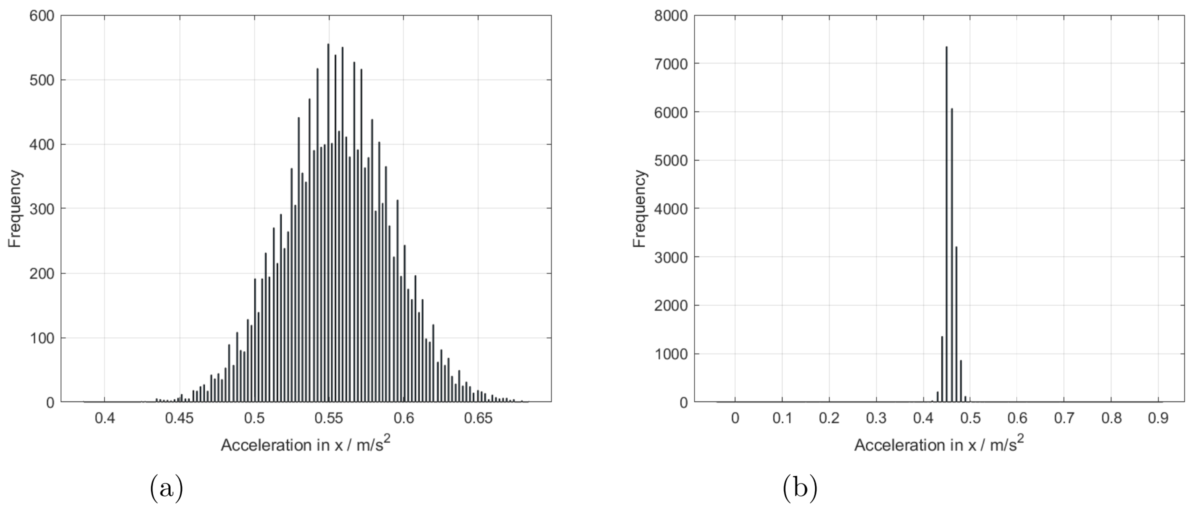
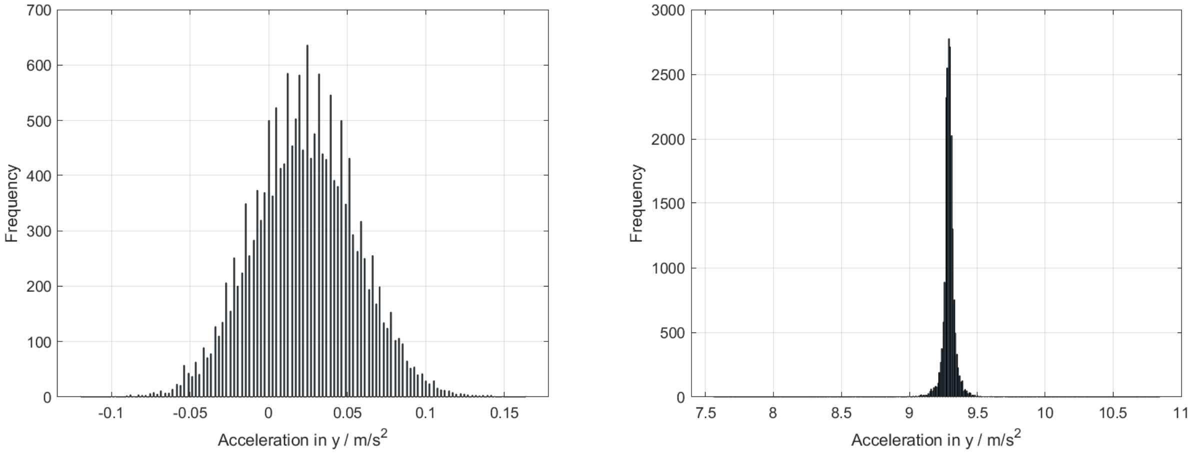


Figure 2.4: Histogram of the x axis readings of the accelerometer while stationary for (a) (MPU6050), (b) (BNO055)

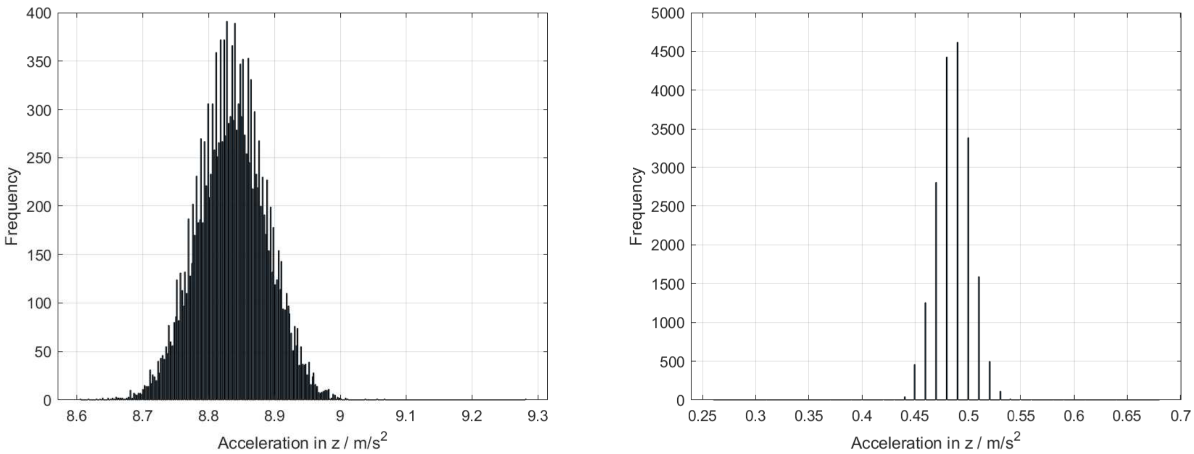
2.2.3 Results and Analysis - MPU6050

As literature on the subject suggested the gyroscope and accelerometer data from the MPU6050 was found to be of a Gaussian, white form was found [34] [45]. This may be seen when the data is plotted as histograms seen in Figures 2.4a, 2.5a and 2.6a and Figures 2.7a, 2.8a and 2.9a The x, y and z gyroscope axis showed a standard deviation of roughly 0.1949°s^{-1}, 0.2633°s^{-1} and 0.1933°s^{-1} respectively. Bias was calculated as the mean of the sample, and found to be 1.0564°s^{-1}, -1.5198°s^{-1} and 2.2818°s^{-1} for x, y and z. The accelerometer biases may not be calculated effectively as the z axis was not oriented parallel to the gravitational vector. Calculation of the IMU accelerometer bias is needed for each experimental setup in situ as each setup will come with its own misalignments. The standard deviation may however be calculated, and was found to be 0.0361ms^{-2}, 0.0326ms^{-2} and 0.0525ms^{-2} for the x, y and z respectively.



(a) (b)

Figure 2.5: Histogram of the y axis readings of the accelerometer while stationary for (a) (MPU6050), (b) (BNO055)



(a) (b)

Figure 2.6: Histogram of the z axis readings of the accelerometer while stationary for (a) (MPU6050), (b) (BNO055)

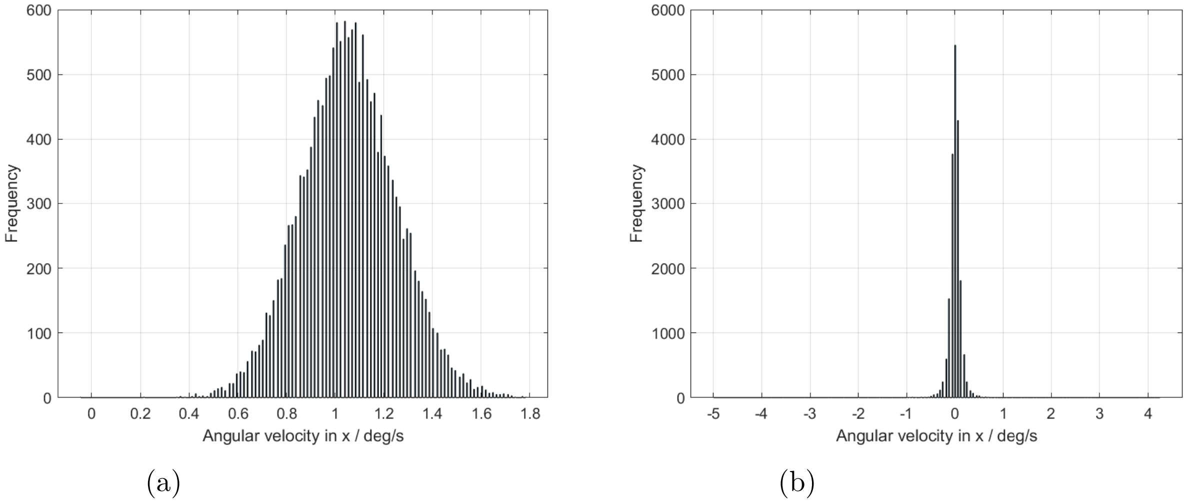


Figure 2.7: Histogram of the x axis readings of the gyroscope while stationary for (a) (MPU6050), (b) (BNO055)

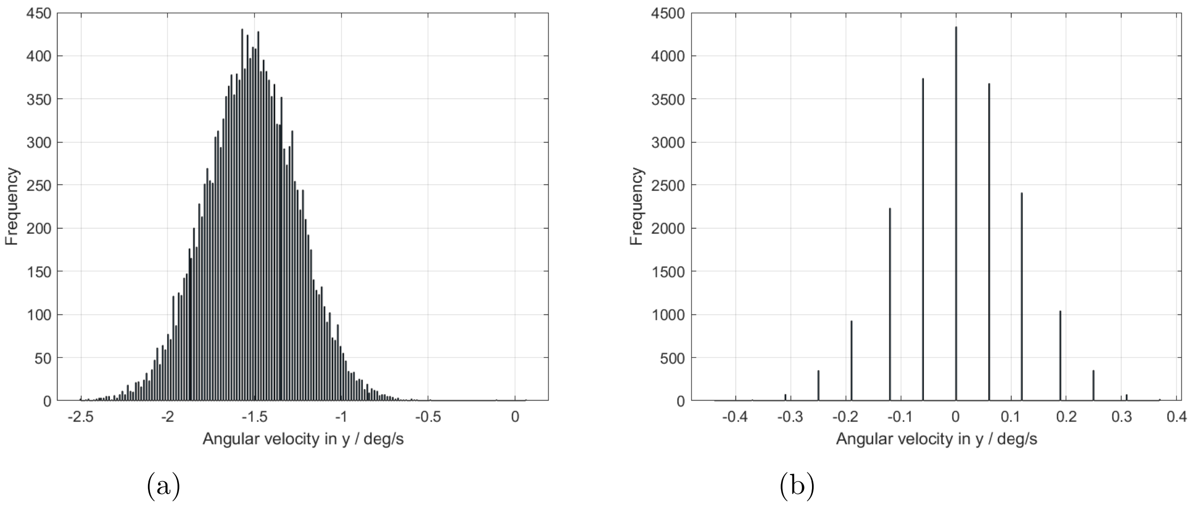
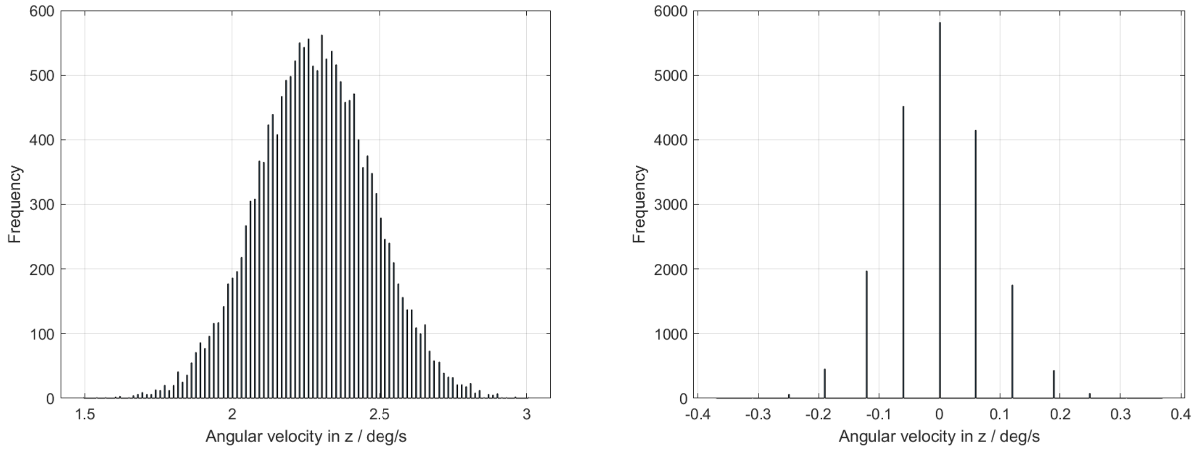


Figure 2.8: Histogram of the y axis readings of the gyroscope while stationary for (a) (MPU6050), (b) (BNO055)



(a) (b)

Figure 2.9: Histogram of the z axis readings of the gyroscope while stationary for (a) (MPU6050), (b) (BNO055)

2.2.4 Results and Analysis - BNO055

Due to the nature of the BNO055 being an IMU with pre-filtering and sensor fusion, it was possible that the noise characteristics may be non-standard; however, as may be seen in Figure 2.5b the Gaussian nature of the sensor data is maintained, while the standard deviation is unusually small for an IMU [36]. The offset in the mean value for the accelerometer in the y axis is due to the mounting orientation chosen, and the fact it is not $9.81ms^-2$ as would be expected is due to an misalignment between the gravitational vector and the axis in question. This is corroborated by the non-zero mean of the other accelerometer axis means shown in Figures 2.4b, 2.5b and 2.6b. The standard deviation for the gyroscope was found to be $0.1640°s^{-1}$, $0.1084°s^{-1}$ and, $0.0810°s^{-1}$ for the x, y and z axis. With the accelerometer standard deviation in the x, y and z axis calculated as $0.0127ms^{-2}$, $0.0652ms^{-2}$ and, $0.0163ms^{-2}$ respectively.

In comparison through the use of Allan Variance analysis it may be seen that the deviation as a function of time is again lower by an order of magnitude for the BNO055 when opposed to the MPU6050. This may be seen in Figures 2.10 to 2.12.

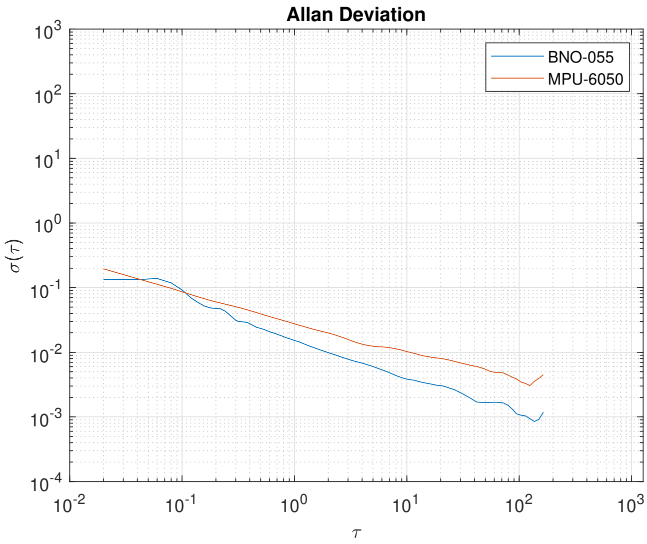


Figure 2.10: Allan deviations of x axis gyroscopes for MPU6050 and BNO055.

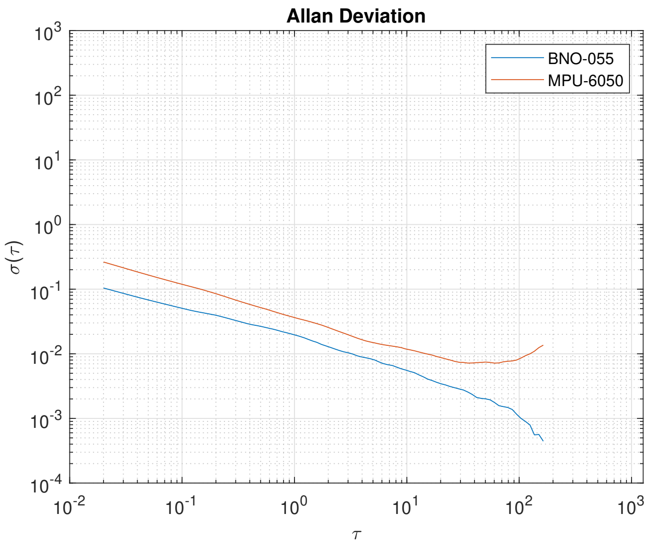


Figure 2.11: Allan deviations of y axis gyroscopes for MPU6050 and BNO055.

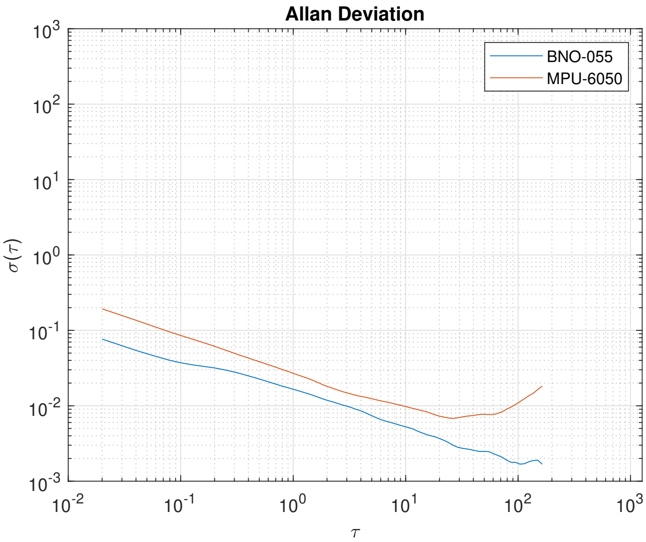


Figure 2.12: Allan deviations of z axis gyroscopes for MPU6050 and BNO055.

Axis	MPU6050	BNO055
Gyroscope - x	$0.1949°s^{-1}$	$0.1640°s^{-1}$
Gyroscope - y	$0.2633°s^{-1}$	$0.1084°s^{-1}$
Gyroscope - z	$0.1933°s^{-1}$	$0.0810°s^{-1}$
Accelerometer - x	$0.0361ms^{-2}$	$0.0127ms^{-2}$
Accelerometer - y	$0.0326ms^{-2}$	$0.0652ms^{-2}$
Accelerometer - z	$0.0525ms^{-2}$	$0.0163ms^{-2}$

Table 2.1: Results of the noise characterisation tests for the BNO055 and the MPU6050

2.3 Frames of Reference

The fusion of id and ego motion sensor measurements necessitates the use of multiple reference frames. In this case, the use of a UWBsystem necessitates a second frame of reference besides the body frame of the UAV. For clarity, the definition of reference frames are made here, before any attempt is made at the formulation of a state estimator or system model. First, a global frame is constructed, also known as the inertial frame as it will be the context for all inertia in the system [47]. As a large component of the work presented here is to be implemented on an UAV and, as there are no other demands to the contrary the usual convention of a North, East, Down (NED) reference frame is used [31]. Rotation within this frame is denoted as I, J, K for the North, East and Down axis respectively. The body frame of the UAV, and the frame upon which all id-motion sensors, such as the IMU, are aligned is denoted by x, y, z, with rotation in these axis defined by i, j, k; in all cases, $\{N, E, D\} \in \mathbb{R}^3$ and $\{x, y, z\} \in \mathbb{R}^3$

In order to translate the sensor readings taken in one frame to another, a rotation is required. This takes the form of the Direction Cosine Matrix (DCM). The "Direction Cosine" is a method for translating a vector from one frame into another, while the Direction Cosine Matrix is the method for translating the set of vectors in one frame of reference into another [31]. Therefore the DCM matrix will be of size $n \times n$, where n is the number of working dimensions considered [31].

2.4 Pure IMU Pose Estimation

To fully understand the requirement of multiple sensor inputs or Bayesian style filtering, an attempt was made at IMU-only pose estimation in 6DOF using only the kinematic equations of the system for estimation. As previously stated, the system functions in two frames of reference with a body centre reference frame from which the strapdown IMU measurements were taken, and an inertial reference frame from which all translations will be considered. The equations governing 6DOF motion tracking is described well in [31][47] and [42].

When using an IMU in a strapdown configuration, to measure the relative change in velocity and orientation, the axis by which these measurements are made must be aligned with that of the body frame of reference [42]. Where this is not the case, a separate frame reference must be defined or a transformation from the sensor axis to the body frame is required [42]. The standard method for identifying the measurement axis for IMU devices are markings on the PCB; however this is not an accurate method of aligning the sensor [42]. Error in manufacturing will lead the sensor to never be perfectly in line. To account for this, calibration is required to account for bias, scale and, angular offset [42][31].

2.4.1 IMU Calibration

As previously discussed, two forms of error in sensors exist, stochastic and deterministic [41]. The earlier section demonstrates a method for characterising the stochastic error of the sensors. This section demonstrates a method for characterising and correcting for the deterministic errors in the sensor readings for a given situation. It should be noted that this characterisation for the deterministic error is only correct for each placement of the IMU; moving the sensor will require repetition of the process. Let M be the measurement output of the sensor, either the accelerometer or gyroscope. M' denotes the corrected measurement output of the sensor. Equation (2.4) describes the procedure for calculating the measurement, corrected for deterministic error. n represents the dimensions of the sensor output. Here b is the bias of the sensor, a $n \times 1$ matrix, while T and K represent the angular offset from body frame and the scaling factors respectively; both are $n \times n$ matrices [42].

$$M' = TK(M + b) \tag{2.4}$$

$$M' = \begin{bmatrix} 1 & \Delta\psi & -\Delta\theta \\ -\Delta\psi & 1 & \Delta\phi \\ \Delta\theta & -\Delta\phi & 1 \end{bmatrix} \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix} (M + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}) \tag{2.5}$$

An expansion of Equation (2.4) may be seen in Equation (2.5) with the contents of T, K and b expanded. In the case of Equation (2.5), ψ, ϕ and θ are the Euler angles between the body frame and the IMU [42]. The unknowns in the calibration of the IMU are those elements in matrices T, K and b, for both the accelerometer and the gyroscope sensors [42].

2.4.2 6-DOF Motion Tracking for a Rigid Body

For the purposes of this system the orientation shall be represented using the quaternion system. The major difference between the quaternion and Euler orientation representations is that when using the quaternion method gimbal lock may be avoided [31]. As there exist a number of ways of ordering the components of a quaternion it is necessary to define the method used here. This may be seen in Equation (2.6), where w represents the scaler. As the IMUs are the only sensor present, the IMU itself becomes the body frame, therefore negating the need for calibration to find offset.

$$Q = [q1, q2, q3, q4] = [w, x, y, z] \tag{2.6}$$

The monitored states of the system are to be the orientation, velocity, and position relative to the inertial reference frame. The state matrix X is defined in Equation (2.7). For ease of reading column matrices such as that shown in Equation (2.7) will be written in their transpose format. The time being referred to is denoted by a subscripted k. Taking the state for example, the current time would be shown as X_k, with X_{k-1} and X_{k+1} denoting the previous next time steps respectively. The measurement inputs from the IMU are taken as i', j' and k' for the gyroscope angular velocity reading and, x'', y'' and y'' for the accelerometer acceleration readings. To account for bias inherent in IMU systems, corresponding variables are also noted as their subscripted counterparts e.g. x''_{Bias}. As seen in Section 2.3

$$X_k = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & x' & y' & z' & x & y & z \end{bmatrix}^T = \begin{bmatrix} Q & V & P \end{bmatrix}^T \tag{2.7}$$

The estimation of the next time step post measurement is as follows: First the orientation is updated based upon the measured change in heading. This requires the calculation of change in heading from angular rate, followed by the conversion from Euler to quaternion, seen in Equations (2.8) and (2.9). As the frequency of the measurement input is very small ($100Hz$) relative to the expected rate of velocities of the system, the small angle approximation may be used to simplify the computation of the Euler to quaternion conversion [48].

$$\Delta i = i'_{Meas} \times \Delta t \tag{2.8}$$

$$\Delta Q = \begin{bmatrix} 1 & \Delta i/2 & \Delta j/2 & \Delta k/2 \end{bmatrix}^T = \begin{bmatrix} \Delta q_1 & \Delta q_2 & \Delta q_3 & \Delta q_4 \end{bmatrix}^T \tag{2.9}$$

Once the change in the orientation in quaternion form has been calculated, the orientation may be updated using the quaternion product rule of the previous orientation and the change in orientation, this may be seen in

Equation (2.10).

$$Q_{k+1} = Q_k \times \Delta Q = \begin{bmatrix} q_1\Delta q_1 - q_2\Delta q_2 - q_3\Delta q_3 - q_4\Delta q_4 \\ q_1\Delta q_2 + q_2\Delta q_1 - q_2\Delta q_4 + q_2\Delta q_3 \\ q_1\Delta q_3 + q_2\Delta q_4 + q_2\Delta q_1 - q_2\Delta q_2 \\ q_1\Delta q_4 - q_2\Delta q_3 + q_2\Delta q_2 + q_2\Delta q_1 \end{bmatrix} \tag{2.10}$$

After the orientation is updated the DCM may be calculated to allow transposition of vectors in the body frame to the inertial frame [31]. This is calculated as described in Equation (2.11).

$$DCM = \begin{bmatrix} (q_1^2 + q_2^2 - q_3^2 - q_4^2) & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & (q_1^2 - q_2^2 + q_3^2 - q_4^2) & 2(q_3q_4 + q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 - q_1q_2) & (q_1^2 - q_2^2 - q_3^2 + q_4^2) \end{bmatrix} \tag{2.11}$$

Now that the relationship between the body frame and the inertial frame is defined for this time step, the accelerometer data may be used to update the velocity of the body. The velocity is then used to calculate the translation. These steps are taken in Equations (2.12) and (2.13).

$$V_{k+1} = V_k + (DCM \cdot \begin{bmatrix} x''_{Meas} \\ y''_{Meas} \\ z''_{Meas} \end{bmatrix} \Delta t) + \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} \Delta t \tag{2.12}$$

$$P_{k+1} = P_k + (V_k \Delta t) \tag{2.13}$$

2.4.3 Experimental Setup

In order to assess the functionality of both the MPU6050 and the BNO055, as 6-DOF motion tracking inputs under equal circumstances, both sets of data were run through an identical rigid body algorithm. The setup for data collection is as before stated in Section 2.2.2.

Both the BNO055 and the MPU6050 were initially tested at rest to investigate the drift in orientation and position. Next, both IMUs were moved in space facing the direction of travel, leading to minimal roll and pitch movement, and with 4 roughly 90° turns to the left and roughly 30 meters of translation, before being returned to their original location. A room layout may be seen in Figure 2.13. Finally the IMUs were pitched, to roughly 45°, then returned to level. This was then repeated for the roll and yaw axis. The purpose of these tests was to asses the drift whilst under non-static conditions, and under greater than noise level changes in readings. These tests were repeated 3 times for each IMU. Any inconsistencies are noted in the results and analysis section.

2.4.4 Results and Analysis

2.4.4.1 BNO055

Figures 2.14 to 2.16 show the estimated orientation of the IMU throughout its movement within the space. As can be seen in Figures 2.14 and 2.15, the x and y axis show little change, as the sensor was kept close to level throughout the test. The z axis can be seen to pass through 180° and return to 0° via −180°. This is due to the definition of rotational coordinates in the BNO055. The final

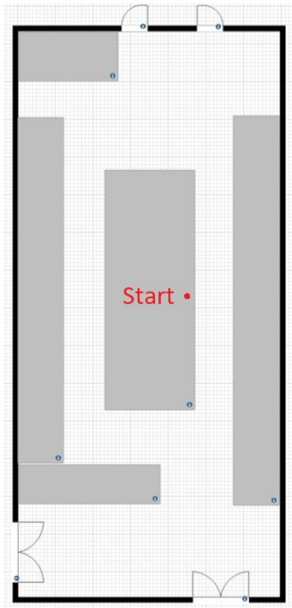


Figure 2.13: Layout of space used for IMU testing.

difference in orientation after the movement was calculated as 2.5491° in x, −2.7342° in y and 15.0092° in z. The results of the orientation tracking through change in roll, pitch and, yaw show a similar level of stability, with angular offset at the end of rotation with respect to the start location of −4.1292° in x, −3.5694° in y and, −1.9739° in z. Graphs showing orientation tracking may be seen in Figures 2.17 to 2.19.

2.4.4.2 MPU6050

In comparison to the results obtained from the BNO055, the MPU6050 motion tracking was very noisy and demonstrated a pronounced drift across the orientation estimation. This can be seen in Figures 2.26 to 2.28. The MPU6050 motion tracking also demonstrated a significant drift in the position estimate. This was significantly greater than that shown in the BNO055, by 2 to 3 orders of magnitude. It should be noted that in the course of the repetitions for this experiment 5 different MPU6050 IMUs were used. This was in light of the highly erratic readings from the z axis gyroscope; however these results were

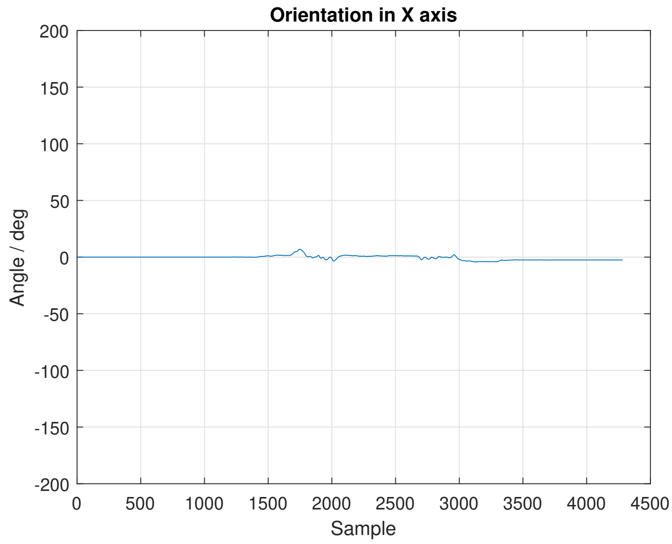


Figure 2.14: Orientation estimation in the x axis during rectangular trajectory (BNO055)

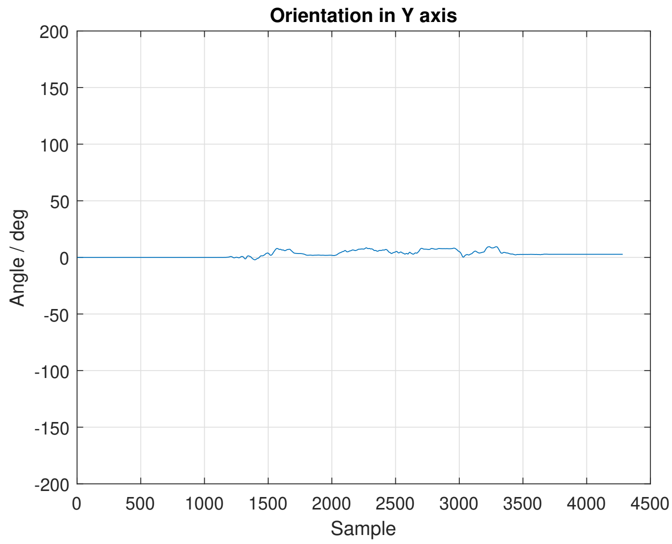


Figure 2.15: Orientation estimation in the y axis during rectangular trajectory (BNO055)

shown to be consistent.

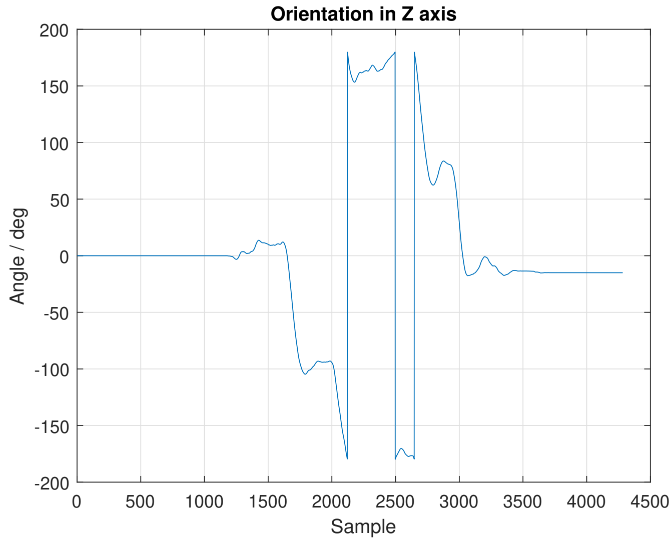


Figure 2.16: Orientation estimation in the z axis during rectangular trajectory (BNO055)

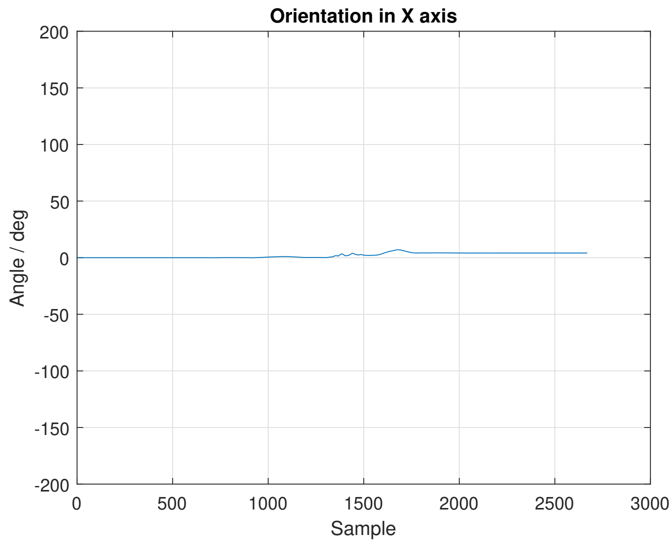


Figure 2.17: Orientation estimation in the x axis during roll, pitch and yaw rotations (BNO055)

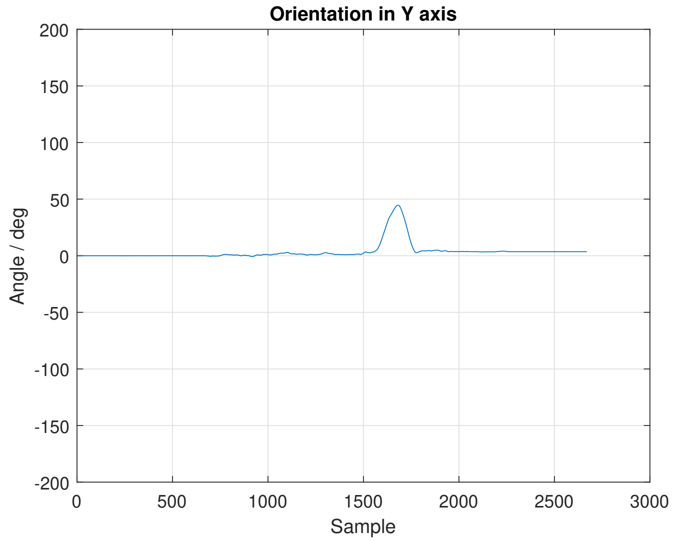


Figure 2.18: Orientation estimation in the y axis during roll, pitch and yaw rotations (BNO055)

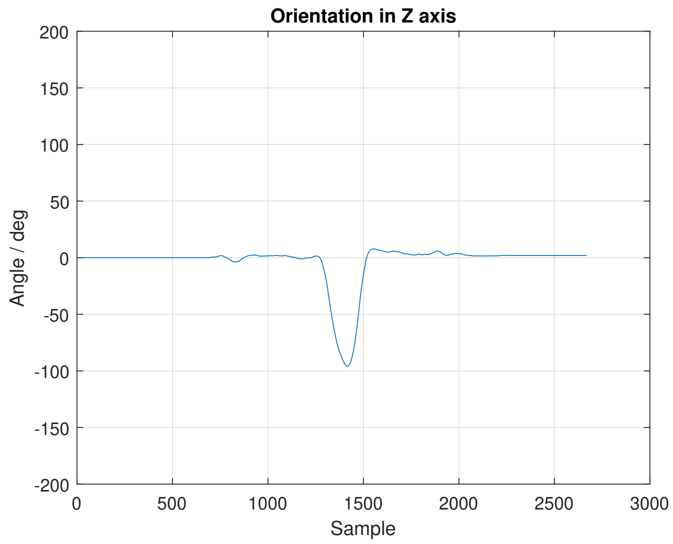


Figure 2.19: Orientation estimation in the z axis during roll, pitch and yaw rotations (BNO055)

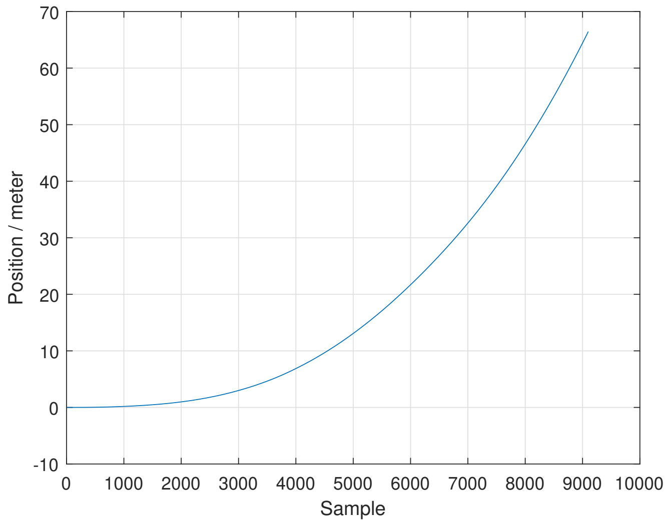


Figure 2.20: Position estimation under no motion in the x axis (BNO055)

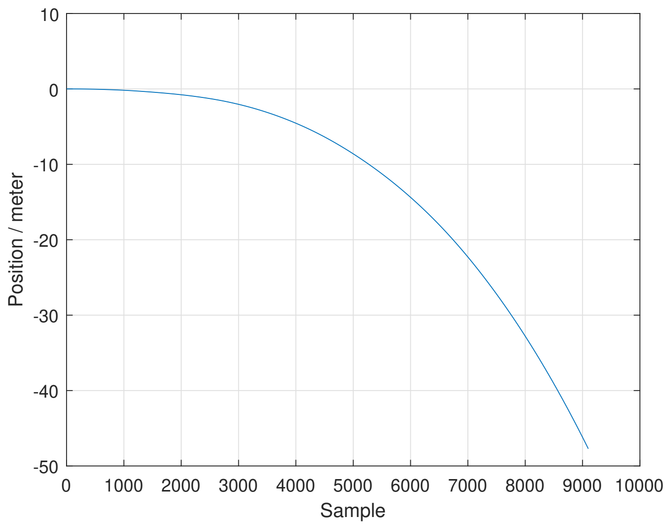


Figure 2.21: Position estimation under no motion in the y axis (BNO055)

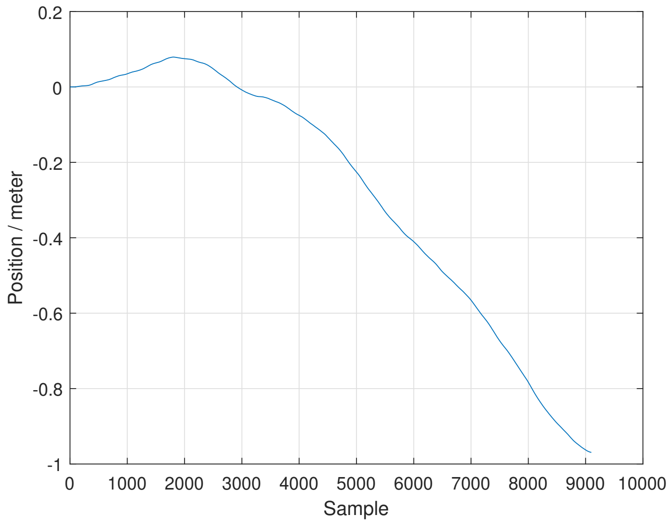


Figure 2.22: Position estimation under no motion in the z axis (BNO055)

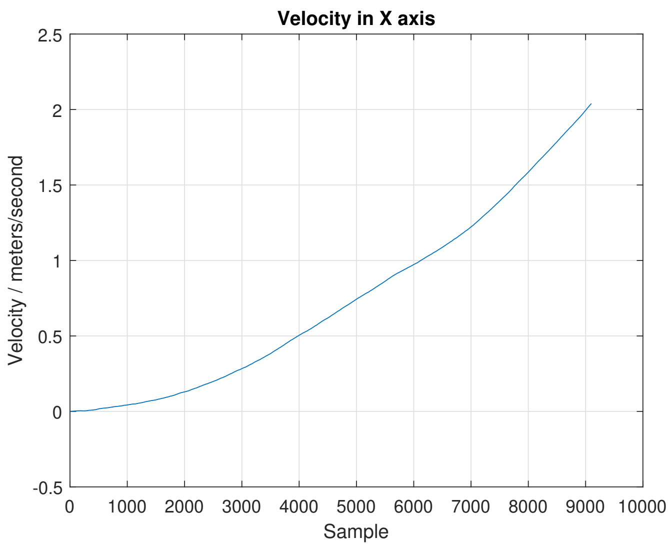


Figure 2.23: Velocity estimation under no motion in the x axis (BNO055)

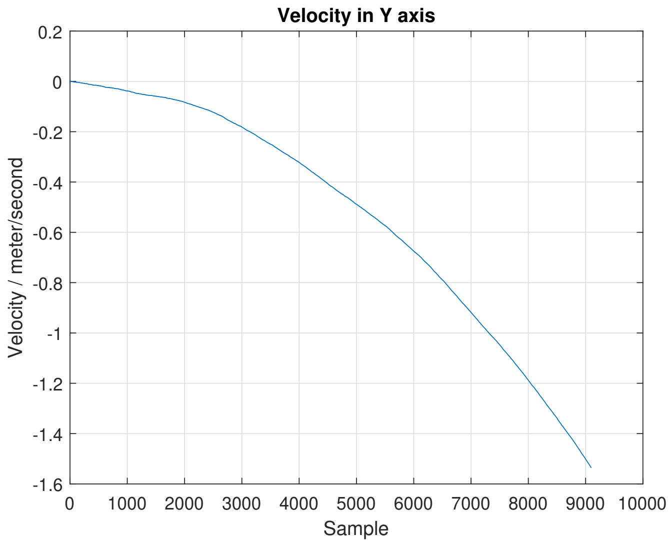


Figure 2.24: Velocity estimation under no motion in the y axis (BNO055)

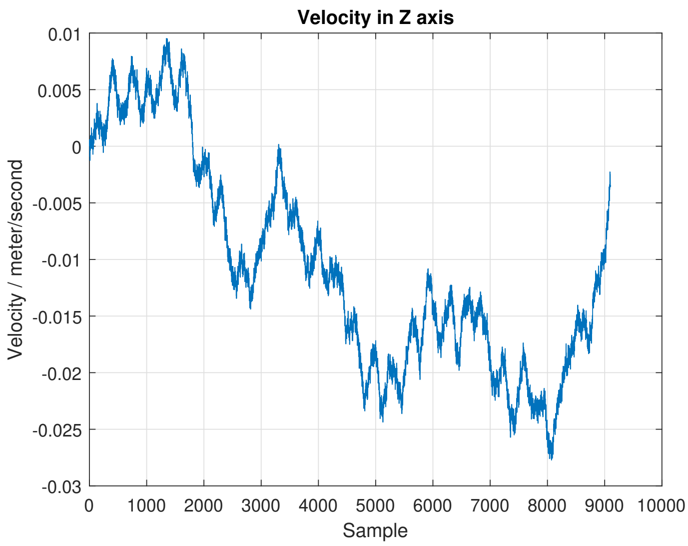


Figure 2.25: Velocity estimation under no motion in the z axis (BNO055)

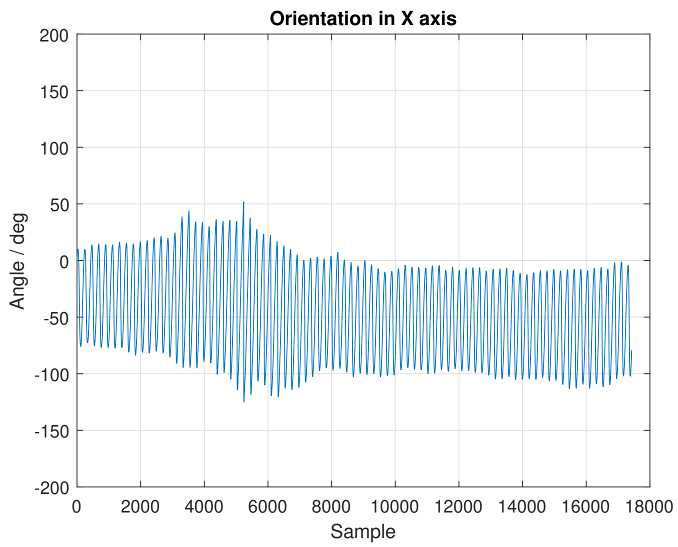


Figure 2.26: Orientation estimation under no motion in the x axis (MPU6050)

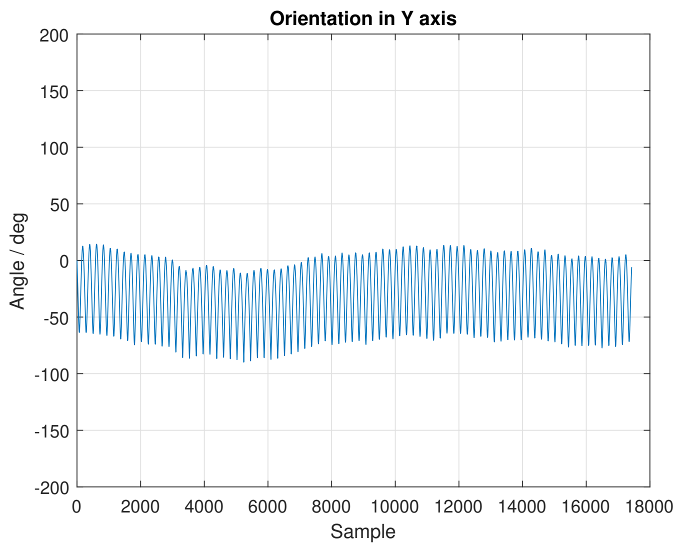


Figure 2.27: Orientation estimation under no motion in the y axis (MPU6050)

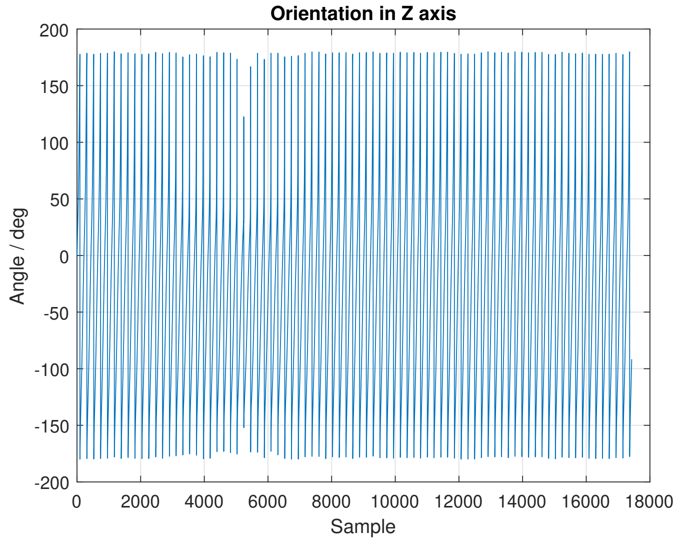


Figure 2.28: Orientation estimation under no motion in the z axis (MPU6050)

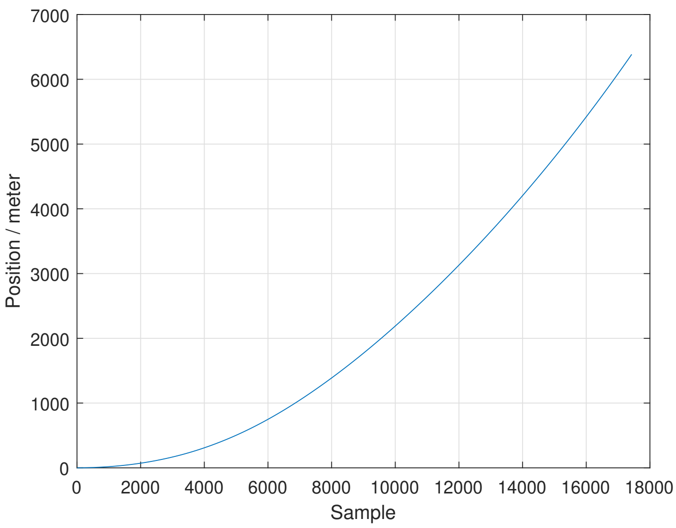


Figure 2.29: Position estimation under no motion in the x axis (MPU6050)

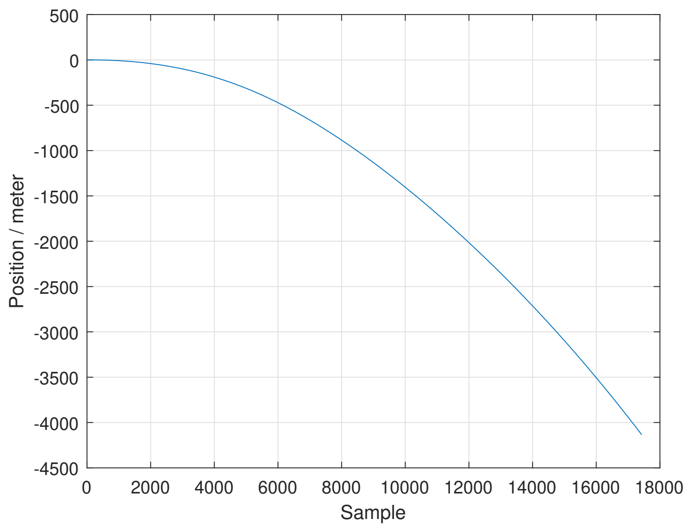


Figure 2.30: Position estimation under no motion in the y axis (MPU6050)

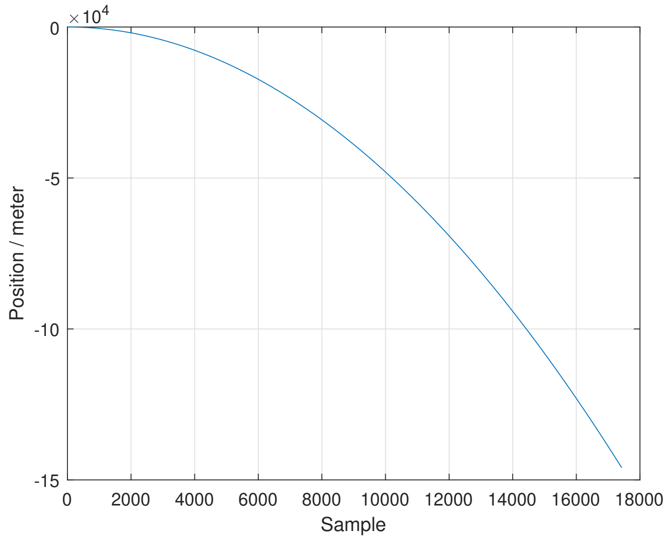


Figure 2.31: Position estimation under no motion in the z axis (MPU6050)

2.5 Conclusions

As expected the BNO055 proved reliable in tracking the orientation throughout the experiment; however the positional tracking was highly prone to drift. From this the necessity of other measurement sources for the x, y and z dimensions is clear. The increased error in the positional tracking as opposed to that seen in the orientation track is due to the order of equation used to calculate it. For determining the orientation, the measured angular velocity requires only a single integration to translate into angular change. However, the positional update came from the acceleration of the sensor, therefore a double integration is required, meaning error is accumulated not in a linear fashion as in the gyroscope data, but in a quadratic fashion. This quadratic drift can be seen best when attempting to track the position of the sensor when placed at rest in Figures 2.20 to 2.22. The velocity estimation of this method displayed significantly less drift to at least one order of magnitude, this may be seen in Figures 2.23 to 2.25. Without further compensation from outside measurement sources, it is clear that although improvements in IMU technology and manufacturing may reduce positional drift, they will not eliminate it. Therefore, for meaningful positional tracking other sensors are needed and a method of fusing data from the various sources is necessary.

The use of a BNO055 in future work would allow for the implementation of an "Attitude Heading Reference System" (AHRS) more quickly and robustly. If the MPU6050 was used the AHRS would require the implementation of other more complex processes than those demonstrated in this chapter.

An important point to be made is that the process of prediction used here does not provide an indication of the estimation performance. Although the system may operate effectively in many conditions, it is not guaranteed to be robust, therefore under BVLOS conditions, the estimate may not be obviously incorrect until too late. It is also clear that without a pre-fusion system such as used in the BNO055 the reading bias is not constant, therefore for future work the effect of an adaptive or model based bias should be considered.

Chapter 3

3 Degrees of Freedom Motion Estimation For an Unmanned Ground Rover

As shown in the previous chapter, the orientation of an agent may be found through the use of an IMU alone. However, due to the measured states, this is not reliably possible for its position [42, 43]. One solution to this, is the introduction of other sensors. The main challenge with this method is the robust fusion of this data. Traditionally one solution is state estimation. State estimation is a method by which complete or partial data in the form of sensor measurements may be used to estimate the states of a system. In this case, the states of interest are the pose components of the agent x, y, z and θ, ϕ, γ.

This chapter describes the process by which such a system may be constructed; the results of experimentation, and presents the next stages in the development of this work. To allow for robust, repeatable investigation of an localisation system it was decided that the first step in development should be in a constrained system. The development of a state estimation system for a

UAV in an indoor setting is not broadly novel [49, 50, 51]. This method has been used successfully in many works to great effect [49, 50, 51]. However, these methods often rely upon specialised system models, and sensors, an approach not conducive to transfer between platforms [49, 50, 52, 53]. The principle aim of this project is to attempt to construct a more holistic localisation system for agents in an unknown, BVLOS or Non-Clear Line of Sight (NCLOS) space. The main route chosen for investigation towards this, is the improved characterisation of sensor uncertainty. While the long term goal is the formulation of an algorithm capable of estimating the full 6-DOF pose elements, this would introduce many factors which cross-couple and thus obfuscate the effects of the changes being made. It is also reasonable to test the components of a platform in a stable, and more mathematically simple system, where irregularities and unexpected results may be more clearly analysed. This process allows for a better understanding of the system and therefore a more reliable interrogation of the system variables post experiment. For this purpose a skid steer Unmanned Ground Vehicle (UGV) was chosen. The system is assumed to move only in two dimensions (x, y), bound to 3 degrees of freedom (x, y, ϕ).

As the first stage in the development of the indoor state estimation system, a supplementary sensor is to be selected, to enable x, y localisation and provide the update measurement for the system. GNSS are not available as the environment is indoors, or assumed to be too agglomerated to allow for regular GNSS operation. As stated in the introduction to this work, the focus of this project is the investigation of improved sensor characterisation and the viability of the use of sensor variance models. This chapter focuses on the construction and testing of a system that may be used towards this aim, and the selection of a suitable sensor for this approach. Therefore, the sensor chosen must be:

- Low cost (less than or approximately equal to £500).

- Capable of operating in NCLOS conditions.
- Self contained, i.e. not rely upon systems or connections to networks beyond the control of the operators.
- Capable of measuring the position of the system to a resolution on the same order of magnitude or less than the scale of the platform (0.6 m across).

The sensor is to be low cost to enable more viable implementation. The reasoning behind the requirement for NCLOS operation is that in the intended environment there is no guarantee of clear lines of sight, this is also applicable for the final requirement. If a UAV is deployed in a complex environment under NCLOS or BVLOS conditions collision avoidance is required, localisation of large granularity would prove ineffective or very limiting. The requirement for a self contained system is to ensure the analysis of factors which may alter the reliability is not impeded. For GNSS systems the accuracy and reliability are linked to factors beyond the control of the operator, such as local and solar weather, satellite positioning and their maintenance. This would lead to experimental variables not within the control of this project, and it is deemed preferable to avoid this.

3.1 State Estimation

This section describes the mathematical foundation behind the state estimation filters employed today, and those in use in later chapters. The definition will begin with a brief summary of joint and conditional probability; moving then onto the recursive filters built upon this as a foundation such as the Kalman Filter. Finally non-linear filters and sensor fusion technology is discussed with an overview of the state of the art in state estimation.

While the Kalman Filter is a widely adopted and well understood, it is useful for clarity and thoroughness to describe its outline and formulation in this section of the work. In this context "state estimation" refers to the calculation of unmeasured, or unreliably measured states through the use of observations. In a theoretical context, state estimation is typically employed as a means of inference for unmeasured states through observational models. The uncertainty in an observation may be transferred to the uncertainty of the inference through conditional probability. In simple terms, the statement of "if this then that" may be employed.

If two events X and Y are considered, the probability of both occurring is given by Equation (3.1).

$$P(X, Y) = P(X)P(Y) \tag{3.1}$$

From this, the probability of event X given event Y is given by Equation (3.2).

$$P(X|Y) = P(X, Y)P(Y) \tag{3.2}$$

If these events are independent then this probability is given by Equation (3.3).

$$P(X|Y) = P(X) \tag{3.3}$$

In many situations, such as the roll of a die the probability is said to be discrete, in that the results are an integer value between 1 and 6 [44]. In many robotics applications the states are taken to be continuous [44]. For example the position of a 1D actuator may be defined over a continuum of points [41]. These probabilities are generally assumed to be of Gaussian or Normal distribution, where the mean represents the most likely value for the state [41, 44]. State estimation may therefore be considered to be the estimation of the mean probability of a state [41] [44]. As best described in Thruns' work "Probabilistic Robotics" the State estimators used here are developed from the Probabilistic Generative Laws as shown in Equation (3.4)[41]. In this case, it may be defined that the probability of the state of interest (x_t) is conditioned by all past states and measurements. Here, z_t and u_t represent the measurement and control inputs respectively; these variables will be defined in more detail later [41, 44].

$$P\left(x_t | x_{0:t-1}, z_{1:t-1}, u_{1:t}\right) \tag{3.4}$$

3.1.1 Kalman Filters

Initially described by Kalman in "A new approach to linear filtering and prediction problems" published in 1960 the Kalman Filter is built upon the work by Bode, Shannon, Wiener and many others [54] [55].

The KF is comprised of four main stages. Firstly, the prediction or estimate stage, whereby the states of the system are predicted based upon the known system model as seen in Equation (3.5). In this case, the system model is a generic function represented by f, which calculates the predicted states X_p from the past states X_{k-1}.

$$X_p = f(X_{k-1}) \tag{3.5}$$

Secondly, the Kalman gain calculation stage, where the reliability of the prediction is compared to that of the observation as seen in Equation (3.6). In this case, P_{Pred} is the prediction covariance matrix, P_{Sen} is the sensor variance of the observation and K is the Kalman gain. Something to note at this point is that the Kalman gain is a direct indication of the difference in reliability between the sensor observation and the prediction of the state. Figure 3.1 demonstrates how this relationship translates from a reliable prediction in relation to the observation to an observation that is reliable in relation to the prediction.

$$K = \frac{P_{Pred}}{(P_{Pred} + P_{Sen})} \tag{3.6}$$

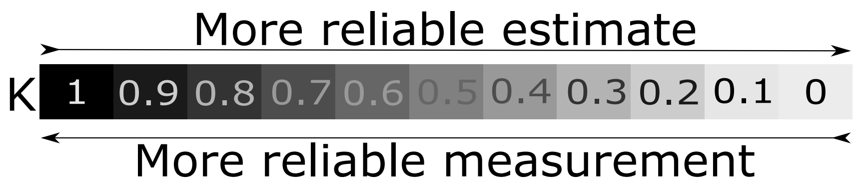


Figure 3.1: Kalman gain scale and its relationship to estimate and measurement reliability.

Third, the fusion stage, whereby the Kalman gain is used to combine the prediction and observations of the system states. This is seen in Equation (3.7) with the observation represented by Y, and the updated state as X_{k+1}.

$$X_{k+1} = Xp + K(Y - Xp) \tag{3.7}$$

Finally, the state uncertainty is updated and the process repeats. The equation used to update the process covariance is seen below in Equation (3.8). In this case I is an identity matrix, and the previous process covariance is represented by P_{k-1}. Here, Q represents the process noise, or the constant rate of uncertainty inherent in the model of the system.

$$P_{k+1} = (I - K)P_{k-1} + Q \tag{3.8}$$

An important point to note is the title of Kalman's seminal work, and the inclusion of the "linear" requirement. For Bayes' law to function, the assumption of Gaussian White Noise (GWN) is required. Additionally, to model systems with non-linear components modification to the above formulation is required.

In practice, this may also be demonstrated as in the flowchart seen in Figure 3.2. In this case, the variables represented by A, B, C, and H are matrices used to condition the inputs between steps.

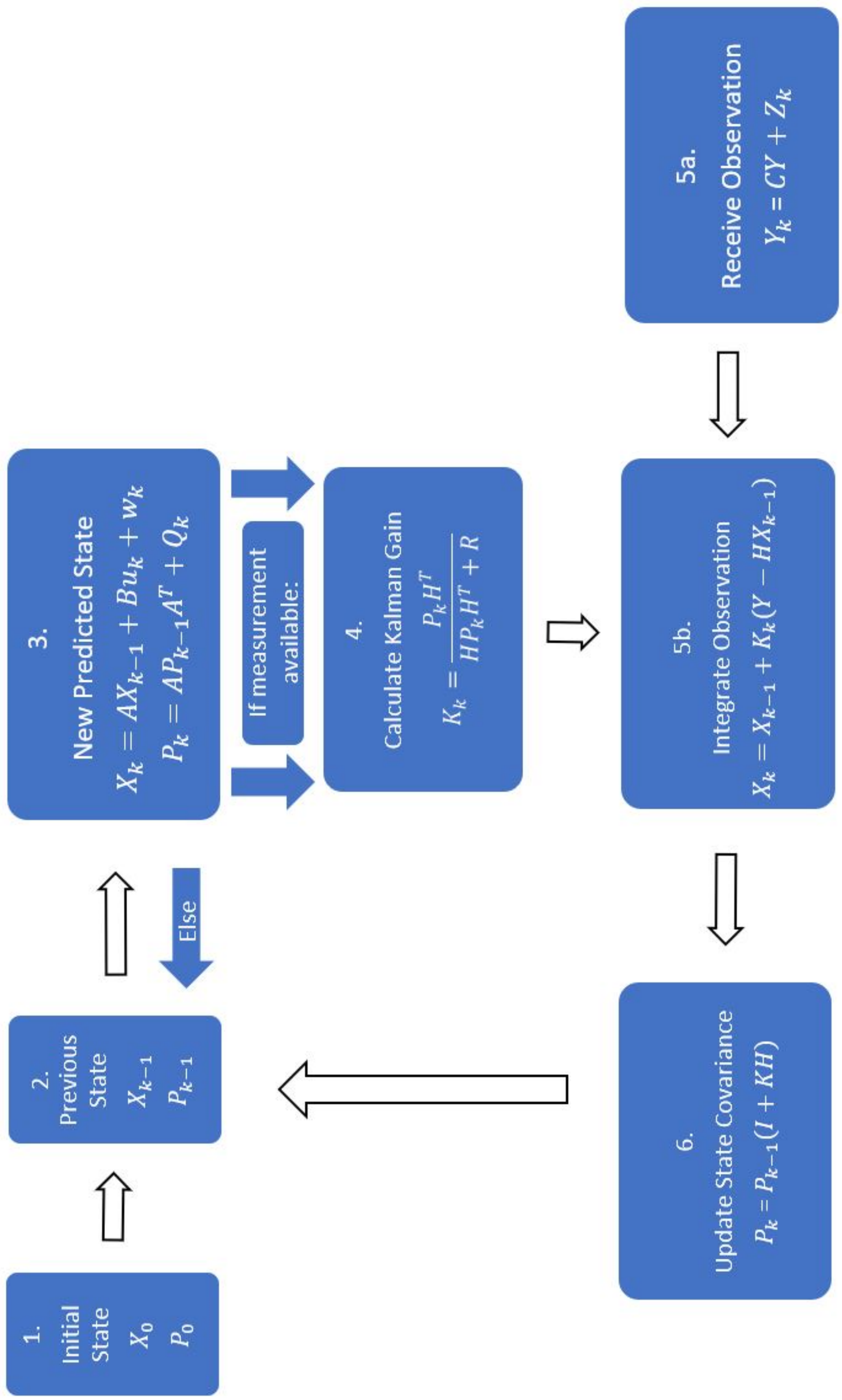


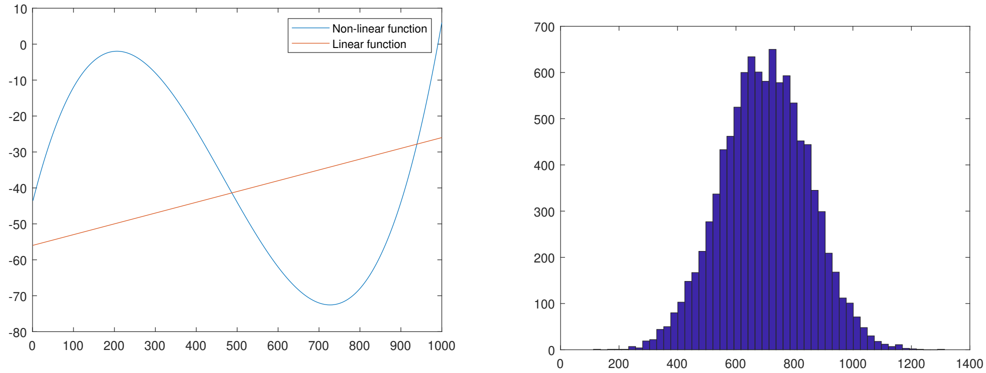
Figure 3.2: Flowchart describing the discussed Kalman Filter algorithm.

3.1.2 Extended Kalman Filters

Where the KF assumes linearity of the system and measurement models the Extended Kalman Filter (EKF) may be used for systems with some non-linearity [44]. An EKF is a state estimator, built upon the KF, with modifications made to accommodate for non-linearities in a system, or observation model. These modifications may also accommodate for noise that is not perfectly Gaussian. Non-linear functions may be used to describe the system or observation or the observation data itself [44].

EKFs have been shown to function sufficiently well in systems whose non-linearity is not great [41, 56]. A linear system may be defined as one which may be scaled and obeys superposition [57]. A system may be described as non-linear and thus require more complex methodologies if any one component of the system does not adhere to this linear requirement used in the KF. This is due to the impact of a non-linear transformation upon the Gaussian distributions assumed by the Kalman Filter [58]. The degree to which the system is non-linear determines the type of filter used [44]. In this case, or in the cases of a non-linear system model an EKF may be applicable. First consider the effect of a non-linear function upon input Gaussian data to demonstrate the reasons why it is inadvisable to use a linear estimator without adequate accommodation for non-linear models.

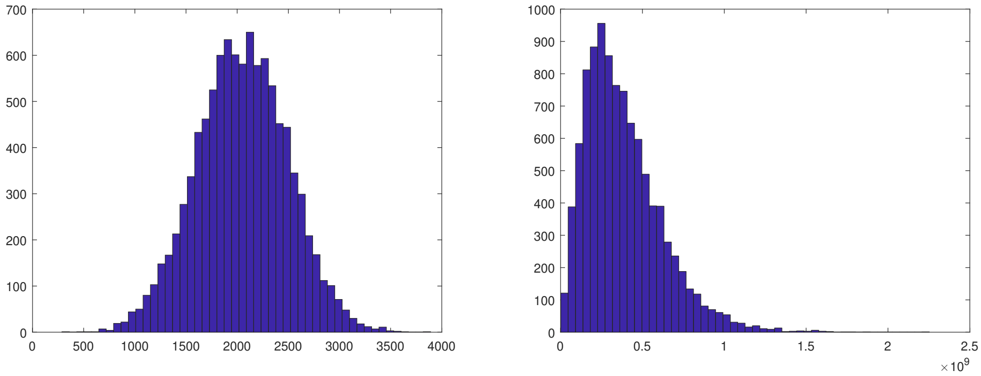
To fully explain the EKF , first a linear function $f(t)$ as seen in Figure 3.3a is considered. Next, if data as seen in Figure 3.3b of a GWN form, is passed through this function, the result is Gaussian which has been transformed in a linear manner, as seen in Figure 3.4a. To continue, a second function of non-linear form as seen in Figure 3.3a may be constructed. If again the same GWN data is passed through this function, it may be seen that the output data is no longer Gaussian, as seen in Figure 3.4b. It is important to note that as the data is transformed in a non-linear manner, the mean of the data affects the way the function transforms the data.



(a) Example linear and non-linear functions.

(b) Input data of Gaussian form.

Figure 3.3: Exemplar functions and Gaussian data.



(a) Input data passed through linear function.

(b) Input data passed through non-linear function.

Figure 3.4: Resultant data post operations.

To accommodate for the non-linearities in a system or observation model the EKF utilises a Taylor series expansion to approximate a linear function [44]. This allows for systems with some non-linearity, which would not be suitable for the KF, to be modelled through the use of an EKF [44]. A drawback of this methodology is that if a system is highly non-linear the Taylor series approximation will violently change its output over a narrow band of locations

[41]. Therefore the rate of change in the gradient of the function must be low in comparison to the standard deviation of the Gaussian being passed through it [41].

In terms of the EKF algorithm, the changes made to deal with the non-linearities may be seen in the components that deal with the probabilities associated with the system. In this case, the predicted state uncertainty is calculated as shown in Equation (3.10)[59], where G_x is the Jacobian of the system model, as described in Equation (3.9)[59].

$$G_x = \frac{\partial f}{\partial x}(X_k) \tag{3.9}$$

$$P_p = G_x P_{k-1} G_x^T + Q \tag{3.10}$$

When an observation is made the overall state uncertainty is updated as shown in Equation (3.11) [59]. Here, H_{z_k} is the Jacobian of the observation model, as seen in Equation (3.12)[59].

$$P_k = (I - KH_z)P_p \tag{3.11}$$

$$H_z = \frac{\partial h}{\partial x}(X_k) \tag{3.12}$$

Finally the Kalman gain is calculated as shown in Equation (3.13)[59].

$$K_k = P_k H_z^T (H_z P_k H_z^T + P_{Sen})^{-1} \tag{3.13}$$

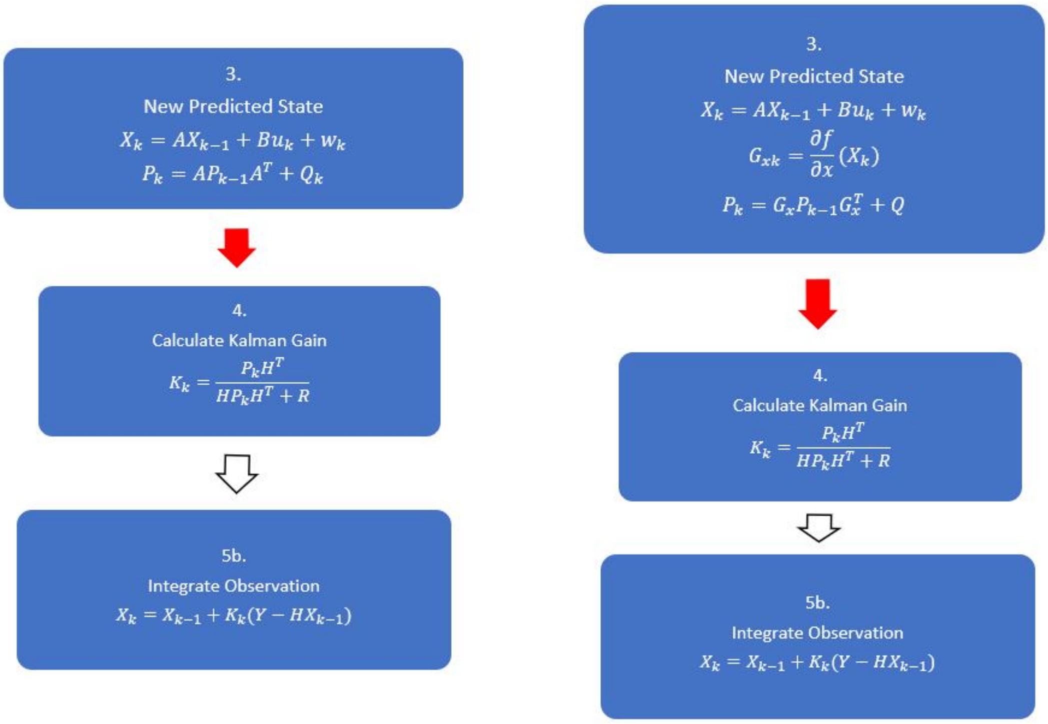


Figure 3.5: Flowchart highlighting the differing stages between the KF and EKF.

3.1.3 Other Non-linear System Filters and techniques

In many cases, a state estimator is used where more than one observation source is present [60, 61, 51]. In these applications, the observation sources often operate at different frequencies. For example, IMUs operate typically at higher update rates than other localisation sensors [50, 62, 63]. In these cases, a "multi-rate" state estimator or an estimator with multiple update phases may be used [64, 65]. An alternate method of dealing with sensors of different update rates is the use of "control inputs" [41]. In the cases described previously the "control input" u is constant and thus may be disregarded in the prediction phase as no change in the states is measured or defined as the only observations made are dealt with in the update phase [41]. However, if a sensor is present which operates at a high enough rate to be compatible with the prediction phase of the state estimator then this observation may be used as the control input [41] [47]. For example, the prediction of the system states is not made based upon the last acceleration measured, but rather the latest measurement of the IMU. In cases where the only sensor input present provides observations in x, y, z the acceleration is always assumed as zero in the prediction phase. As gravity is present, this is not the case. Therefore the use of the control input would improve the state estimation [41]. As described in the previous chapter, estimation of position based solely upon IMU measurements is destined to drift in a quadratic manner, therefore constraining this drift by the update phase is desirable.

For systems where the EKF is not appropriate other tools such as the Unscented Kalman Filter (UKF) or "Particle Filter" may be applied [41]. The "Unscented"[1] transform refers to the manner in which the UKF linearises the system. The UKF functions differently from the EKF in that it does not linearise the system or measurement models through approximation of these functions. Instead it uses the characteristics of the Gaussian data.

3.2 State of the Art

As stated previously, the function of this chapter is the selection of a supplementary sensor which will provide direct measurements of the x, y and z states of the platform. As unmanned system technology has matured there has been an increased focus on alternatives to GNSS systems for localisation. This section will discuss the options currently available and research into their use. It will also discuss the state estimation strategies investigated and means by which they can be assessed.

[1] An interesting note is the "Unscented" filter supposedly gained its name from a group of researchers who thought the EKF "stunk", and therefore constructed a filter with an "Unscented" transform as an alternative.

3.2.1 State Estimation and Localisation for UGVs

When presented with the requirement to operate any remote device and collect data, positioning is a key challenge [66, 67, 68]. Much work has been done on the subject of localisation of platforms [69, 70, 41]. As a UGV system is in constant contact with a surface upon which it moves, odometry naturally forms a key component in many approaches. The work presented by Marron et al. demonstrates a method of fusing odometry data with vision measurements of the space [71]. The vision system comprises the update component of the EKF implemented in this work, with the odometry taking the role of the dead reckoning "prediction" phase input [71]. This approach was shown to be successful, however the vision system used here relies upon markers placed throughout the space [71]. Although this is not practical, especially in uncontrolled environments, lighting is needed to allow for recognition of these targets [72]. This general approach has been favoured in the past due to its reduced computation needs, however more modern systems are capable of visual odometry without the need for placed markers [73]. The use of wheel odometry is a common tool in much of the work focusing on UGV localisation [74, 75].

As in other works, the use of an IMU alone to estimate the position of an agent is not practical for a number of reasons already shown in this work [42, 43]. Many of the solutions to this involve the integration of supplementary sensors, in the case of the work done by Darmstadt and Aachen universities this took the form of a magnetic field sensing [76] [77]. This technology presented an interesting solution to the issues of obstacles and signal absorption / reflection found in UWB or Wireless Local Area Network (WLAN) systems. In this case, the system performed well under the conditions applied, however the presence of Brushless Direct Durrect (BLDC) motors and the magnetic fields included demonstrate an issue for implementation for multirotors. The system presented showed a clear smoothing of the positional estimate and aligned the estimate more closely to the real movement of the system, however, again the ground truth of this system was sparse, taking the form of points in space on the order of metres apart and no verification as to their reliability.

The work presented by the Southeast and Shandong universities further demonstrated the advantages of the inclusion of UWB systems to IMU based position estimation through the use of an EKF [78]. The work clearly shows the noisy characteristics of UWB systems alone, and how this may skew results when included in standard EKFs. The work presented a distributed method for the computation of position for human tracking.

As shown by other works,UWB systems are generally noisy and often contain significant outliers [79, 80, 56]. There is an large body of work on the subject of the causes of the noise in UWB systems, and outliers therein. The work presented by Henan University of Science and Technology demonstrates a method of post-processing by which these outliers may be accounted for [81]. In this work an outlier detection component to their filtering and fusion algorithm designed to deal with large errors from the UWB system based on the innovation orthogonal criterion is presented. This system is shown to be more effective in practice than an Error State Kalman Filter (ESKF), as it is more robust to large outliers which although tolerant to, the ESKF will be affected by. Although this paper presents a reference trajectory it does not show a method by which this is monitored.

The work presented by Zhang et al. describes the implementation of UWB systems into a linear Bayesian filter for indoor localisation [82]. As is standard practice, this system utilises wheel encoder driven odometry. The UWB system is based around the DWM1000 chip. A VICON system provides the ground truth, allowing for a good assessment of performance, however restricts the testing space volume. An interesting component of this work is the use of a linear Bayesian filter. In many of the other studies on the subject, the non-linearities of the system are deemed too great to allow for linear filters. It is due to the fact that the platform used operates closely to the skid steer model, however there will still be introduced noise due to this approximation.

Segura et al. conducted work into improving the functionality of NLOS operation for UWB systems in indoor environments [83]. A key point made in this work is the effects environmental conditions have upon the UWB system. Although the UWB system is considered more reliable than other systems such as WiFi or ZigBee, there are several circumstances that lead to drastic decrease in performance. In the case of this paper, indoor walls are the key focus, due to the possibility of multipath error, however the results are based on simulation only.

The work presented by Lima et al. shows an implementation of a UWB system using a differential drive rover [84]. The system utilised a standard KF, which although capable of fusing the data from onboard wheel encoders proved unsuited to the task. The results showed significant noise prior to the filter implementation, and although improved, the non-linearity of the system introduced significant unmodelled components, which may be to blame for the unreliable position estimate.

Li et al. presents work on a method for implementing a UWB sensor fusion system through the use of an EKF [85]. This work focuses on pedestrian agents in frequently obscured environments leading to NLOS conditions. As is often found, the ground truth used for the position of the platform is not reliable; however the system focuses more upon the steps taken, which may be easily counted. Here the UWB system serves to reduce drift encountered due to the dead reckoning approach used for the pedestrian tracking scheme. An EKF is used to fuse the sensor data and, is found to deal sufficiently with the non-linearities of the system and sensor measurement model. The heading of the system is computed separately from the position element through the use of a Madgwick filter.

Work has been conducted in the field of IMU UWB fusion EKF s for UAV s by Guo et al.[56]. One approach utilised the sensors from a Pixhawk 1 flight controller, and the P410 UWB system [56]. Linear Regression or LR is combined with an EKF for the position estimation, with the ground truth provided using a VICON system [56]. The LR method is also applied to finding the locations of the anchors in the space [56]. Their work also demonstrates scenarios where the UKF is non-ideal for the UWB system when compared to the EKF [56]. The research presented demonstrates that the UKF unscented transform and the placement of sigma points are liable to error in the case of noisy data [56]. This is expected to be the case with the UWB , especially in cases of non-ideal anchor placement and cluttered environments [56, 86].

Much work has been done on new methods for tuning a state estimation system [87, 88]. In the construction of such a system there are several unknown components, particularly the "process noise" and the "sensor variance". Traditionally the sensor variance is measured empirically, then after implementation this value is modified to better represent the variance of the sensor in the testing environment. Better characterisation of the sensor variance allows for this tuning process to be minimised or omitted entirely. The tuning of the process noise is more difficult and complex. The process noise of the system is taken as the errors caused by non-linearities in the system, uncertainty due to methods used to linearise the system, factors not modelled, and may also include the variance of the input sensors. The process noise may be minimised through a more complete model of the system, however this is often not possible or would lead to increased computation time. Better linearization of the system will also reduce the process noise. Again this is not always possible and normally increases computation complexity. These possible methods of minimising the process noise will also not aid in its calculation, however they will reduce the complexity. If an input is used in the prediction phase, its variance will be a component, therefore better characterisation will provide a starting point from which the tuning may begin.

Not all work in the field of mobile robotic systems using UWB localisation utilise an EKF . The study conducted by Gonzalez et al. proposes a Particle Filter approach [89]. The proposed system is intended for a 3 Degrees of Freedom (3-DOF) system as with other work, with a state made up of x, y and theta. The work makes the point that the EKF and UKFs assume that the errors in the system are Gaussian. The authors state that the range based UWB system will not adhere to this criteria, however other work shows evidence to the contrary [56, 86, 90]. This is the principle reason for the use of the Particle or Monte-Carlo Bayesian Filter. Although the principle reason for the use of the Particle Filter is debatable, the work presents interesting solutions to the inherent issues with multilateration of several uncertain range measurements. These solutions are fundamental to Particle Filter design; however, they are highly computationally intensive meaning online computation is infeasible. This is because a Particle Filter deals with the non-linearities of the system through the placement of "particles" over a distribution to approximate it. The more non-linear and less Gaussian the system is, the more particles are needed. As stated by Thrun in his work "Probabilistic Robotics" the computation time of such a filter is often exponential to the number of states assessed [41].

3.2.2 Ground Truth Systems

Some work has been conducted into the use of the RTS as ground truth systems.

Dabove et al. utilised a Total Station (TS) system for the initial calibration of the anchor positions and the angle measurement features of the TS [91]. This provided an excellent ground truth metric for the IMU system onboard the UWB tag [91]. However, there was no ground truth for the positional accuracy of the UWB system in operation. Other work has been conducted into the use of RTS systems, and under these conditions continuous ground truth for position was shown to be practical. An example of this type of study is the work conducted by Yang et al. [92]. In this investigation, the use of a UWB system for tracking workforce movement on construction sites was tested. To understand the reliability of the system an RTS system was used to provide to ground truth for assessment [92]. This work has shown the capability of the use of RTS systems for continuous tracking [92]. A Leica Total Station has been shown to effectively track a UAV in space for the purposes of ground truth assessment, however the work does not mention real time synchronisation of the ground truth to the sensor data [40]. This would mean that although an overall trajectory assessment may be made, there is no way to tell the location of the UAV at each time step. This is mirrored by the work conducted by Roberts [93].

In terms of alternatives, a VICON system is the only alternative in terms of ground truth technology as this level of accuracy is not available elsewhere in the same environments. However, other systems with comparable accuracy such as the VICON system do not function reliably outdoors at the same ranges as the RTS [94]. The reliability of VICON systems is also dependent on the initial calibration for each setup, and may vary throughout testing if the camera position alters [95]

3.3 Robotic Total Station

A key component in this project is the determination of the positional estimates' reliability. In order to effectively and robustly determine this, a ground truth system is required. As previously described, there is a distinct split in the current research on this topic; much of the state estimation research contains no reliable ground truth. Systems such as the VICON motion tracking camera system, is not applicable in daylight at extended ranges, expensive and not easily portable [95]. Some of the existing work uses RTS for the ground truth. However, none of this work is focused on the reliability of the ground truth and the impact such a reliable system may have.

A theodolite is a type of surveying equipment used to measure the distance and angle to targets [96]. The RTS or Robotic Theodolite is more advanced version of this tool [96]. The RTS measures distance through the use of a 918nm infrared laser via time of flight [96]. It may also robustly measure horizontal and vertical angles through the use of magnetic encoders [96]. The RTS is capable of automatic distance measurement, storage and processing [96]. In surveying, the RTS may be set up over a known point and used to measure points of interest relative to the known location [96]. Most importantly for this application, some RTS systems are capable of "tracking" a moving target through the use of an onboard camera, optionally aided by infrared LEDs placed on the desired target [96]. Targets may be prisms or retro-reflectors with or without LED tracking assistance [96]. Measurements may be taken to surfaces without targets, however these points may not be tracked [96].

A key point in the use of an RTS is the performance. The Trimble S7 RTS used in this investigation may measure distances to $\pm 2mm + 2ppm$ for standard mode and $\pm 4mm + 2ppm$ while tracking (out to a maximum range of 1000m), dependent upon the prism in use [96]. In this case the ppm represents parts per million over the range tracked [96]. The base accuracy of the measurement in the first instance is $\pm 2mm$, with an extra $2 \times 10^{-6}m$ for every meter measured. The angular resolution is defined as $1''$ (arc second) or $2.77° \times 10^{-6}$ [96]. The key point in the use of the RTS is the definition of its reliability. The RTS used for this study, is specified to be accurate to DIN 18723, and is regularly calibrated to this standard [96]. This calibrated and defined accuracy opens the door to verification of position to levels not previously demonstrated. An RTS system is certified to maintain a set minimum accuracy throughout operation provided adequate setup conditions are met [96].

3.3.1 Robotic Total Station Operation

As mentioned, a key component of the RTS use is its setup. This section describes the procedure used and the target selection process.

The RTS was connected to a base station computer using the TSC3 remote control. The controller allows for wireless control of the station, but also allows for serial connection to a computer [96]. Trimble utilises a specialised NMEA string to transfer information via this link [96]. The format may be seen in Figure 3.6. The TSC3 controller was connected to a Linux machine via a RS232 to USB cable. To synchronise the measurement times between the UGV sensors and the RTS , the Robot Operating System (ROS) was utilised. A specialised node was written that read the strings from the TSC3 controller, separated the pertinent data and appended a header containing the system time. This was then recorded on the UGV platform along with all other sensor data through the use of the ROSBag function.

Setup of the RTS is made by fixing the system to a surveying tripod, and securely placing it onto stable ground. The system is then roughly levelled using the adjustable legs, as seen in Figure 3.7b. Next, the system is switched

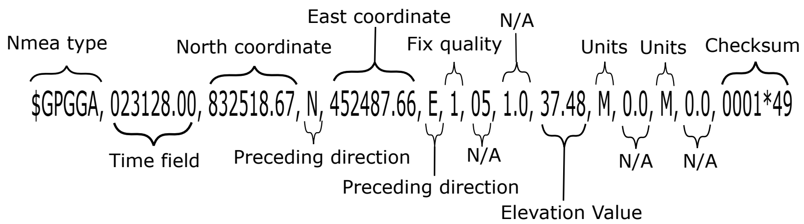


Figure 3.6: Breakdown of Trimble pseudo NMEA string.

on, and the calibration process is started. The first step is to define the location of the RTS , for readability this is often chosen as 100m in x by 100m in y and 10m in z. The location of the RTS is inconsequential to its operation, so such accommodations may be made for ease of trouble shooting. The second step is levelling the RTS to an acceptable degree. This was chosen as ± 2 arc seconds as over repeated usage it was found that the stability of the ground did not allow for greater levelling to be consistent throughout the test. This is achieved using the onboard level shown on the TSC3 controller, and adjusted using 3 thumbscrews between the tripod and the RTS base. The RTS is then left untouched for two minutes to ensure it is stable at this level. Next, the axes are defined. This is again an arbitrary decision and the definition was made on a case by case basis to aid readability of the results. Normally the direction with the most activity is chosen as positive y and the right of that is chosen as positive x. The z axis is always oriented vertically, with positive as up.

In order to utilise the functionality of the tracking capability of the RTS, the optimal target for tracking and the viable circumstances in which the target may be tracked had to be determined. To determine this, a selection of the different styles of target is chosen. A MT1000 target, with and without tracking LEDs, and, an AT360 target, again with and without LEDS was also used, seen in Figures 3.8a and 3.8b respectively. A 180° Korec prism was also tested along with a basic retro-reflector sticker, seen in Figures 3.8c

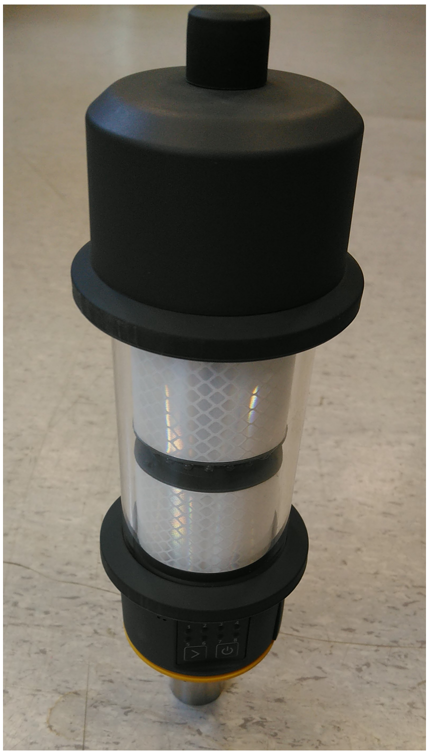
(a) (b)

Figure 3.7: Trimble S7 RTS (a), and RTS mounted and set up (b).

and 3.8d respectively. Testing is carried out by locking the total station onto the target, then the target is moved. As there are no tracking statistics available for the RTS, movements made are representative of those expected during experimental conditions, and assessment of the lock durability is made under expected conditions.

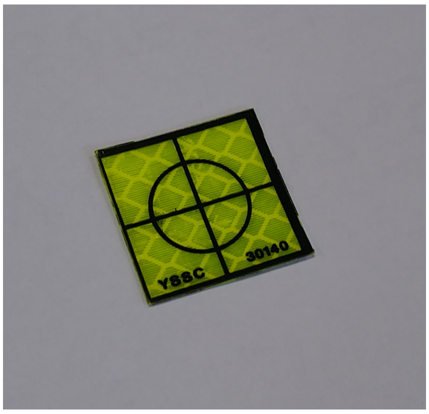
No data is logged on the RTS when a loss of target tracking occurs, therefore for this experiment a target is moved, while the operator listens for the audio target loss alert. The retro-reflector sticker target lock is not reliable, frequently dropping out. With assistance from LEDs the AT360 prism configuration proved very robust, with no dropouts. The MT1000 and 180° Korec prism were found to allow for a robust target lock, with no dropouts unless obscured. The MT1000 allowed for more diverse range of movement, due to the 360 field of view of the prism. For future work it was decided that the experiments should be conducted using the AT360 target with LED assistance. This is due to its lighter weight and 360° target. Meaning a lower chance of the mass of the target causing deflection in the mounting pole, and therefore lower change of errors in the ground truth measurement.

(a) MT1000 target.



(b) AT360 target.

(c) Korec 180 target.



(d) Retro-reflector sticker target.

Figure 3.8: RTS targets used in testing

3.4 Ultra Wideband Localisation

As shown in the literature, there are a number of localisation systems that may operate similarly to GNSS where satellite operation is not viable. Out of these systems UWB is an interesting candidate, due to its relative accuracy, portability, low cost and applicability to a variety of environments. Therefore, the UWB system fulfils the requirements stated in the introduction of this chapter. It is low cost, capable of NCLOS operation, is self contained and has a resolution on the order of centimetres [97]. This system has a major advantage over the vision based systems due to it not relying upon the lighting conditions of an environment.

3.4.1 Operating Principles

UWB localisation systems use radio frequency (RF) signals to calculate range, and sometimes angle between "nodes" [69]. UWB systems differ from other techniques due to the operating spectrum [98]. As the system transmits over an extensive bandwidth the power spectrum density decreases, leading to decreased interference from other RF signals [98, 99, 100]. UWB systems perform ranging through a variety of methods, however most common are Time Of Flight (TOF) or Time Of Arrival (TOA) and, Time Difference Of Arrival (TDOA) [101, 102]. TOF calculates the range between the nodes by measuring the return time of a message, in other words the message flight time. The RF signal travels at the speed of light, therefore the range may be taken as the time to signal return, divided by two to account for the return trip, and multiplied by the speed of light. TDOA calculates the position of a node relative to the other nodes in the network by using the differing times of signal arrival [101]. The TDOA approach requires networking between all nodes and an absolute time for the system [101]. TOA calculates the range by finding the time taken for the signal to move between node pairs [102]. This allows individual range measurements be made for individual node pairs, without the need for information from all the networked nodes. In this case, the node common to all pairs is known as the "rover ", and the nodes paired with the rover are known as the "anchors". The anchor node positions are known, while the rover node position is unknown.

In a TOA configuration the range is calculated as the time taken for the signal to travel between nodes in one direction multiplied by the speed of light. The range may also be given as shown in [80, 99]. This equation is non-linear, and is described in Equation (3.14), where the anchor positions $A = A1, A2, A3...An$, $An = xn, yn$ and x_k, y_k are the rover positions at time k. The distance between the rover and anchor nodes is represented as D_{RAn}

$$D_{RAn} = \sqrt{(x_k - x_{An})^2 + (y_k - y_{An})^2} \tag{3.14}$$

3.4.2 Pozyx UWB system

The Pozyx system is built upon the Decawave DWM1000 chip [103]. Other commercial systems currently available on the market include Ubisense, BeSpoon, and Sappire Dart [99, 104], however the main difference in these platforms is the implementation of localisation algorithms and their networking [99, 104]. Many of these other systems utilise the same DWM1000 chip [105, 106]. Some work has been conducted into comparison between the DWM1000 chip and its competitors, however the DWM1000 has been shown to perform reliably, and under certain circumstances more accurately [104]. The DWM1000 chip utilises the TOA method of UWB ranging [104].

The Pozyx system operates through two types of nodes. The "Anchor" node and the "Rover" node [103]. Both nodes contain the same UWB components and may be used interchangeably, however the Rover node PCBs also contain an IMU and magnetometer [97]. A principle advantage of UWB systems as stated previously is the ability to communicate between nodes during ranging

[103]. The Pozyx system allows for range measurement inquiries to be sent from any node to any other node [103]. This would allow for localisation of remote devices, without the need of onboard processing. However in this investigation onboard processing is available so it is used. The Rover node is connected to the Raspberry Pi via USB serial, and interfaced to the ROS system through the use of ROS nodes, allowing for time synchronisation.

3.5 Formulation of the Extended Kalman Filter

As previously described, for the intended system a discrete EKF is an acceptable state estimator choice. As described in Section 3.4.1, the measurement model needed for a range based sensor such as the UWB system is non-linear. Therefore, irrespective of any system model constructed there is a demonstrable need for a non-linear filter.

As stated in Section 3.1, the prediction phase of the state estimator may use "control inputs" to improve the reliability of the prediction. In this investigation, an IMU sensor input will be taken as a control input along with quadrature wheel encoder measurements. This allows for more reliable state prediction in between measurement corrections from the UWB system [41].

The non-linear state transition and measurement models for an EKF are described in Equation (3.15) and Equation (3.16) where $\widehat{X_k}$ and $\widehat{Z_k}$ represent the state and measurement vector estimations at time k, controlled input into the system is represented as u_k and where w_{k-1} and v_{k-1} are the system and measurement noise respectively, at the previous time step.

$$\widehat{X_k} = f(X_{k-1}, u_k; w_{k-1}), \tag{3.15}$$

$$\widehat{Z_k} = h(X_{k-1}, v_{k-1}). \tag{3.16}$$

As the UWB outputs range measurements for each anchor, (at known location) the system is treated as a range-based localisation problem. For this case, the UWB measurements are considered as ranges between the UGV position and the anchor locations, not Cartesian x, y, z values. This has been shown to be an effective method of incorporating UWB systems into an EKF [107].

3.5.1 Motion Model

As described in [59] the control input from the odometry model for the formulation of the state transition function was demonstrated by [108] and [109]. In this case ΔD_k is the linear displacement at time k, calculated from the wheel circumference and input pulses from the encoders representing angular displacement $\Delta\theta$. This is calculated for each wheel then averaged between the two sides as shown in Equation (3.17). The state model may be seen in Equation (3.18).

$$\Delta D_k = \frac{r_L \Delta\theta_{L_k} + r_R \Delta\theta_{R_k}}{2}. \tag{3.17}$$

$$f(\widehat{X}_{k-1}, u_k) = \widehat{X}_{k-1} + \begin{bmatrix} \Delta D_k \cos(\phi_k + \frac{\Delta\phi_k}{2}) \\ \Delta D_k \sin(\phi_k + \frac{\Delta\phi_k}{2}) \\ \phi_k + \Delta\phi_k \end{bmatrix}, \tag{3.18}$$

From this the state vector is defined as being a 3×1 dimensional matrix containing Cartesian position x, y and heading, ϕ. The control input is therefore represented as $u_k = [\theta_L \theta_R \Delta\phi]^T$. Where $\Delta\phi$ is provided by the onboard IMU. This allows the definition of the state Jacobian as shown in Equations (3.19) and (3.20).

$$G_{u_k} = \frac{\partial f}{\partial u}(\widehat{X}_k^-, u_k), \tag{3.19}$$

$$G_{u_k} = \begin{bmatrix} \frac{r_L \cos\left(\phi_k + \frac{\Delta\phi_k}{2}\right)}{2} & \frac{r_R \cos\left(\phi_k + \frac{\Delta\phi_k}{2}\right)}{2} & -\frac{D_k \sin\left(\phi_k + \frac{\Delta\phi_k}{2}\right)}{2} \\ \frac{r_L \sin\left(\phi_k + \frac{\Delta\phi_k}{2}\right)}{2} & \frac{r_R \sin\left(\phi_k + \frac{\Delta\phi_k}{2}\right)}{2} & \frac{D_k \cos\left(\phi_k + \frac{\Delta\phi_k}{2}\right)}{2} \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.20}$$

3.5.2 Measurement Model

The measurement model for the UWB readings relates the range readings taken between the rover and anchor node pairs to the position of the rover in x and y. At each measurement update, it is assumed that all 6 of the anchors provide a range measurement $Z_{Bi} = [z_1, z_2 ... z_6]$. Uncertainty in the range measurements is represented by the sensor variance R, an $n \times 1$ matrix where n is the number of anchors in the system. For the initial assessments and in keeping with the principle of "black box modelling", the Pozyx localisation algorithms are used. This provides an x, y position as an output. The equation shown in Equation (3.14) is used to calculate the range measurements without the need to compensate for measurement uncertainty as mentioned previously. This method was chosen as it allows for an assessment of the improved sensor variance with minimal other variables. By formulating the EKF in the range model form, rather than as a direct observation model, it is possible to assign different variance values for each anchor in the future work. This process is described in Algorithm 1. Equation (3.21) shows the measurement function, with the expansion of this seen in Equation (3.22).

$$\widehat{Z}_{Bi} = h(\widehat{X}_k^-, A_{B_i}), \tag{3.21}$$

$$h(\widehat{X}_k^-, A_B) = \begin{bmatrix} \sqrt{(x_k - \mathrm{x}_{b1})^2 + (y_k - \mathrm{y}_{b1})^2} \\ \\ \sqrt{(x_k - \mathrm{x}_{b2})^2 + (y_k - \mathrm{y}_{b2})^2} \\ \\ \sqrt{(x_k - \mathrm{x}_{b3})^2 + (y_k - \mathrm{y}_{b3})^2} \\ \\ \sqrt{(x_k - \mathrm{x}_{b4})^2 + (y_k - \mathrm{y}_{b4})^2} \\ \\ \sqrt{(x_k - \mathrm{x}_{b5})^2 + (y_k - \mathrm{y}_{b5})^2} \\ \\ \sqrt{(x_k - \mathrm{x}_{b6})^2 + (y_k - \mathrm{y}_{b6})^2} \end{bmatrix}, \tag{3.22}$$

With the Jacobian of the measurement model is obtained as seen in Equation (3.23), Equation (3.24).

$$H_{z_k} = \frac{\partial h}{\partial x}(\widehat{X}_k^-), \tag{3.23}$$

$$H_{z_k} = \begin{bmatrix} \frac{x_k - \mathrm{x}_{b1}}{\sqrt{(x_k - \mathrm{x}_{b1})^2 + (y_k - \mathrm{yb}_1)^2}} & \frac{y_k - \mathrm{y}_{b1}}{\sqrt{(x_k - \mathrm{x}_{b1})^2 + (y_k - \mathrm{y}_{b1})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b2}}{\sqrt{(x_k - \mathrm{x}_{b2})^2 + (y_k - \mathrm{y}_{b2})^2}} & \frac{y_k - \mathrm{y}_{b2}}{\sqrt{(x_k - \mathrm{x}_{b2})^2 + (y_k - \mathrm{y}_{b2})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b3}}{\sqrt{(x_k - \mathrm{x}_{b3})^2 + (y_k - \mathrm{y}_{b3})^2}} & \frac{y_k - \mathrm{y}_{b3}}{\sqrt{(x_k - \mathrm{x}_{b3})^2 + (y_k - \mathrm{y}_{b3})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b4}}{\sqrt{(x_k - \mathrm{x}_{b4})^2 + (y_k - \mathrm{y}_{b4})^2}} & \frac{y_k - \mathrm{y}_{b4}}{\sqrt{(x_k - \mathrm{x}_{b4})^2 + (y_k - \mathrm{y}_{b4})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b5}}{\sqrt{(x_k - \mathrm{x}_{b5})^2 + (y_k - \mathrm{y}_{b5})^2}} & \frac{y_k - \mathrm{y}_{b5}}{\sqrt{(x_k - \mathrm{x}_{b5})^2 + (y_k - \mathrm{y}_{b5})^2}} & 0 \\ \frac{x_k - \mathrm{x}_{b6}}{\sqrt{(x_k - \mathrm{x}_{b6})^2 + (y_k - \mathrm{y}_{b6})^2}} & \frac{y_k - \mathrm{y}_{b6}}{\sqrt{(x_k - \mathrm{x}_{b6})^2 + (y_k - \mathrm{y}_{b6})^2}} & 0 \end{bmatrix}. \tag{3.24}$$

The calculation of the Jacobian was performed using the MATLab symbolic toolbox; this was then used to construct the functions used in the offline EKF .

Algorithm 1 Range based EKF Localisation

Prediction:

$\widehat{X}_k^- = f(\widehat{X}_{k-1}, u_k)$

$G_{x_k} = \frac{\partial f}{\partial x}(\widehat{X}_k^-, u_k)$

$G_{u_k} = \frac{\partial f}{\partial u}(\widehat{X}_k^-, u_k)$

$\widehat{P}_k^- = G_{x_k}\widehat{P}_{k-1}G_{x_k}^T + G_{u_k}QG_{u_k}^T$

Correction:

$\widehat{Z}_{Bi} = h(\widehat{X}_k^-, A_{B_i})$

$H_{z_k} = \frac{\partial h}{\partial x}(\widehat{X}_k^-)$

$K_k = \widehat{P}_k^- H_{z_k}^T (H_{z_k}\widehat{P}_k^- H_{z_k}^T + R)^{-1}$

$y = Z_{Bi} - \widehat{Z}_{Bi}$

$\widehat{X}_k = \widehat{X}_k^- + K_k y$

$\widehat{P}_k = (I - K_k H_{z_k})\widehat{P}_k^-$

if *measurement_is_available* **then**

    do **Correction**

else

    do **Prediction**

end if

3.6 Methodology

This section will give an overview of the system employed to test the EKF formulated in the previous section, the data collection and networking system employed, and the procedure for experimentation. The characterisation of the Pozyx system is also included in this section, the information gathered from this process is used as the measurement variance for the EKF in the later experiments.

3.6.1 Unmanned Ground Vehicle

The UGV used in this system was a platform designed and constructed specifically for this experiment can be seen in Figure 3.9. The kinematics of the system required a skid steer platform, therefore a 4 wheel drive system, with fixed axels was used. The frame was constructed from 15mm square extruded aluminium sections. Brushed 12V DC motors with integrated quadrature encoders were chosen for propulsion. The system did not include suspension, as this would add un-modelled complexity for very little benefit considering the relatively flat environment. The overall purpose of this test being a more constrained, less complex kinematic model; therefore increased complexity was viewed as contrary to this objective.

An overview of the system architecture may be seen in Figure 3.10. Two computers are used onboard the system, with the principle data collection for this investigation running on the Nvidia Jetson TX1. ROS master is run on this system, along with the data capture carried out though ROSBag. ROSBag allows for all activity on the ROS network in use to be saved as it is transmitted, allowing for playback and access at a later time. This tool allows for simulation of systems offline in a manner similar to if they were present at the time of data collection. This is done to avoid timing issues which may arise the EKF sensor observations are transmitted over WLAN prior to timestamping. The RTS is connected to the system as stated via a custom ROS node and TSC3 controller connected via a serial link to a laptop. This code may be seen in the appendix of this work. This laptop is then connected to the ROS main via WLAN.

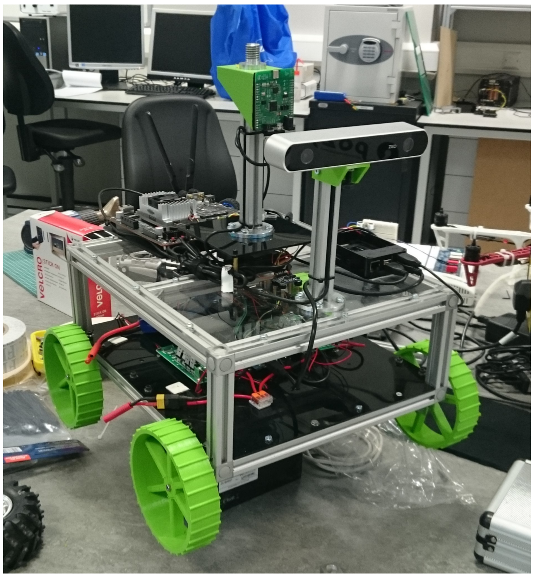


Figure 3.9: The Unmanned Ground Vehicle platform used in testing.

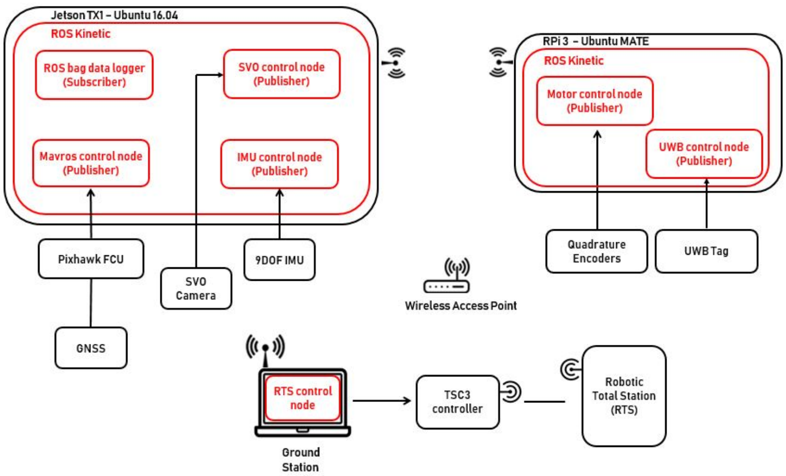
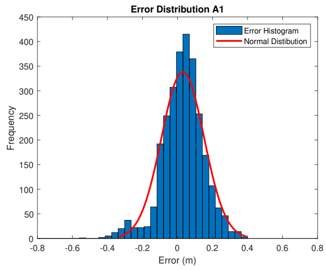


Figure 3.10: Plan of the system architecture.

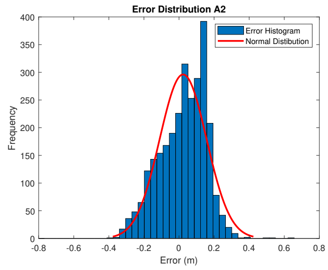
3.6.2 Pozyx Characterisation

To determine the sensor variance of the UWB system under the conditions used for testing the following procedure was applied. Six Pozyx anchor nodes were placed in the intended environment roughly equidistant around the perimeter mounted on tripods at approximately 1m height, off set from metal components. A rover node was mounted upright onto a mast on the UGV platform. Positions of the anchor nodes were measured by the RTS, with the reference being their antennae. The range measurements between the anchors and rover node were recorded during movements over the space in paths and trajectories consistent with those expected for a UVG. The position of the UGV was monitored through the use of the RTS . Using the ground truth taken from the RTS system the absolute range error was calculated. These range error measurements can be seen in histogram form in Figure 3.11. The range error measurements were then collected together and the standard deviation was calculated. The combined range error measurements is displayed in histogram form in Figure 3.12. As may be seen, the range error takes the Gaussian White Noise form expected from the literature. The mean was roughly 0m, however some variation was found.

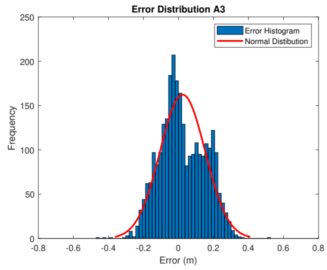
The variance used in the state estimator was taken as described in the literature as the square of the standard deviation. In this case the variance of the combined anchor range error was found to be $0.0175m^2$.



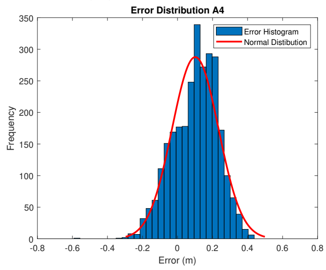
(a) Anchor 1



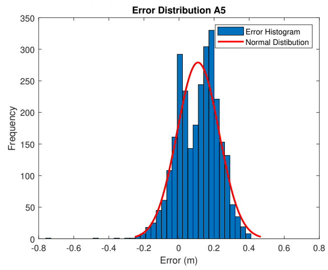
(b) Anchor 2



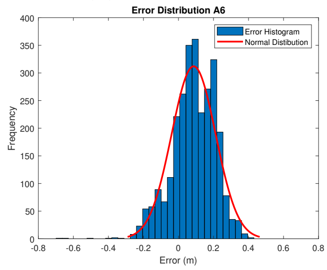
(c) Anchor 3



(d) Anchor 4



(e) Anchor 5



(f) Anchor 6

Figure 3.11: Error Distributions of UWB range measurements.

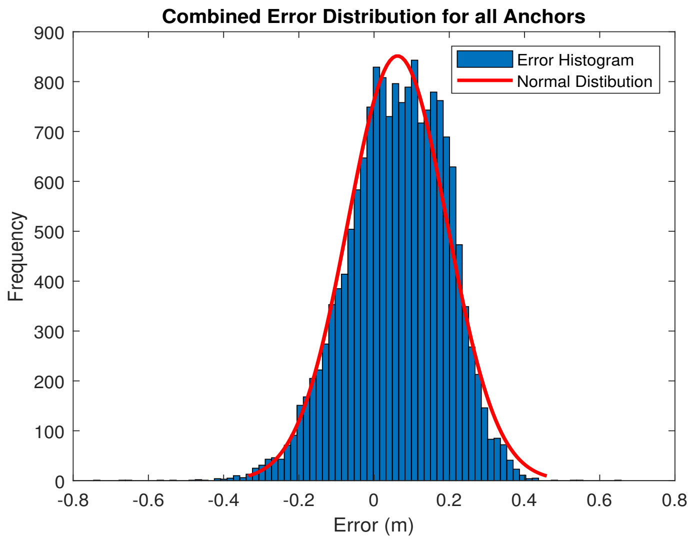


Figure 3.12: Error distribution for all anchors.

3.6.3 Testing Procedure

The testing space was an outdoor environment on the Byrom Street campus, next to high metal structure buildings and terrace houses. This space was chosen as GNSS positioning had been demonstrated to be unreliable in the past. As demonstrated during initial set up of the RTS system. A survey grade

GNSS system was unable to acquire satellite lock within half of the testing space. Therefore, this space is an example of an outdoor environment where the system outline in this project may be applicable. The testing space shown in Figure 3.13 had an area of $214.07m^2$.

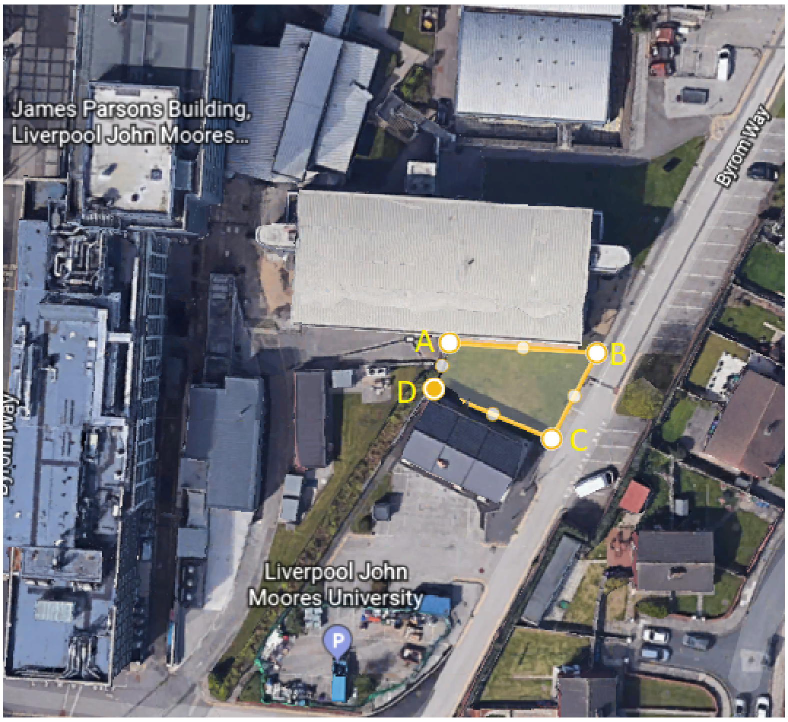


Figure 3.13: Aerial view of testing space outlined in yellow.

Six Pozyx anchors were placed roughly equidistant around the perimeter of the space, mounted on tripods and powered from the mains supply outlets. This may be seen in Figure 3.14 and Figure 3.15.

Figure 3.14: View of Pozyx anchor mounted on tripod as implemented.

Figure 3.15: Looking North, view of the Pozyx anchors mounted on tripod as implemented. Running left to right along line AB

The RTS system was positioned outside of the testing space, where it had a clear view of all of the area to allow for tracking of the UGV as shown in Figure 3.16. The RTS was then used to measure the position of the anchors for the UWB setup. The anchor positions were then input into the ROS node to allow for localisation. To demonstrate an additional benefit of using the RTS , the system was geo-referenced to a known ground control point, meaning that the UWB measurements were transferable to other sources through the use of a common reference frame. The ground control point used was on the edge of the space and is used in ongoing site maintenance. This point was measured through the use of a bipod - surveying pole and the AT360 prism as shown in Figure 3.17.

Figure 3.16: Looking North-West, view of testing space showing placement of RTS. Taken from point C

Figure 3.17: View of the backsight and georeference point with AT360 prism, located at point B.

Once the UWB system was set up, the RTS was set to lock onto and track the AT360 prism mounted on top of the UGV. The system was then set to output measurements at its maximum rate of 1Hz, this data was the networked and synchronised via the custom RTS ROS node. The fastest rate allowable was chosen to allow for a more full comparison between the EKF and ground truth data.

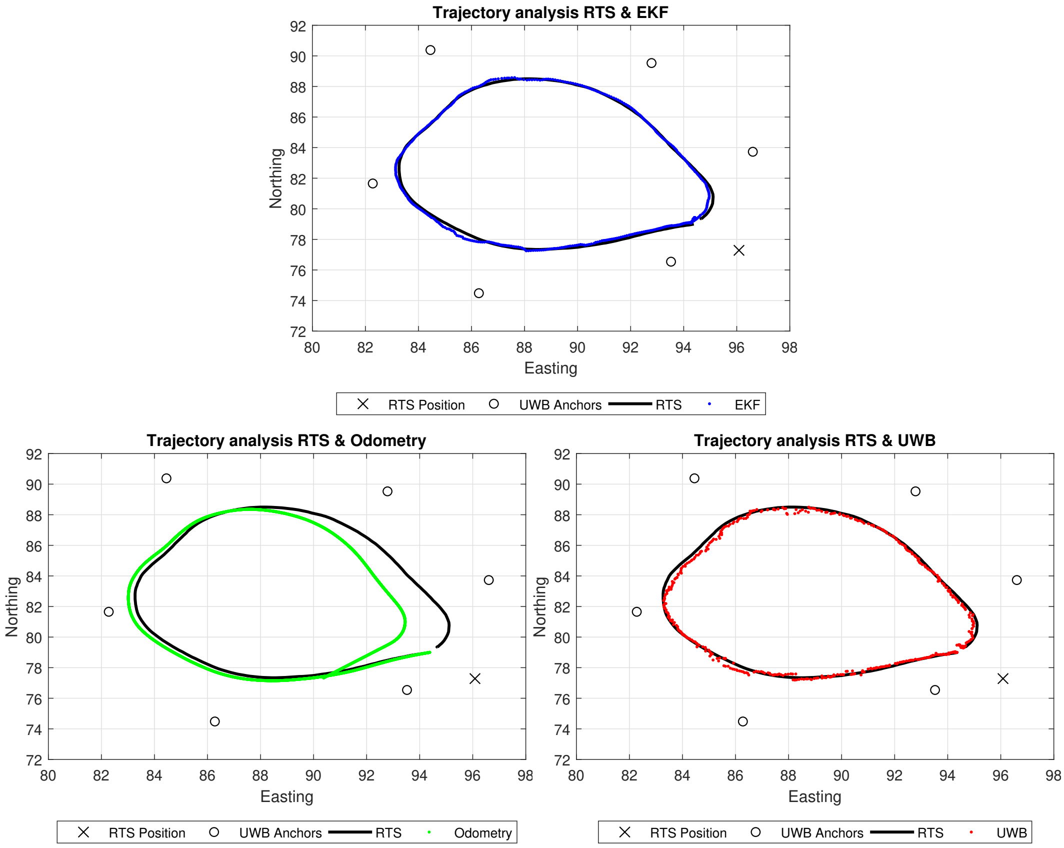
The UGV was then piloted around the space manually by an operator, whilst the sensor readings were recorded through the use of the ROSBag function. This data was then stored for post processing into a .mat format which was easily used for the EKF offline operation. Several trajectories were tested, with the aim of space coverage, to obtain the most varied UWB conditions. To allow for ongoing error checking throughout the experimental process the TSC3 controller was used to monitor the recorded position of the UGV. The TSC3 allows for an augmented reality view of the space though the use of the onboard camera of the S7 RTS . This allows for an overlay of the measured points as seen in Figure 3.18.

Figure 3.18: View of the measured points of overlaid onto the video output of the RTS .

3.7 Results and Analysis

The results of the EKF estimate of the UGV trajectory may be seen in Figures 3.19a, 3.20a and 3.21. As can be seen from the odometry output in Figure 3.19b, it is clear that this sensor provides a smooth measurement, which drifts as a function of distance travelled. In contrast, the UWB data shown in

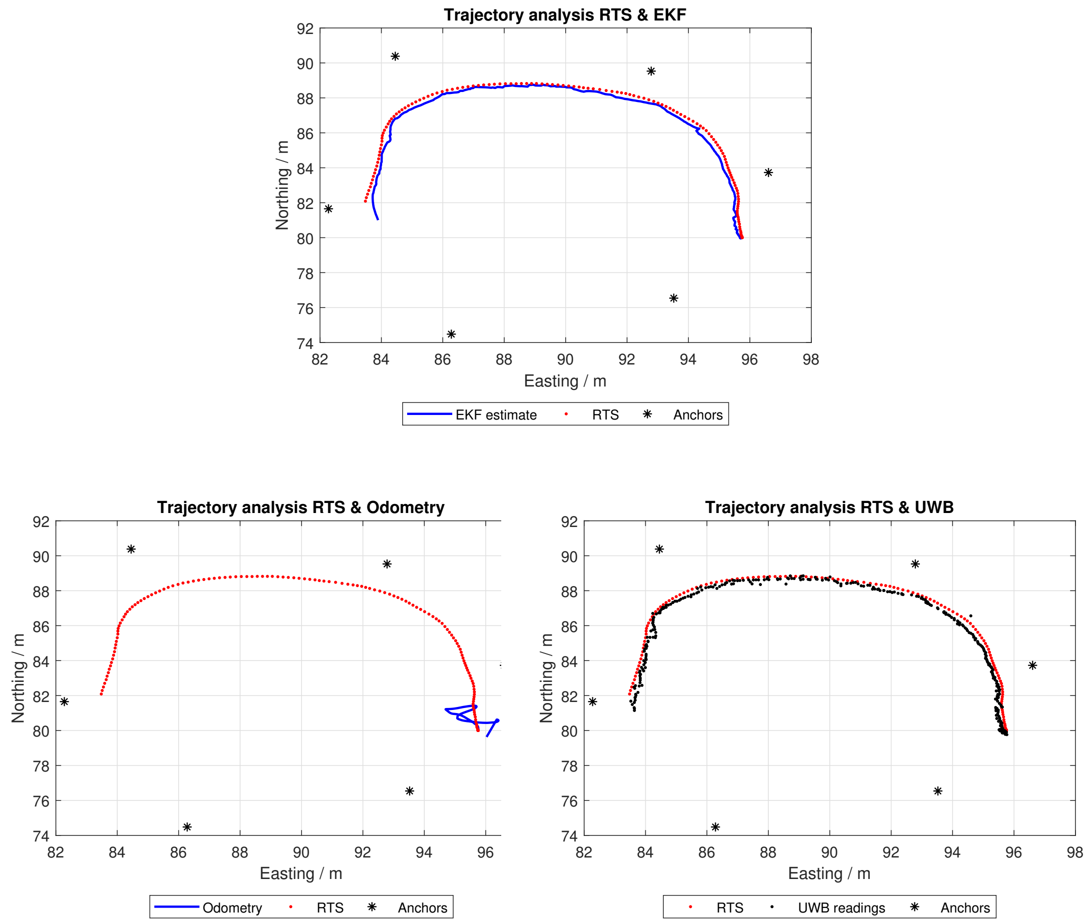
Figures 3.19c and 3.20c does not show appreciable drift, but is relatively noisy.



(b) RTS vs Odometry

(c) RTS vs UWB

Figure 3.19: Resulting paths from all techniques and RTS active tracking.



(b) RTS vs Odometry

(c) RTS vs UWB

Figure 3.20: Resulting paths from all techniques and RTS active tracking.

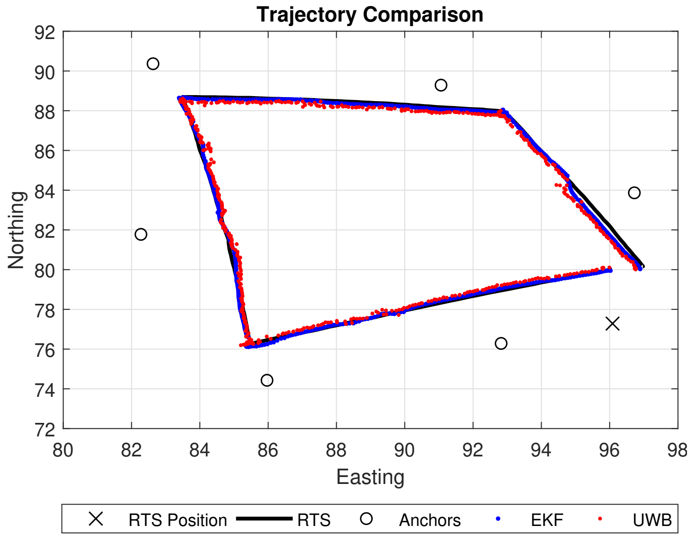


Figure 3.21: Trajectory comparison between EKF and UWB .

Through the use of the EKF, the measurement sources may be used to estimate the trajectory of the system with reasonable accuracy, as defined by the specification laid out at the beginning of this chapter. Figure 3.21 demonstrates the variation found in the UWB measurements increased when closer to the east side of the testing space. This is logical as that area presents the lowest density of UWB anchors. In this area the EKF does diverge from the RTS track, and begins to run parallel with an offset, this is most likely due to this divergence, all of this is seen in Figure 3.21. The assessment that the EKF system improved upon the UWB estimate is supported by the lower errors found in the EKF trajectory seen in Table 3.1. An important point to make clear is that although the EKF allows for a smoother estimate of the position of the UGV, it is not able to account for errors such as those seen in the furthest east point of Figure 3.19c and the western side of Figure 3.20a. In these cases, the UWB data is not reporting a position in line with that seen by the ground truth, and the odometry data has drifted sufficiently to be unable to accommodate for this. While it may be possible to alter the process noise and sensor variance constants to a degree which may "tune" this out, such

Axis	**Mean Error (m)**	**Standard Deviation of Error (m)**
UWB (x)	0.0621	0.1478
UWB (y)	0.0718	0.1510
EKF (x)	0.0167	0.1611
EKF (y)	0.0071	0.1326

Table 3.1: Table of error metrics used to asses EKF performance.

an approach would lead to over-optimisation of the EKF , and such results only being attainable under select datasets. Although the EKF provides an estimate of the UGV position it cannot accurately predict this information with incomplete or erroneous data. Where the mean of the sensor readings are not zero, the EKF will not provide an accurate estimate, as shown in the examples presented here. Specifically shown at the furthest east point of Figure 3.21.

3.8 Conclusions

The work demonstrated in this chapter builds on the information provided in the previous chapter which demonstrated the requirement for a supplementary sensor to the IMU. This study shows that the use of an IMU, wheel encoders and UWB systems allow for robust positional estimation. From the results collected, it may be concluded that the use of an RTS as a means of calculating the sensor variance is highly effective. We can also see that the means by which the ground truth is collected allow for a good assessment of the functionality and performance of the EKF used.

An interesting development is that the UWB system is noticeably more reliable when the ranges between nodes are lower. This range error variation is noticeable on the scales used in this work. As described in the introduction to this project, the purpose of this work is the investigation of the efficacy of improved sensor characterisation. As the sensor characteristics vary as a function of a known and measurable variable, it is logical to pursue the modelling of such a function. The next chapter focuses on the modelling and implementation of this as a sensor measurement variance function. The current configuration of the system gives preference to rover UWB node positioning, it is also necessary to investigate non-ideal configurations. The next chapter will also include a modified UGV, more akin to a platform useable for purposes other than UWB navigation.

Chapter 4

Model Based Variance for Motion Estimation in 3 Degrees of Freedom

As shown in the previous chapter, the use of an EKF for the fusion of UWB sensor measurements allows for an effective means of tracking the position of an agent. The work shown in the last chapter also demonstrates a novel method of calculating the sensor variance of the UWB system through the use of the RTS. The RTS is shown to provide a reliable ground truth for this system. This chapter will build on the work of the last chapter and further develop the EKF system for indoor use. Indoor environments are expected to provide significantly more challenging circumstances, due to increased reflective surfaces which may lead to multipath errors within the UWB system. The UGV testing platform will also be altered. As stated, the system used previously was custom built for the experiment, and therefore incorporated an ideal mounting point for the rover node. However, in many circumstances this is unlikely to be the case, due to limitations in design and other sensor such as Lidar which require a clear field of view. To develop a system capable of functioning in real world environments it is necessary to investigate these effects. In this experiment the rover node will be mounted in a less prominent location to investigate the importance of rover node placement and the extent to which the EKF may accommodate for non-ideal node positioning. A large component of this chapter will involve more complex analysis of the sensor measurement variance and investigate alternate methods of representing this within the EKF .

4.1 State of the Art

As previously mentioned in the last chapter, there are a large number of ways in which to construct a state estimation algorithm. The conventional manner has been described in the last chapter for the linear Kalman filter and the non-linear EKF. However, for some applications variations of these algorithms have been formulated. An example is the Adaptive Kalman Filter (AKF) [110, 111, 112]. This filter may be used as a modification to several conventional filter forms such as the EKF or the UKF [110, 111, 112]. The key difference in the formulation of the AKF is the manner of specifying the sensor variance. In conventional filters as mentioned already, the sensor variance is static. However, in the AKF, no prior information about the sensor variance is used. Instead, the residual and innovation of the observation is used to estimate the measurement covariance [110, 111, 112]. In cases where the sensor noise is shown to alter throughout the operating environment either in terms of time as in the work by Hu et al. or in relation to another variable this method is highly applicable [112].

Work has been conducted into the use of an adaptive variant of the EKF known as the AKF for UAV applications [113, 114]. As described in many works, inaccurate or incomplete information about the measurement and state noise R and Q respectively will lead to differences and divergence in the filter [113, 114]. The adaptive component of the filter used to re-calculate the R and Q values is constructed through the use of a Fuzzy Logic Adaptive Controller [113, 114]. This controller monitors the residuals of the observation with respect to the estimated states [113, 114]. If these residuals do not form the expected GWN then it is assumed that the R value is incorrect as the filter is diverging [113, 114]. Although this system is shown to function well in the tested systems, as with all adaptive variants of the KF the lack of an initial empirical investigation leads to the possibility of an incorrect R value [113, 114].

The Cubrature Kalman Filter (CKF) was proposed as an alternate means of estimating the states of a non-linear system [115, 116, 117]. As discussed, the filter in use in this work linearises the system through the use of a first order Taylor series approximation. This is done to maintain the Gaussian assumption upon which this branch of Bayesian filtering is based [41]. The CKF utilises the filter namesake "Cubrature" rules [117]. These rules allow for what is referred to in the work as "nonlinear filtering through linear estimation theory" [117]. However, as stated before, one of the main issues with non-linear systems is their characterisation [117]. In order to fully construct a CKF, a full model of the non-linear system in question and the posterior noise statistics are required [117]. An issue with this filter construct is that although this method deals with issues arising from non-linear systems, it is sensitive to non-Gaussian noise in the observation and prediction [118, 119]. As the CKF remains dependent upon an accurate and high resolution description of the non-linear system to be improvement upon the EKF this method is not within the scope of the project.

Some work has been done to develop a practical method of implementing the CKF framework, however the necessity of a specific system model is against the intention of this PhD work [118, 119].

Analysis of the dynamic sensor measurement variance of the DWC1000 module was conducted in the work undertaken by Ledergerber et al [120, 121]. Of particular interest to this investigation was the effect of relative pose upon the ranging performance. To investigate this two DWC1000 modules were placed 1m, 1.7m, and 2m away from one another. Then the orientation of one of these modules was changed relative to the other and the distance measured by the system was logged. Upon comparison to their VICON system ground truth it was shown that clear variations in the range were present. However, a number of components are lacking in this investigation. Among these are the ranges at which the experiments were performed, the method by which the UWB system is set up, and the variables present in the experimental setup. The initial section of the investigation examines the effect of relative orientation by comparing the effects of rotating a single module about its x axis at a range of 1.7 meters. Other works have demonstrated that UWB systems have variable uncertainty linked to the range between nodes, therefore a single range test without investigating the effect at other ranges leaves room for anomalous readings. The system is also known to be noisy in many environments. This work does not attempt to characterise the base noise of the setup. Therefore, it is not practical to draw conclusions seen from changing uncertainty as this may be a normal characteristic of the system when no pose change is present. The method of modelling used in this work is interesting and does lead to an improvement when implemented into a Kalman Filtering system. Further development of this work may be seen, however the issues regarding range at which the relative orientation characterisation is carried out remain

[120, 121]. The result of the work shown is interesting and does demonstrate a clear improvement over the raw sensor input data. However, as seen in other literature the relative pose is not the only factor [99]. Gaussian Process modelling has also been used as a means of identifying the change between LOS and N-LOS for UWB systems in this study [122]. This work utilised the GP and other machine learning techniques to construct a mitigation regime for UWB sensor error [122].

Work has been conducted into the comparison between 3 prominent UWB systems [104]. Included within this work is an assessment of the standard deviation of the range measurement error provided by these systems [104]. This range error was calculated through the use of 70 ground positions located throughout an indoors space, with some NLOS features [104]. The work presented here supports the other work on the subject with data showing a zero mean distribution for the range error [104]. It should also be noted that the data found in this work demonstrates an increase in the standard deviation of the range error in relation to distance for all of the UWB systems tested [104]. However, some of the UWB systems demonstrated a higher positive correlation between sensor error and node to node range than others [104]. The Decawave based DW1000 system used in both the work presented in this paper and this thesis shows a clear positive correlation to range [104]. In terms of positing accuracy, the study concluded the Decawave system is found to be more accurate than both the Bespoon and Ubisense systems [104].

4.2 UWB Variance Modelling

It has been demonstrated in the previous chapter and literature on the subject that the variance of UWB systems is not a static value and varies as a function of certain components, most notably the node to node range [104, 80, 120, 121, 122]. As has been defined from the literature, there exists a wide variety of variables which will affect the performance of a UWB system [104, 80, 120, 121, 122]. The purpose of this work is to define a method by which a UWB system's variance may be modelled in the intended environment and then used to improve the functionality of a state estimator. The previous chapter demonstrated the method by which a static variance may be calculated for an outdoor environment. This strategy allows for a reliable method for the calculation of sensor error.

As discussed previously, the aim of this project is to firstly, investigate a means of improving estimator performance without further specifying the platform or application; and secondly to develop a means of reliably characterising sensor and estimator performance. An hypothesis of this work that through improving the detail to which the sensor is modelled and represented within the estimator, the degree to which the system itself needs to be modelled may be reduced. This would allow for a wider application to platforms, environments and circumstances, with the possibility of reducing the non-linear elements modelled in the system model.

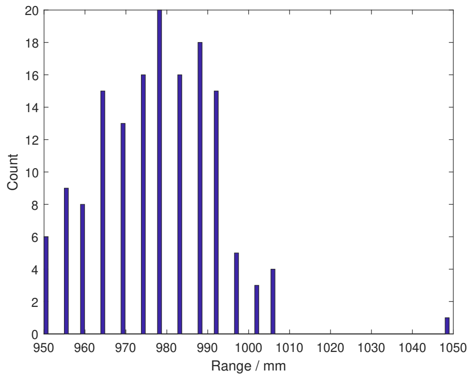
4.2.1 Methodology

For this section of the investigation several setups for sensor variance characterisation are tested. This has to be undertaken as it may not always be practical to characterise the UWB system within the space intended, leading to the need to run these tests in other spaces. Initially, this experiment will characterise the system in a space of similar construction, in terms of materials used, but dissimilar in shape to the intended environment. This is not expected to affect the results, as the characterisation process uses only two nodes in clear view, however it is worthwhile investigating. The principle factors demonstrated in the literature to affect sensor UWB system performance are principally, materials and the RF environment [104, 80, 120, 121, 122]. The spaces used in this work have concrete floors, with the similar exposed and covered areas, suspended ceilings and plasterboard walls. Also, a similar test will be run in the intended EKF testing environment.

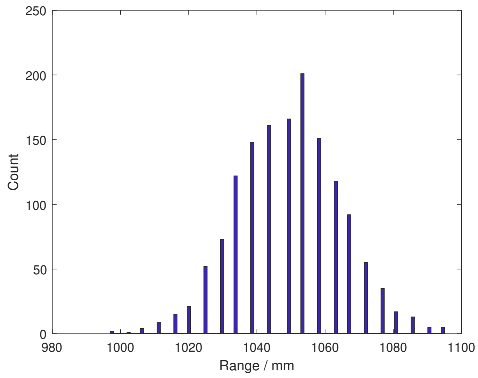
The first experiment consists of, a two node setup with no UGV was used. To collect the required samples, two Pozyx nodes were affixed to tripods. One remains stationary at the end of the corridor, while the other is moved in 0.35m increments starting at 1m away from the stationary node. As the likelihood of a platform closing to less than this range is low. This increment is chosen as it is the size of floor tile used, allowing for easier movement of the tripod. Due to the corridor length, the maximum range is roughly 18m, more than the expected maximum operational range for the EKF testing. To remove possibilities of the power supply leading to unusual results, both nodes were powered from the same wall outlet through the same type of 240v AC to DC USB adapter. The nodes were orientated inwards, with each node pointing at the other.

For the second experiment, the system was characterised in the similar environment used for EKF testing. A Pozyx node is fitted to the UGV platform. The UGV is driven in a straight line away from one node pausing at increments in the same manner as described in the first method. In this case the distance that may be traversed was limited to 7m due to use of the space, however the distance available was thought acceptable to assess whether the test needed to be run again over the same distance. For this setup, as the distance travelled was shorter, the spacing between samples was decreased to 0.1m. To judge this while driving a tape measure was laid on the ground.

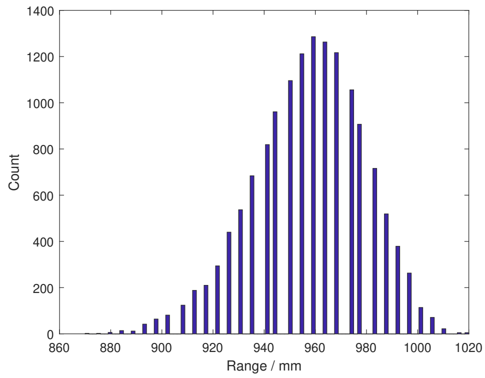
At each range increment 5000 measurement samples were collected. This was found from an initial assessment carried out prior to the main testing. As found in the previous chapter the Pozyx measurement uncertainty takes the form of GWN. Also as previously discussed there is no applicable means of calculating the required sample size for the standard deviation testing, therefore the most robust means to determine this is through the use of a pilot study. This was conducted on the bench top in the laboratory. One Pozyx anchor node was placed 4m away from a rover node. The range was then measured for 150, 1500 and, 15000 samples. As may be seen in Figure 4.1a below the histograms produced for the 150 sample group do not show a strong correlation with a normal distribution.



(a)



(b)



(c)

Figure 4.1: (a)Histogram of 150 sample group. (b)Histogram of 1500 sample group. (c)Histogram of 15000 sample group

However, when the sample size is increased to 1500 and 15000 the normal distribution becomes clear, as seen in Figures 4.1b and 4.1c. Although it would be ideal to use the largest possible sample size time constraints for testing place a limit upon this. A sample size of 15000 with a measurement rate of 50Hz would mean a sample time of 300 seconds per group. Therefore, to compromise a sample size of 5000 is chosen, with a more reasonable time required of 100 seconds per group. Under other circumstances with the remote systems powered though wall outlets a larger sample size may be practical, however with LiPo batteries providing the power in this case this is thought reasonable. Although this may be overcome through larger LiPo batteries, these nodes may be mounted on walls, rendering larger batteries difficult to use. Most importantly, a shorter sample time reduces the possibility of the variables which may effect the UWB system changing during operation.

To determine the effects of the relative orientation upon the sensor variance the first setup was repeated with the nodes in a variety of relative orientations. For clarity of explanation, the terms "front", "back" and, "edge" are applied to the UWB nodes, in this case the front is defined as that side to which the antenna is mounted, a visual representation of this may be seen in Figure 4.2.

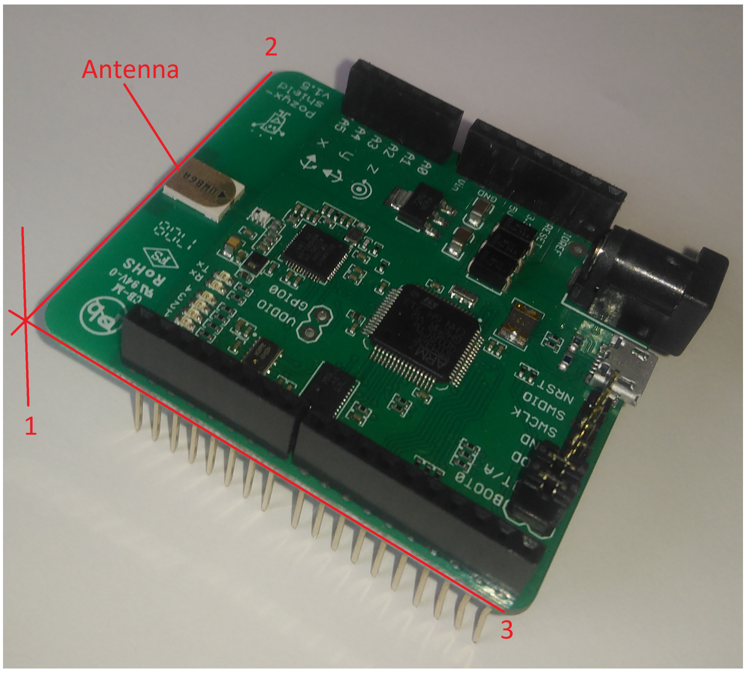


Figure 4.2: UWB node, lines 1, 2 and 3 represent the top left of the device, the "top side" and the "edge" of the UWB node respectively.

The UWB anchor was placed on a tripod at a known location, with the rover node placed on a separate tripod which was then moved in increments, first facing the Anchor, then facing away, and with its edge towards the Anchor node. Finally a test was run with both nodes facing towards the wall of the corridor and their edges facing each other. These orientations were chosen as they were representative of what is seen during the normal operation for ground vehicles. As the UGV moves in a place, perpendicular to the axis of rotation used in this test this is thought prudent.

4.2.2 Results and Analysis

The results from the range measurements in the first experiment in the corridor were saved along with the RSSI and Decibel values output from the Pozyx. In this case, RSSI refers to the Received Signal Strength Indicators, and Decibel refers to the dB or the radiated power received. First the absolute error of the range measurement was calculated for each sample. This was done by subtracting the ground truth range from all of the samples gathered for that measurement. An example distribution of an absolute range error sample may be seen in Figure 4.3.

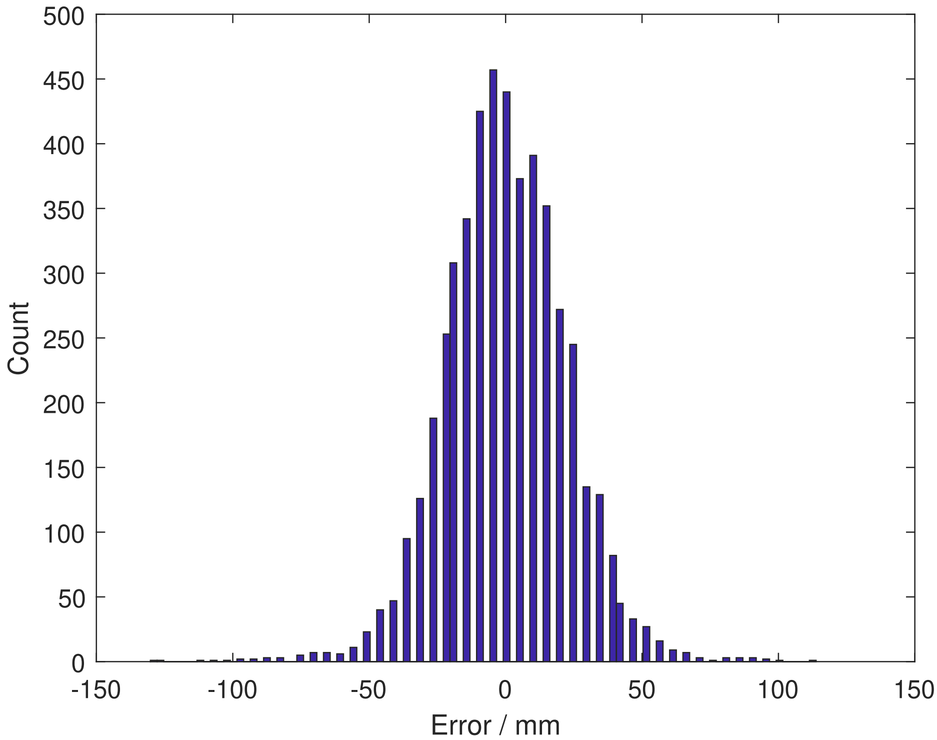


Figure 4.3: Static UWB range sample gathered from setup one.

Next, the standard deviation of the absolute error for each sample was calculated. The standard deviation plotted against the range may be seen in Figure 4.4 and Figure 4.5. It may be seen that there is a clear increase in the standard deviation of the readings as the range increases. This indicates that the reliability of the range measurement decreases as a function of the node to node distance. It was also seen in Figure 4.6 that there was no major change in the measurement dropout rate, however there was some decrease when the nodes are facing one another until the dropout rate settles out at the average of 2%. Increased dropouts would lead to fewer range measurements available for multilateration, decreasing the reliability of the position estimate.

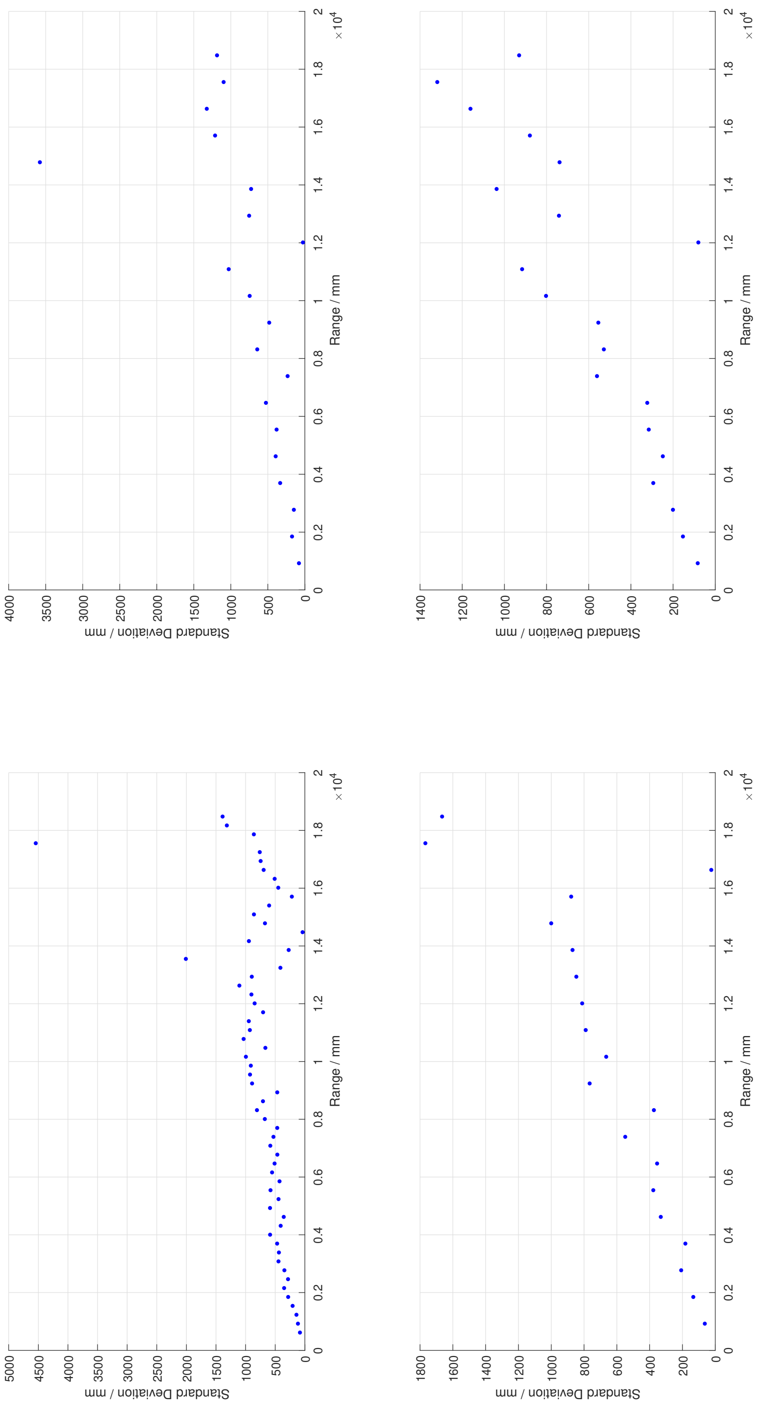


Figure 4.4: Various relative orientation UWB standard deviation as a function of range for experiment 1. Clockwise from top left : front to front, back to front, edge to front, edge to edge

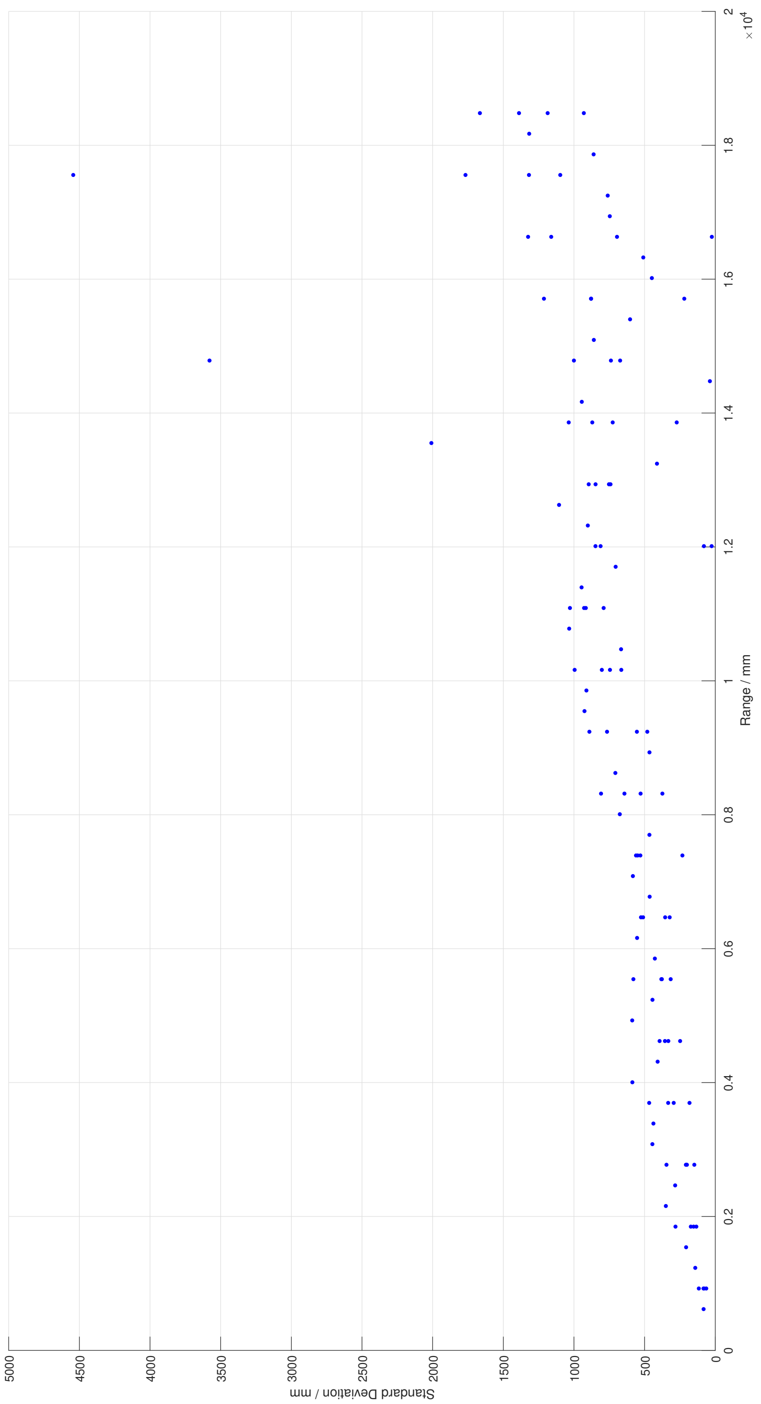


Figure 4.5: Combined orientation UWB standard deviation as a function of range for experiment 1.

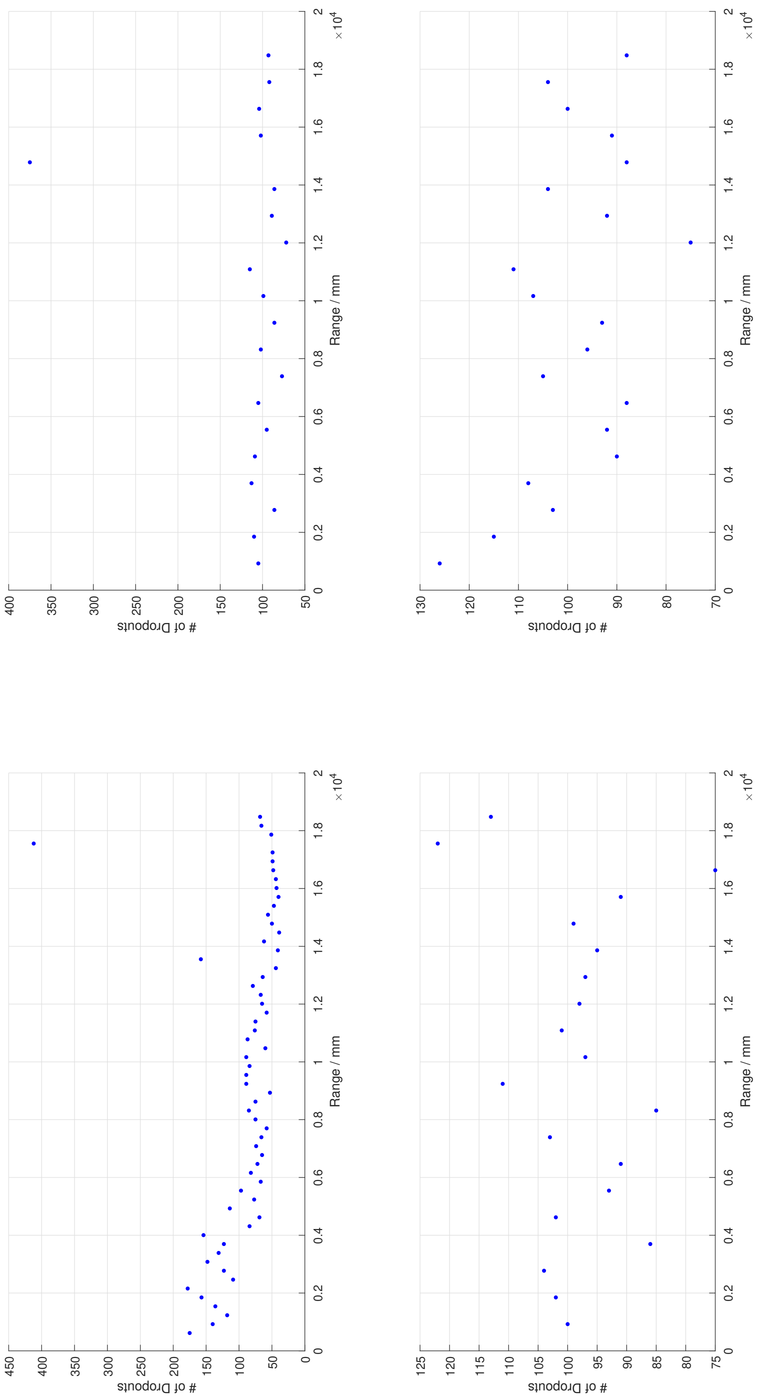


Figure 4.6: Various relative orientation UWB dropout rates as a function of range for experiment 1. Clockwise from top left : front to front, back to front, edge to front, edge to edge.

The findings shown from the first setup in a single orientation are confirmed in the same tests with varying node to node relative orientation as seen in Figure 4.4. The increase in standard deviation seen in the singular orientation is also seen in different orientations. However, at a range of 11m the stable increase in standard deviation seen at closer ranges becomes less coherent.

The second experiment demonstrated results similar to those found in the first experiment as seen in Figure 4.7. As can be seen Figure 4.8 when plotted on the same chart the two standard deviation results show a clear correlation, supporting the hypothesis that similar environments lead to similar UWB performance.

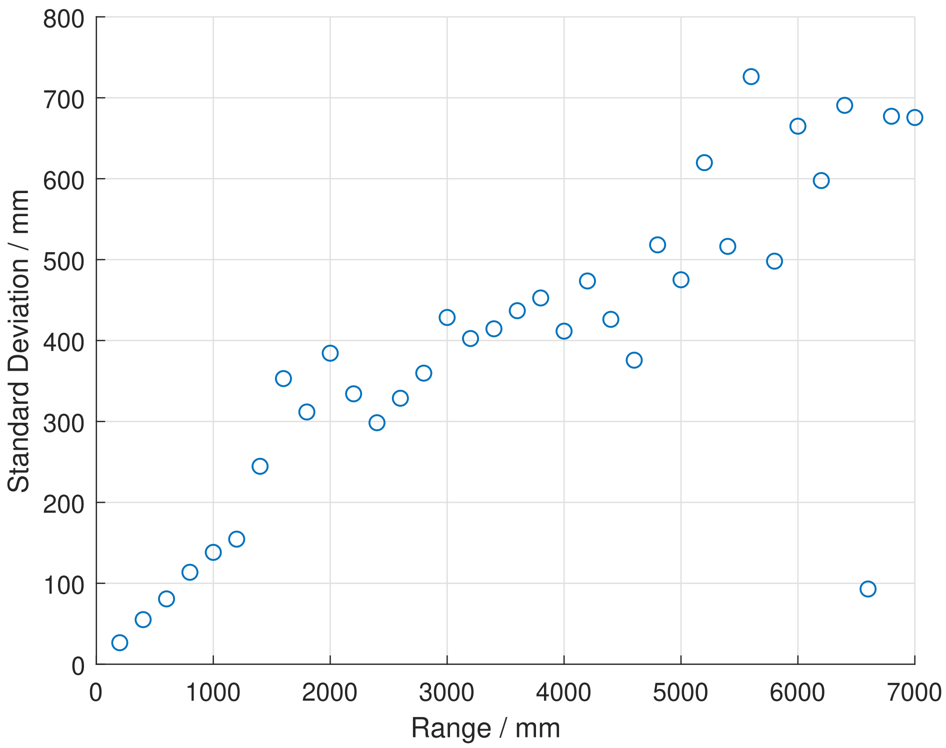


Figure 4.7: UWB standard deviation as a function of range for experiment 2 mounted on the UGV over 7m.

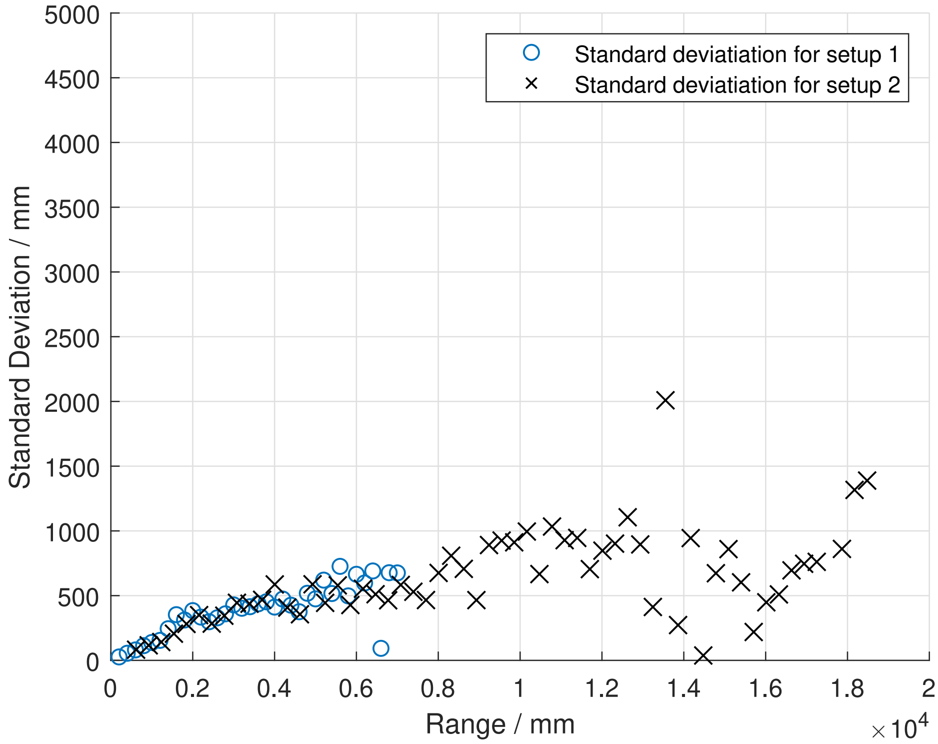


Figure 4.8: Comparison of UWB standard deviation as a function of range for experiment 1 and 2.

From the results gathered from the two setups we may see that the standard deviation of the sensor error follows a roughly linear relationship with the range between nodes. This allows us to generate a linear equation with coefficients based upon this empirical data. An appropriate fit may be generated through the use of the curve fitting tool in MATLAB. To determine the most suitable way to represent the relationship between variance and range a variety of fits were tested, the "goodness of fit" metrics "R-squared" and the "RMSE" may be seen in the table below. This was repeated for both the data over the two ranges in both setups as seen in Figures 4.9 and 4.10. For the second setup, as the data is available, all of the variances calculated for the various orientations are used. This is done to compensate for the erratic variances found at ranges beyond roughly 12m.

Fit number	Data used	Fit type	R-squared	RMSE
1	Setup 1	First order	0.8883	64.2436
2	Setup 1	Second order	0.8936	61.7575
3	Setup 2	First order	0.6408	211.7907
4	Setup 2	Second order	0.6438	211.8820

Table 4.1: Table of "goodness of fit" metrics for various sensor variance models.

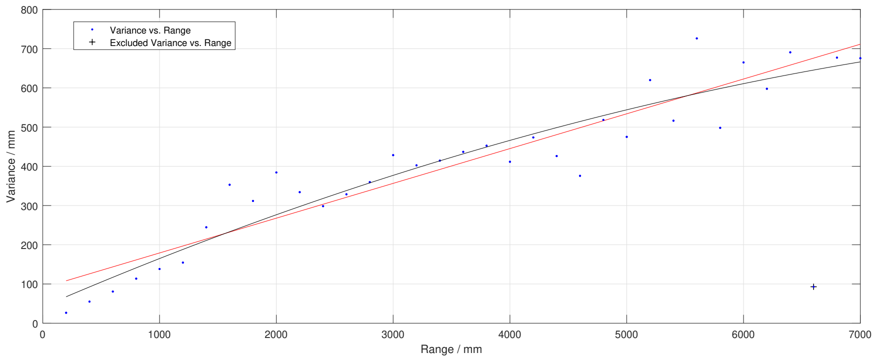


Figure 4.9: Comparison of first (red) and second (black) order polynomial fits for setup 1.

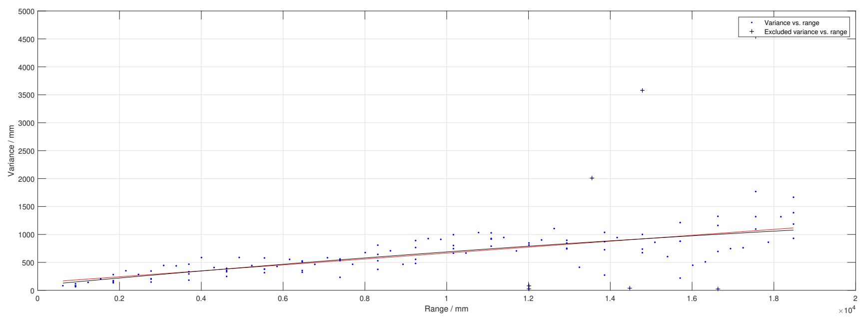


Figure 4.10: Comparison of first (red) and second (black) order polynomial fits for setup 2.

The results shown in Table 4.1 must be taken with a grain of salt. Although the RMSE and R-squared values are lower for the models constructed from the data obtained from experiment 1, this does not necessarily mean the model represents the data more accurately. The data gathered from setup 2 was over a shorter range and therefore contained fewer outliers as is expected with the

UWB system and described in the literature [104, 80, 120, 121, 122]. To be certain the best model is constructed from the observations made all of the data is combined and the process is repeated. The initial process has shown us the general form of the model we should expect. However, by incorporating all of the data the validity of the model is more reliable. Figure 4.11 shows the first order polynomial fit and residuals from all of the data collected in both the corridor, tripod based, and EKF testing environment, UGV based tests. the R-squared and RMSE values for this fit were found to be 0.6635 and 195.7853 respectively. These values are better than those found from the first order fit based solely upon the data collected from the UGV setup, however are not as good as those found from the tripod based setup. Although the goodness of fit criteria suggest a more effective representation is found from the data collected using the UGV alone, this model is only good for 7m.

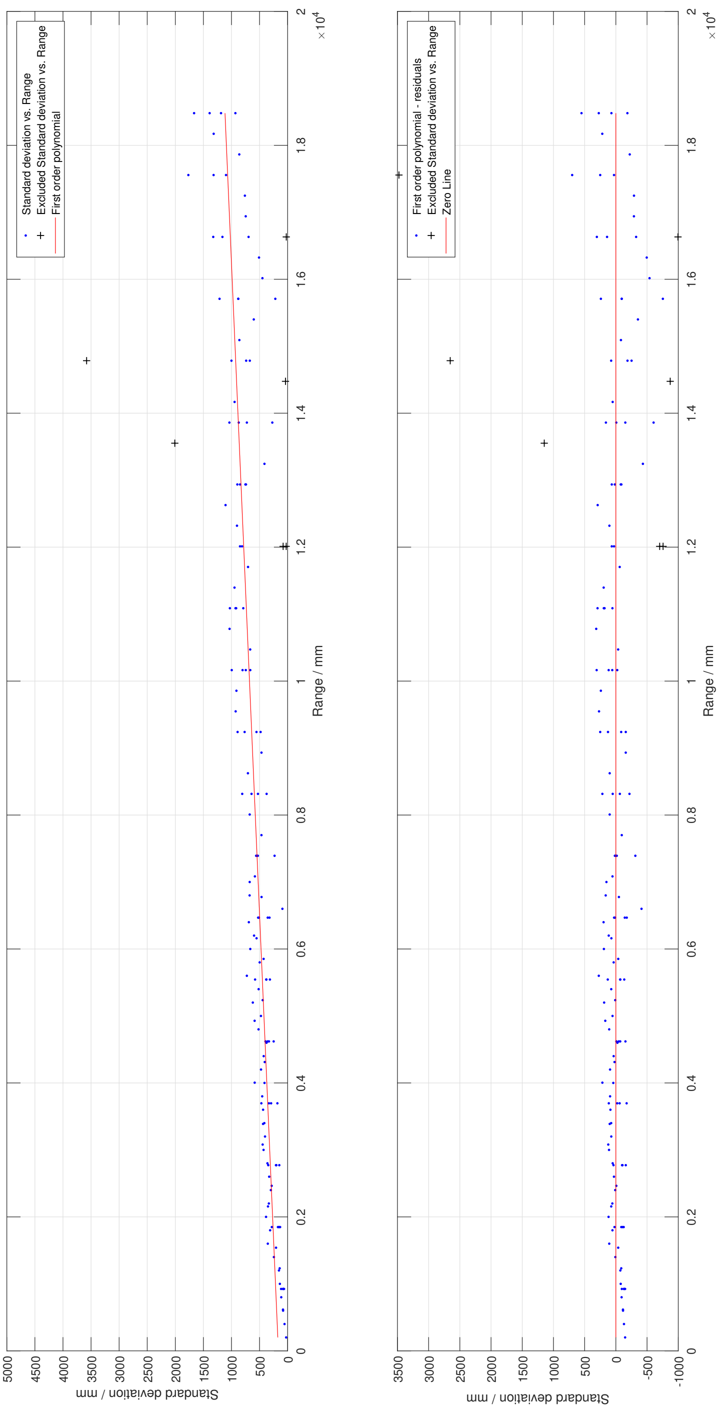


Figure 4.11: (Top) First order polynomial fit for for all of the data gathered. (Bottom) Residuals of first order polynomial fit for all of the data gathered.

4.2.2.1 Model Assessment

As the variance model constructed here provides an approximation of the experimental data it is logical to asses the model in comparison to real world data. The output of the model here is not a full distribution, but rather a single characteristic of an expected distribution. Therefore, the data for comparison may be extrapolated based on the assumptions of the filter. As the application is for an Extended Kalman Filter, with assumed Gaussian, White noise of zero mean. With these conditions, a distribution for comparison for any range modelled by the variance function may be constructed.

Next, a comparison criteria is needed. In inferential statistics, the comparison of two independent samples may be made with a "T-test". The T-test tests the null hypothesis that two distributions of the same mean are statistically similar to a degree of significance, usually 95 or 99 percent [123]. The test may be run in MATLAB, with a returned h and p value. In this case the h value is a binary value stating either true or false, a result of false indicates similarity between the data groups. The p value indicate the probability of this result occurring again. Therefore, a false h result may indicate similarity, however a low p value would indicate that this result is not robust. It should be noted that the results of the T-test are not a firm indication of the similarities; as noted by work on the topic T-test results often mislead, therefore in this work the T-test is used only as an indication of the similarity of the model to the experimental work [124].

To test the robustness of the constructed model, experimental data are gathered in the same manner as in the first experiment, through the use of tripods in a corridor. The T-test was run in comparison for each of the experimentally gathered range measurements and a model generated distribution of the same sample size (5000) was generated for the corresponding range. The results showed a false h value for every test run, indicating similarity. The results of the T-test in terms of the p values returned may be seen in Figure 4.12. An interesting point is that although the p value does not drop below 0.45 throughout the analysis the lower bound drops linearly as a function of range. This suggests that although the modelled distribution is similar, it becomes less representative as a function of range. This is most likely due to the erratic standard deviation values seen in the testing procedure as seen in Figure 4.4.

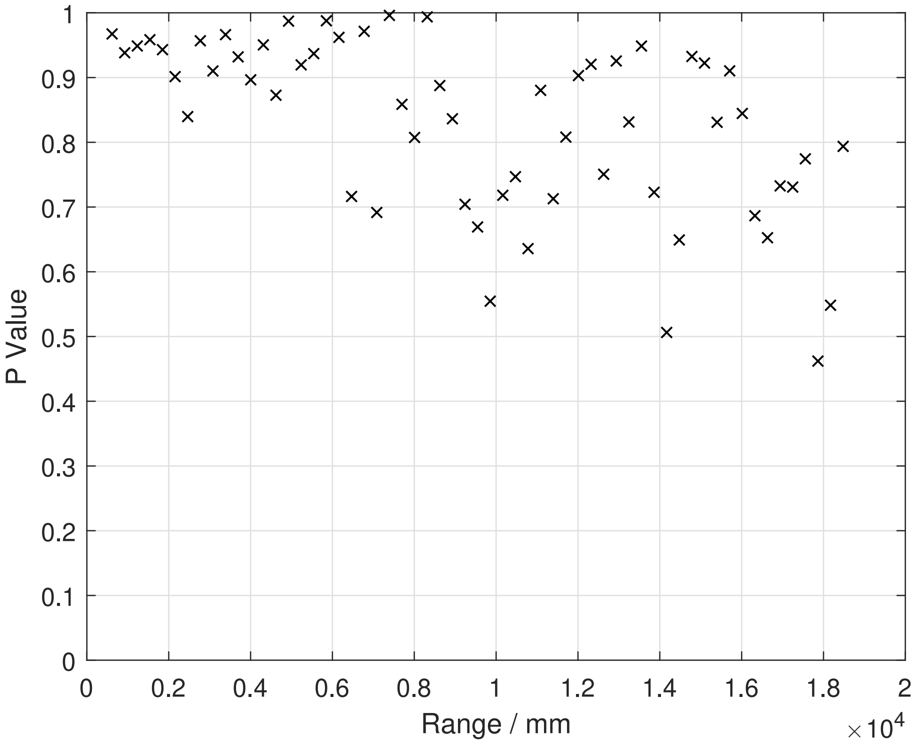


Figure 4.12: P value results of two sample T-test between experimental and model generated distributions.

4.3 Static Variance Versus Model Based Variance

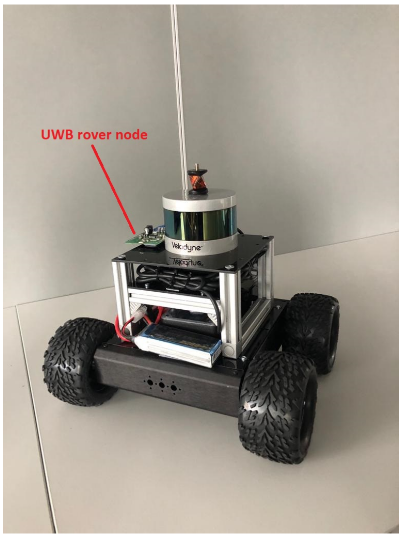
The principle focus of this experiment is the comparison of an EKF utilising a static sensor variance to that of one which uses a model based sensor variance. The purpose is to determine the validity of the hypothesis that a more robust representation of the sensor will improve state estimation. In this case, improvements may be taken as more smooth tracking, less error with respect to the ground truth, faster convergence and, a more robust system. The final point of a robust system refers to increased variation in sensor noise due to non-ideal circumstances such as NLOS or non-ideal rover node placement. To this end, as previously stated, the rover node is not given a purpose built mounting pole, but rather placed on the chassis of the UGV.

4.3.1 Methodology

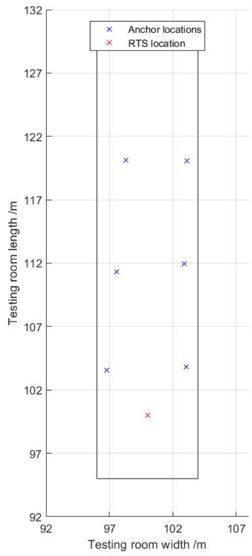
As previously stated in Section 4.2.2.1, the intended testing environment is indoors. As in the last chapter, six anchors were used, placed around the perimetre of the testing space using hook and loop tape onto tables as seen in Figures 4.13 and 4.14b.

Figure 4.13: The space used for testing with the UGV in the distance and the RTS.

No objects were in the testing space that would obscure the view leading to NCLOS conditions, aside from very oblique angles to the antenna. Due to the non-ideal placement of the rover node, as seen in Figure 4.14a and the environment, the UWB data was very noisy. Therefore to test the initial functionality of the model based variance in an ideal situation, synthetic data was constructed. The synthetic data was constructed using the ground truth readings collected from the RTS. The RTS data was then increased in frequency from 1Hz through linear interpolation to match that of the UWB system of 50Hz. Next a static noise was applied in line with the expected manufacturers stated value of $0.1m$ [97].



(a)



(b)

Figure 4.14: (a) View of UGV. (b) Testing room layout.

The synthetic data may be seen in comparison to the experimental data in Figure 4.15. This data was used as an initial verification data set to determine the functionality of the MBEKF. The functionality is defined as the EKF's capacity to generate a consistent trajectory in the space which is reasonable for the input data used. Once the MBEKF was determined to function, the system was run again with the experimentally gathered data.

The motion model of the system is unchanged from that used in the previous chapter as the UGV kinematics are again that of a skid steer rover. The quadrature encoders are of a different source, however the data output is the same. The number of ticks per revolution is all that is altered to 360.

4.3.1.1 Measurement Model

In normal circumstances, the sensor variance is taken into account at the calculation of the Kalman gain seen in Algorithm 2 on line 8. However in this case, the sensor variance must be calculated and therefore is added as a step prior on line 7 of this figure. As the UWB observation may contain outliers which would affect the sensor variance to a high degree that predicted state $\widehat{Z}_k^-$ is used to calculate the variance. This means that drastic variations in the observation will not skew the system's trust in the sensor.

Algorithm 2 Range based MBEKF Localisation

Prediction:

$\widehat{X}_k^- = f(\widehat{X}_{k-1}, u_k)$

$G_{x_k} = \frac{\partial f}{\partial x}(\widehat{X}_k^-, u_k)$

$G_{u_k} = \frac{\partial f}{\partial u}(\widehat{X}_k^-, u_k)$

$\widehat{P}_k^- = G_{x_k}\widehat{P}_{k-1}G_{x_k}^T + G_{u_k}QG_{u_k}^T$

Correction:

$\widehat{Z}_{Bi} = h(\widehat{X}_k^-, A_{B_i})$

$H_{z_k} = \frac{\partial h}{\partial x}(\widehat{X}_k^-)$

$\boldsymbol{R_k = V(\widehat{Z}_k^-)}$

$K_k = \widehat{P}_k^- H_{z_k}^T (H_{z_k}\widehat{P}_k^- H_{z_k}^T + R_k)^{-1}$

$Y_k = Z_{Bi} - \widehat{Z}_{Bi}$

$\widehat{X}_k = \widehat{X}_k^- + K_k Y_k$

$\widehat{P}_k = (I - K_k H_{z_k})\widehat{P}_k^-$

if *measurement_is_available* **then**

    do **Correction**

else

    do **Prediction**

end if

4.3.2 Results and Analysis

As expected the UWB system did not perform as well as that shown in the previous chapter. As seen in Figure 4.15a the UWB data was highly noisy and drifted considerably over the testing trajectory, this was expected due to the placement of the rover node and the new environment. The synthetic data shown in Figure 4.15b allowed for a good initial tuning of the EKF parameters, notably the process noise, Q. The resultant track seen in Figure 4.16 shows a smoother track of the position than that provided by the UWB system and a track closer to that of the RTS collected ground truth.

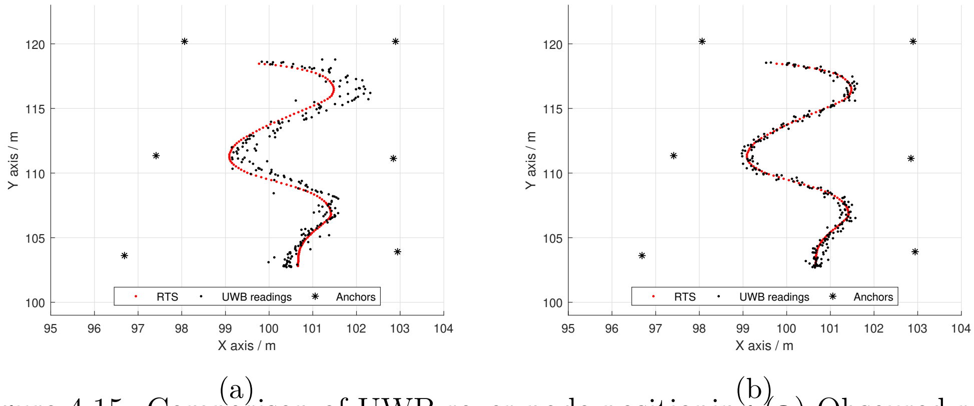


Figure 4.15: Comparison of UWB rover node positioning (**a**) Obscured rover Node; (**b**) Simulated Un-obscured rover Node.

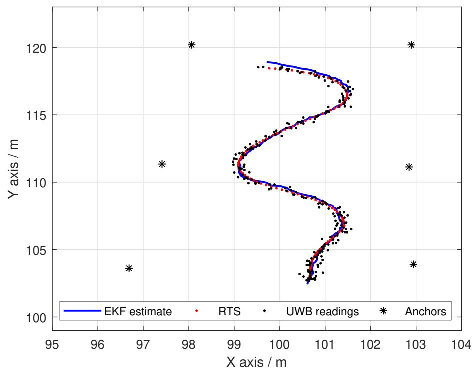
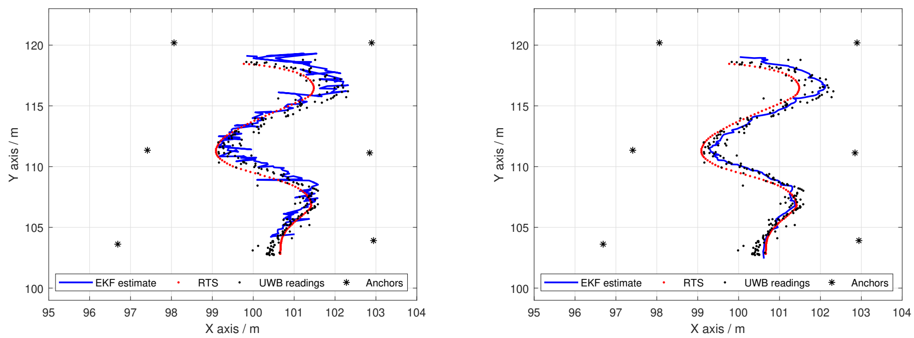


Figure 4.16: Model-based EKF trajectory using simulated UWB data.

The EKF developed in the previous chapter did not perform as well as previously when used to estimate the position of the platform in non-ideal UWB environments as seen in Figure 4.17a. Throughout the estimation the EKF trajectory was very noisy and only marginally improved upon the position estimate of the UWB, however the resultant trajectory was not similar in translation to that seen in the platform, with the estimated position jumping left and right in relation to the direction of travel. On the other hand, as seen in Figure 4.17b, trajectory generated from the EKF using the model based sensor variance was significantly smoother. Although the estimated trajectory was still noisy it did not demonstrate the same offsets as seen in the EKF using the static sensor variance. A comparison of the Euclidean distances between each logged point for the EKF , MBEKF and the ground truth may be seen in Figures 4.19 and 4.20. In these figures, the ground truth positions are increased in frequency through linear interpolation to provide a point to point comparison. It is clear from this plot that the standard EKF is much more erratic in its track showing peaks of up to 1.2m between points, significantly above that seen in the ground truth or the MBEKF.



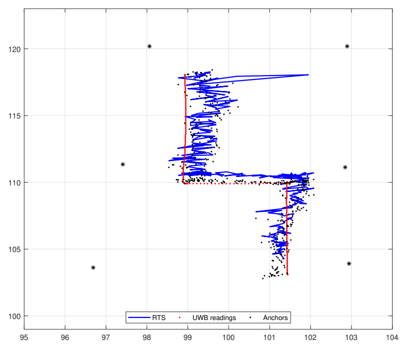
(a) (b)

Figure 4.17: Trajectory 1. Comparison of EKF with a generalised variance to an EKF using a model based one, with an obstructed UWB rover node. (**a**) generalised variance; (**b**) Model-based variance.

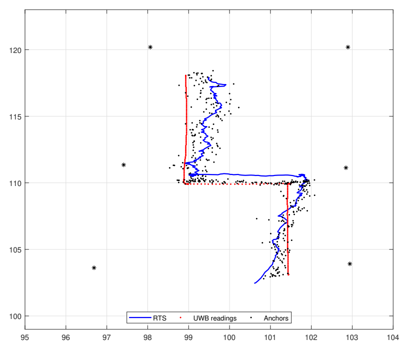
There is still a noticeable offset to the right of the ground truth track towards the top of the trajectory. This is most likely due to the offset of the UWB data. Although it may be possible to tune the EKF parametres to account for this, it is unlikely that this effect would be carried through to other trajectories. In this case to avoid over optimisation of the EKF for certain trajectories and circumstances the offset was considered unavoidable without less noisy UWB data.

It must be stated that an EKF may be able to improve the positional estimate of a single sensor, it is not possible for it to estimate the position of an agent if that information is not contained within the current and past observations. In other words, there is a limit to that which may be estimated, based upon the relevant data available.

This improvement is seen again in Figure 4.18, where a different trajectory is shown. In this example, the trajectory is particularly challenging as it incorporates 2 sharp 90 deg turns. In these situations, the encoders are most unreliable due to slippage, therefore, the prediction phase is liable to offset from the ground truth. Again the UWB is particularly noisy in this situation, as already mentioned, the MBEKF compensates well in comparison to the EKF using the generalised sensor variance. It is important to note that the EKF parametres such as system noise are not tuned between trajectories to give a better result, as this project is not focused on the development of a "perfect" estimator, but rather the effects of a model based sensor variance upon the reliability of said estimator.



(a)



(b)

Figure 4.18: Trajectory 2. Comparison of EKF with a generalised variance to an EKF using a model based one, with an obstructed UWB rover node. (**a**) generalised variance; (**b**) Model-based variance.

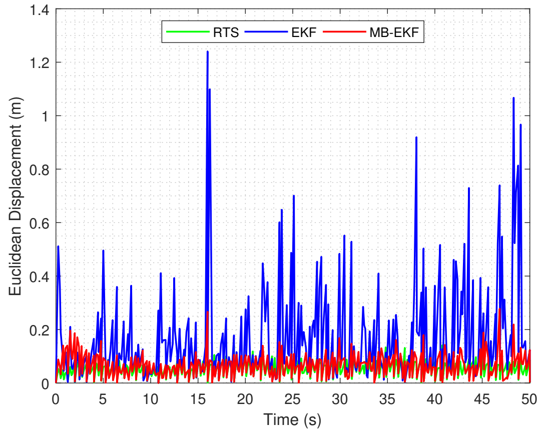


Figure 4.19: Comparison between Euclidean Distance Estimations for trajectory 1.

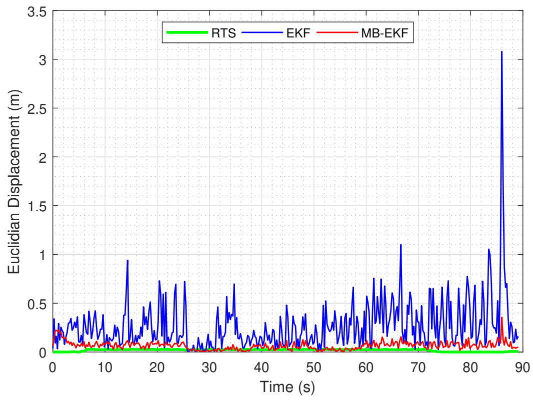


Figure 4.20: Comparison between Euclidean Distance Estimations for trajectory 2.

4.4 Conclusions

The work in this chapter demonstrates a method by which a model of the sensor variance may be constructed and implemented into an EKF . This model has been proven to produce range measurement distributions that are statistically similar to that of range measurement distributions obtained through experimentation. For this section of the investigation, the experimental space has been moved indoors, and the UWB antenna has been moved to a non-ideal location. This was done first to investigate the EKF performance in a range of spaces and second, to test the effects of noisy sensor data on the EKF performance. The initial testing and tuning of the EKF was conducted through the use of synthetic data to allow for a less biased tuning of the EKF parameters and avoid over fitting the filter to data. As shown, the model based EKF performs more reliably than the EKF developed in the previous chapter, and is more resilient to noisy measurements. An important point to note is that although the Model Based EKF improves trajectory estimation in noisy environments, it performs very well under reduced noise measurement input.

The next chapter will further develop the EKF to other platforms and move away from 2D systems. It has been shown that rather than the traditional route of further specifying the system model a more thorough model of the sensor behaviour can be used to improve the track in noisy conditions. To extend this principle, the next EKF will use a 6-DOF model as demonstrated in Chapter 2. This will generalise the EKF applications to systems such as UAV s, while also possibly maintaining effective tracking for UGVs.

Chapter 5

6 Degrees of Freedom Motion Estimation For an Unmanned Aerial Vehicle

From the previous chapters we have shown that an improved understanding of the sensor variance will improve the estimate provided by a state estimator. Furthermore, we have demonstrated that the integration of the characterised sensor uncertainty into a variance model is possible. The previous chapter shows this model based variance improves both the estimate and estimator's tolerance to abnormally noisy data input.

The main drawback of the algorithms previously discussed is that they are only applicable to specific dedicated platforms. Although it is shown that under these circumstances the system operates effectively, the state estimator system model requires reformulation for each platform in use. Also this process is required for a given environment which introduces changes not incorporated or accounted for in the original system model. An ideal system would be capable of estimating the states of several different types of platforms without modification. This would allow the systems' operation to be verified thoroughly, a single time. In cases such as in the UK, the question of the legal requirements to be placed upon a UAV for use in BVLOS or autonomous operation are still being developed. However, the documentation issued by the CAA in CAP722 for the purposes of governing and guiding such deployments places a requirement upon the integrity of the positional data used [125]. As stated previously, this would require the verification of the state estimator, however for example if a platform was modified to utilise different sensors or control systems, its state estimator and navigational sensors may change. Consequently this would require the verification of the new system. However, this may be avoided if a self contained sensor package and estimator were employed, that may be transferred between systems. For such a tool to function, the system model would need to be flexible enough to be transferable between many types of platform. This would therefore compromise the efficiency of the state estimator in circumstances where a more specific system model is available. To compensate for this reduced level of model specificity, the improvements found in the previous chapter may be employed.

To this end, this chapter demonstrates a method by which a more holistic system model may be employed in the estimation of the states of both an aerial and ground platforms. Under normal operation this would lead to a radical decrease in the accuracy of the prediction for both platforms. However, as demonstrated in the previous chapters, the estimator performance may be improved through a more thorough characterisation of the sensor variance; this chapter will further investigate the concept of relying upon improved sensor characterisation as opposed to specific system modelling.

5.1 State of the Art

Work conducted in the field of multirotor UAV localisation by the Kyoto Institute of technology presents a method for the fusion of IMU, UWB and distance measuring sensors [126]. In this work, an Extended Kalman Filter was implemented and was shown to successfully fuse the readings between an UWB system and the IMU with altitude corrections provided by a Time-of-Flight laser-ranging module (VL53L0X) [126]. Limitations in this system include a short operating range for the laser range finder (2 m) and no verified ground truth for the resultant EKF performance assessment [126]. The results presented also show that the testing area for the system was relatively small ($1.5 \times 1 \times 1$ m) in comparison to usual operating environments for UAVs [126] [127]. This work shows that conventional Extended Kalman Filtering is possible, however also demonstrates the need for more robust assessment criteria.

Work has been conducted into the combined use of camera and laser range finders, however this system again requires suitable lighting conditions and significant processing capabilities in comparison to other work with more sparse measurement data such as GPS data [51]. This approach does, however, aid in the reduction of drift as direct measurement of the position is made, assuming a robust laser scan matching system is employed. This, in conjunction with a camera providing direct velocity measurements, allows for a more robust update phase [51]. Non-linearity in the system was shown to be acceptably dealt with through the use of an EKF .

Temakek Laboratories has demonstrated the application of laser range finders as a measurement device for the localisation of UAVs in GPS denied cluttered spaces [11]. This implementation combines a 2D laser scan feature matching technique with magnetometer measurements to provide updates to the position and heading of the flying platform, using the IMU as control inputs for heading and velocity updates. No mention is made in terms of how non-linearity is dealt with, therefore it may be deduced that this approach would be highly dependent upon the trajectories it is tested on. As there is no update made for the altitude of the agent it is also plausible that the system would not be robust in 3D trajectories. This is confirmed by their results, which although show reasonable tracking in a plane, show noisy estimation in the z axis.

Work has been conducted into the autonomous operation of aerial vehicles in cluttered environments [128]. Approaches include the use of monocular cameras with IMU fusion, however this is dependent upon lighting conditions. The implementation shown in this case uses the camera to provide a 3D velocity vector, this leads the author to comment on the observability of the system. In this case no direct measurement of the position is being made so all updates are relative to the body frame of the system. This leaves this method susceptible to drift.

5.2 Formulation

As previously discussed the state of the ground based system are comprised of position x, y and heading ϕ. In this formulation the state must take into account new degrees of freedom. As the intended platform may include a multirotor UAV , it is necessary to include the positions x, y and z. This approach also allows for terrain elevation changes to be taken into account in ground rover platforms. The expected measurement input for the system is from an IMU and the UWB system, therefore states for change in position (velocity), and orientation are needed. The need for orientation is especially required here as the platform is capable of moving freely in all directions and axis. Therefore the transformation between the body and inertial reference frames is vital. The orientation is represented using quaternions, therefore the states of the system are given as matrix X as shown in Equation (5.1).

$$X = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & x' & y' & z' & x & y & z \end{bmatrix}^T \tag{5.1}$$

The state transition function may be taken as that shown in Chapter 2. As previously discussed, the state transition function will incorporate the IMU measurement as control inputs. The process noise w of the system will therefore use the variance of the IMU as a foundation to reduce the time needed to tune the system.

The state transition function was coded within the EKF loop as seen in pages 3, 4 and 5 of the MATLAB script seen in the appendix to this work.

5.2.1 UWB Measurement Model

As described, the working space of the new system is 3D, being comprised of x, y and z. In previous work the measurement model of the UWB input was defined as shown in Equation (5.2). However, as this form only represents the system in two dimensions it must be modified. As described, the multilateration model may be modified to allow 3D range calculation as shown in Equation (5.3).

$$D = \sqrt{(x_k - x_{Ai})^2 + (y_k - y_{Ai})^2} \tag{5.2}$$

$$D = \sqrt{(x_k - x_{Ai})^2 + (y_k - y_{Ai})^2 + (z_k - z_{Ai})^2} \tag{5.3}$$

Following the process defined in earlier chapters, the Measurement Jacobian may be formulated as Equation (5.4). Again, as in previous chapters the assumption is made that all six UWB anchors provide range measurements for each measurement update. This allows for the construction of the 10x6 Jacobian matrix as shown in Equation (5.5)

$$H_{z_k} = \frac{\partial h}{\partial x}(\widehat{X}_k^-), \tag{5.4}$$

$$H_{z_k} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x-2\,\mathrm{xb}_1}{2\,\sqrt{(x-\mathrm{xb}_1)^2+(y-\mathrm{yb}_1)^2+(z-\mathrm{zb}_1)^2}} & \frac{2\,y-2\,\mathrm{yb}_1}{2\,\sqrt{(x-\mathrm{xb}_1)^2+(y-\mathrm{yb}_1)^2+(z-\mathrm{zb}_1)^2}} & \frac{2\,z-2\,\mathrm{zb}_1}{2\,\sqrt{(x-\mathrm{xb}_1)^2+(y-\mathrm{yb}_1)^2+(z-\mathrm{zb}_1)^2}} \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x-2\,\mathrm{xb}_2}{2\,\sqrt{(x-\mathrm{xb}_2)^2+(y-\mathrm{yb}_2)^2+(z-\mathrm{zb}_2)^2}} & \frac{2\,y-2\,\mathrm{yb}_2}{2\,\sqrt{(x-\mathrm{xb}_2)^2+(y-\mathrm{yb}_2)^2+(z-\mathrm{zb}_2)^2}} & \frac{2\,z-2\,\mathrm{zb}_2}{2\,\sqrt{(x-\mathrm{xb}_2)^2+(y-\mathrm{yb}_2)^2+(z-\mathrm{zb}_2)^2}} \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x-2\,\mathrm{xb}_3}{2\,\sqrt{(x-\mathrm{xb}_3)^2+(y-\mathrm{yb}_3)^2+(z-\mathrm{zb}_3)^2}} & \frac{2\,y-2\,\mathrm{yb}_3}{2\,\sqrt{(x-\mathrm{xb}_3)^2+(y-\mathrm{yb}_3)^2+(z-\mathrm{zb}_3)^2}} & \frac{2\,z-2\,\mathrm{zb}_3}{2\,\sqrt{(x-\mathrm{xb}_3)^2+(y-\mathrm{yb}_3)^2+(z-\mathrm{zb}_3)^2}} \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x-2\,\mathrm{xb}_4}{2\,\sqrt{(x-\mathrm{xb}_4)^2+(y-\mathrm{yb}_4)^2+(z-\mathrm{zb}_4)^2}} & \frac{2\,y-2\,\mathrm{yb}_4}{2\,\sqrt{(x-\mathrm{xb}_4)^2+(y-\mathrm{yb}_4)^2+(z-\mathrm{zb}_4)^2}} & \frac{2\,z-2\,\mathrm{zb}_4}{2\,\sqrt{(x-\mathrm{xb}_4)^2+(y-\mathrm{yb}_4)^2+(z-\mathrm{zb}_4)^2}} \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x-2\,\mathrm{xb}_5}{2\,\sqrt{(x-\mathrm{xb}_5)^2+(y-\mathrm{yb}_5)^2+(z-\mathrm{zb}_5)^2}} & \frac{2\,y-2\,\mathrm{yb}_5}{2\,\sqrt{(x-\mathrm{xb}_5)^2+(y-\mathrm{yb}_5)^2+(z-\mathrm{zb}_5)^2}} & \frac{2\,z-2\,\mathrm{zb}_5}{2\,\sqrt{(x-\mathrm{xb}_5)^2+(y-\mathrm{yb}_5)^2+(z-\mathrm{zb}_5)^2}} \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x-2\,\mathrm{xb}_6}{2\,\sqrt{(x-\mathrm{xb}_6)^2+(y-\mathrm{yb}_6)^2+(z-\mathrm{zb}_6)^2}} & \frac{2\,y-2\,\mathrm{yb}_6}{2\,\sqrt{(x-\mathrm{xb}_6)^2+(y-\mathrm{yb}_6)^2+(z-\mathrm{zb}_6)^2}} & \frac{2\,z-2\,\mathrm{zb}_6}{2\,\sqrt{(x-\mathrm{xb}_6)^2+(y-\mathrm{yb}_6)^2+(z-\mathrm{zb}_6)^2}}
\end{pmatrix} \tag{5.5}$$

5.3 Robotic Total Station UAV tracking

As has been previously demonstrated in this work, the RTS is reliable and effective tool for acquiring the ground truth position of a UGV platform [59, 129]. In the previous chapters, work was conducted to determine an effective setup for using the RTS. As this section of the work now includes large translation in the vertical axis, and the platform to be tracked is now also more volatile in its possible manoeuvres, it is logical to reassess the suitability of this tool.

To begin, the Leica micro prism was again chosen to be used. This was decided as in terms of mass and profile as it was small enough to be non-intrusive when included as a payload in a UAV platform. This prism is stated to allow for passive tracking in a similar manner to the Trimble targets mentioned in the previous chapter. The stated measurement range of the S7 RTS is 2500m, or 5500m in long range mode [96]. The Trimble S7 RTS is stated to be capable of tracking passive prisms such as the Leica used here out to between 500m and 700m [96]. The weather conditions for this track have been defined as "clear, no haze, overcast or moderate sunlight with very bright heat shimmer" [96]. Therefore, from the fact that the weather conditions are described in the tracking range it is reasonable to assume this has an effect. As has been shown, RTS tracking is normally reliable, however as the prism was of a small size, it was felt that sudden movements may cause a loss of "lock".

A feasibility test was conducted outdoors in a field near Poulton, Cheshire, the testing location may been seen in Figure 5.1. The prism was mounted on the rear side of the UAV clear of obstructions. To avoid obscuring the line of sight between the RTS and the prism, the UAV was piloted with a constant orientation in Yaw in relation to the RTS. This reduces the possibility of landing legs interfering with the measurements. The UAV platform with the prism may be seen in Figure 5.2. The flight was conducted out to the maximum range comfortably attainable within the legal UAV operating principles, therefore the maximum range of this test was determined to be 400m prior to take off. The weather on the day of testing was clear, with some cloud cover and no haze, similar to that described in the Trimble specifications [96]. Therefore similar accuracy to that described would be expected.

Figure 5.1: Satellite view of testing space. Red lines indicate pathways and roads, yellow/green hatching indicates acceptable landing and take-off areas. Yellow box indicates flying space. Source: Google Maps.

The results of RTS tracking are presented in Figure 5.4. It can be seen in Figure 5.4, it is clear that the RTS is capable of reliable tracking of the UAV at extended ranges of up to 300m. There are two sections where the RTS lost its lock on the mounted prism; this was caused by pilot error as the UAV rotated about its Yaw axis and the prism became obscured. The lock was regained through the use of the video streaming feature of the RTS and remote control of its orientation using the TSC3, as seen in Figure 5.3. The RTS maintained lock under the complex rastering manoeuvres of the UAV , as seen in Figure 5.5. As this was an initial survey of the RTS' capabilities

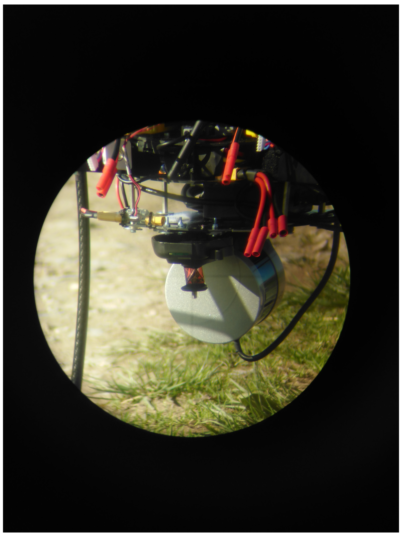


Figure 5.2: B.O.B UAV with mounted Leica prism. View taken from down RTS telescope while locked.

for tracking UAV s, the UAV sensor output was not saved during this test as it would require extended ranges of WiFi access points that are not needed elsewhere in the project.

Figure 5.3: B.O.B UAV with mounted Leica prism. View taken from down RTS telescope while locked.

An important point to note in the readings obtained from the RTS is the limitation of the tracked platform's velocity. As the RTS measurement is relatively slow (1Hz) in comparison to the possible velocities of a UAV (18m/s) then care must be taken when specifying the operating velocities of the platform in question [130]. As may be seen in Figure 5.5, the points measured are dense in areas where the UAV reduces velocity to manoeuvre, however in areas such as the ascent where the velocity is high only four points are measured over a 4m distance. This is also seen in the 2D overview of the whole flight shown in Figure 5.4. The translations at range above 250m in Y and 200m in X are fairly dense. However, the flight path away from the pilots seen just below this was carried out at high speed, leading to fewer points generated along the trajectory. In cases such as this, assuming the UAV is moving with high enough momentum to be robust to wind gusts, linear interpolation may be used, however this does leave room for error. Therefore in the future use of this system the UAV will be piloted at lower velocities. This velocity threshold will be found through trial flights and analysis of tracked data sets.

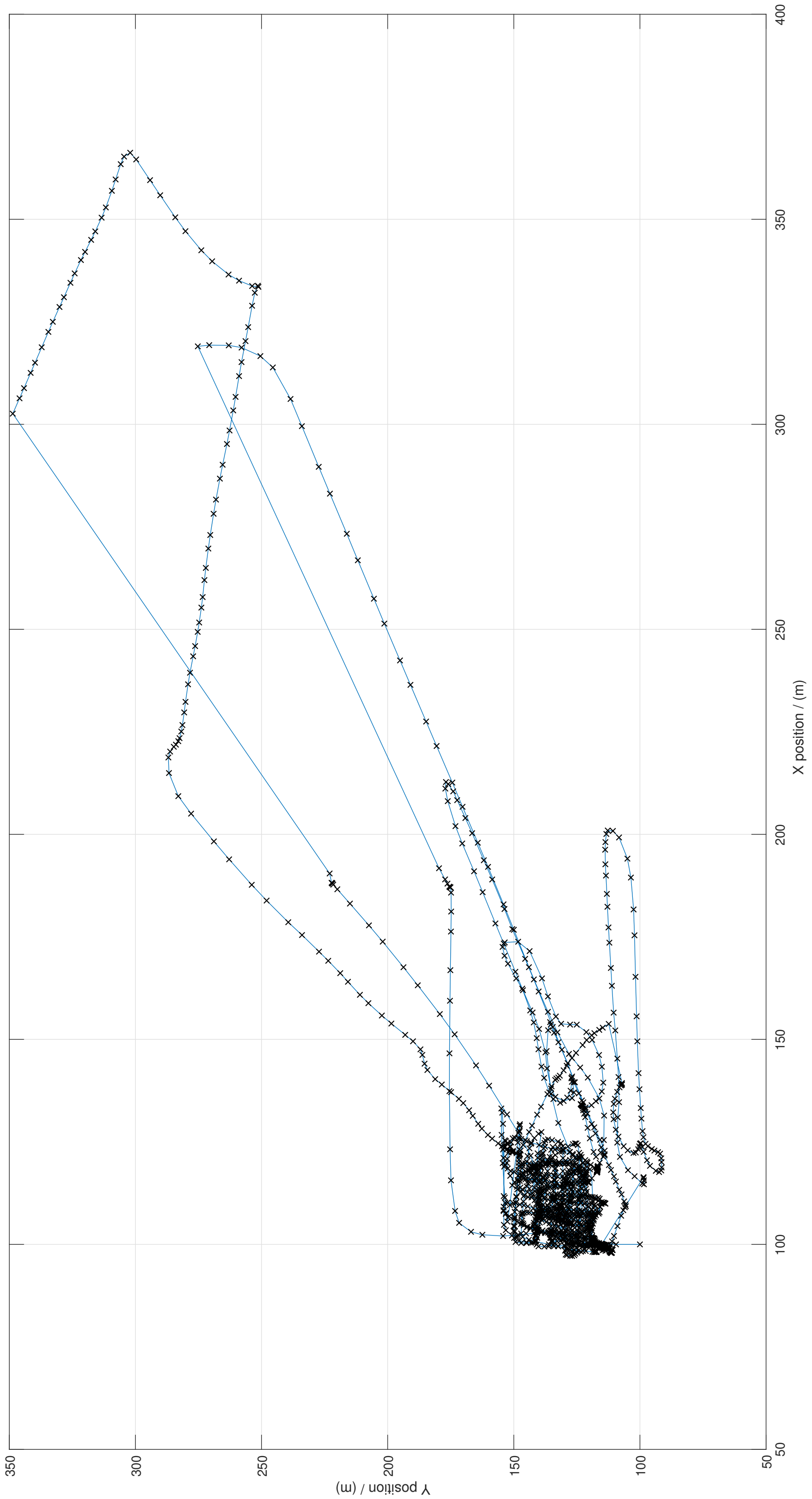


Figure 5.4: Overall view of the measurements taken using the RTS of the UAV trajectory.

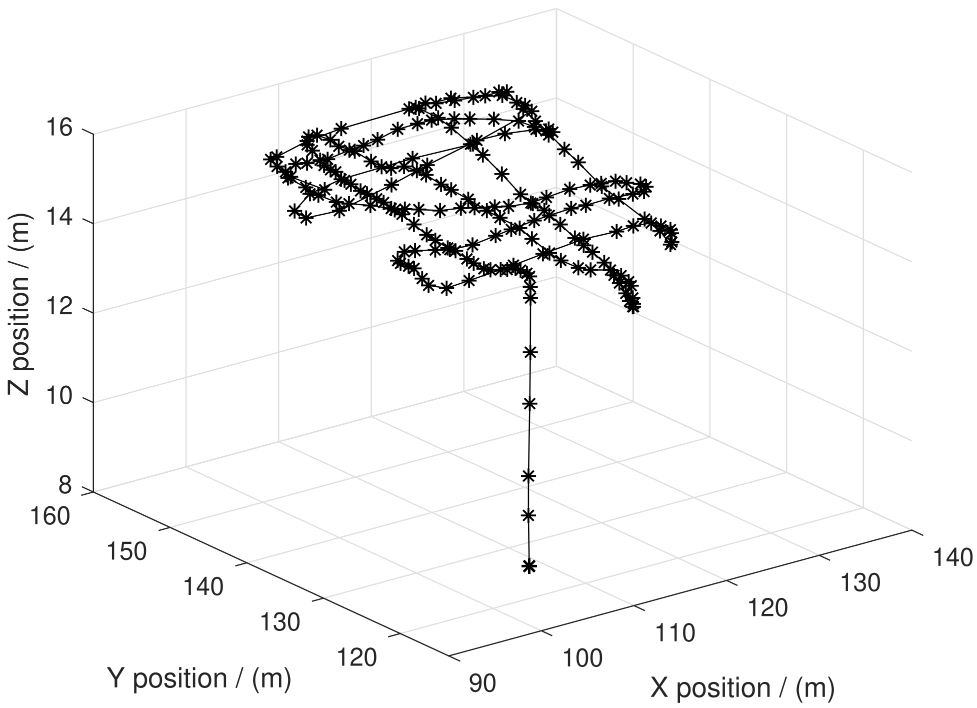


Figure 5.5: 3D plot of measurements taken using the RTS of the UAV trajectory for small raster manoeuvre.

5.4 Experimental Setup

5.4.1 Ground Rover

As in previous experiments the P.E.R.C.I platform was used. For the 6-DOF EKF as defined the sensors used are the BNO055, and the Pozyx system. For this investigation it was desirable to have a data collection system that could be used in the UGV and UAV platforms with minimal alterations. Therefore, the ROS packages and sensor connections were handled onboard a Raspberry Pi 3 connected via wifi access point to the laptop ground station also serving as RTS host. In this case it was desirable to continue to use an RTS prism that could be transferred between platforms, so the Leica Mini Prism was again employed.

5.4.2 Unmanned Aerial Vehicle

For the aerial components of the testing, a F550 "Flame Wheel" hexcopter was used. The system was small enough to allow for testing indoors safely, however large enough to practically carry the required payload. The expected flight time for the F550 was at maximum 15 minutes, with a maximum wind tolerance of 17 knots.

5.4.2.1 UAV Configuration

The UAV was constructed using the standard propulsion components that come with the kit. This includes the OPTO DJI E300 40A Electronic Speed Controller (ESC)s and the DJI 2212/920kV BLDC motors. This was selected as this platform configuration has been used previously and has a demonstrated track record of reliability and payload capacity (2.5Kg demonstrated Maximum Operating Mass (MOM)). The flight controller used was a Pixhawk V2.1 Flight Control Unit (FCU) running the PX4 flight stack. This was chosen over the DJI Nava-M V2 controller as it allows for far greater access to the flight data, configuration and sensor integration. The Pixhawk flight controller allows for serial communication based upon the MAVLink protocol. The FCU was then connected via serial to the onboard Raspberry Pi 3 using MAVROS ROS nodes utilising the MAVLink protocol. This allowed for control if required of the UAV via the ROS network already in use. The Pi was then connected to the BNO055 IMU via I2C, which was mounted at the rear of the top frame plate. This location was chosen as it allowed for easy access to the IMU, and was away from the main power distribution board of the UAV to reduce the chances of interference. The Pozyx was mounted onto the bottom frame plate of the

UAV using hook and loop tape, and connected to the Pi using a standard USB lead. This location was chosen to avoid obscuring the rover node in operation

5.4.3 PID Tuning

As stated, the UAV in use is controlled via a Pixhawk V2.1 flight controller. When using the PX4 firmware stack the Pixhawk system utilises a PID controller in an inner-loop, outer-loop PID attitude stabilisation system. The outer loop of the controller deals with the orientation error, while the inner loop controls the rate section of the control axis. In this system, the PID systems control the Pitch, Roll, Yaw and Throttle axes. Alteration of the system PIDs is done by connecting the UAV to "Mission Planner" an open source control and configuration tool. This connection is made via USB serial to the laptop.

The order of PID axis tuning is important for a multirotor. If the Yaw axis is unstable, then tuning of the Roll and Pitch axis may lead to errors due to cross-coupling, or joint axis movement. By stabilising the direction the UAV is facing, the Roll and Pitch axes may be tuned with minimal effect to each other. The tuning process used here is manual, and carried out in an indoor environment. This situation along with the multirotor's performance constraints impose certain restrictions upon the tuning procedure. For example, testing of the I component of the rate PID system would be done by applying continual demand on the rate of an axis, for example Roll. This would allow for a better idea of how the PID reaches convergence, and if a greater I value is needed. This is not possible in this situation, therefore the best approximation was made to fly the UAV in a sinusoidal manner along the axis being tuned. This allowed for the greatest range of setpoints, while also summing to a change of zero in position, ideal for restricted space. As the space was also constrained in terms of altitude, the throttle stabilisation element of the system was not tuned. While the UAV operates in a controllable manner in this testing space, the close proximity of the walls and the presence of ground effect mean that the altitude stabilisation must be reassessed in the designated testing space. If the operation of the UAV is considered safe then testing will continue, however it may be that the platform undergoes further tuning to improve altitude stabilisation.

The procedure for tuning was carried out using the expected payload, and battery type, charged to a minimum of 50% throughout the flight. The space was enclosed with a net, as may be seen in Figure 5.6. The default tuning parametres of the multirotor were close to stable, therefore the tuning process was relatively short (approximately three hours). Flight of the UAV was carried out in "Stabilise mode" which gives rate control of Yaw, Pitch, and Roll axes. The pilot controls the average motor speed of the throttle in this mode, with no altitude stabilisation. The UAV is stabilised in such a way as to self level when no input to Roll and Pitch is made. Throttle is adjusted to compensate for aircraft attitude actively, however there is still the possibility for loss of altitude for aggressive manoeuvres.

First the Yaw axis was tuned. To begin with the system was over damped, and did not reach the Yaw setpoint; this may be seen in Figure 5.7. An increase in the Yaw P gain was applied to first the outer loop, resulting in better setpoint attainment, then the inner loop was tuned, leading to effective rate control of the Yaw axis, as shown in Figure 5.8. Next the Roll and Pitch axis were tuned. This was done in a similar manner to the Yaw axis, with the UAV positioned in the centre of the space. It was then flown with a series of aggressive sinusoidal rate commands, first on one axis, and then the other. As the UAV is symmetrical it may be taken that the Pitch and Roll PID gains may be the same. This was confirmed by carrying out the same manoeuvres in

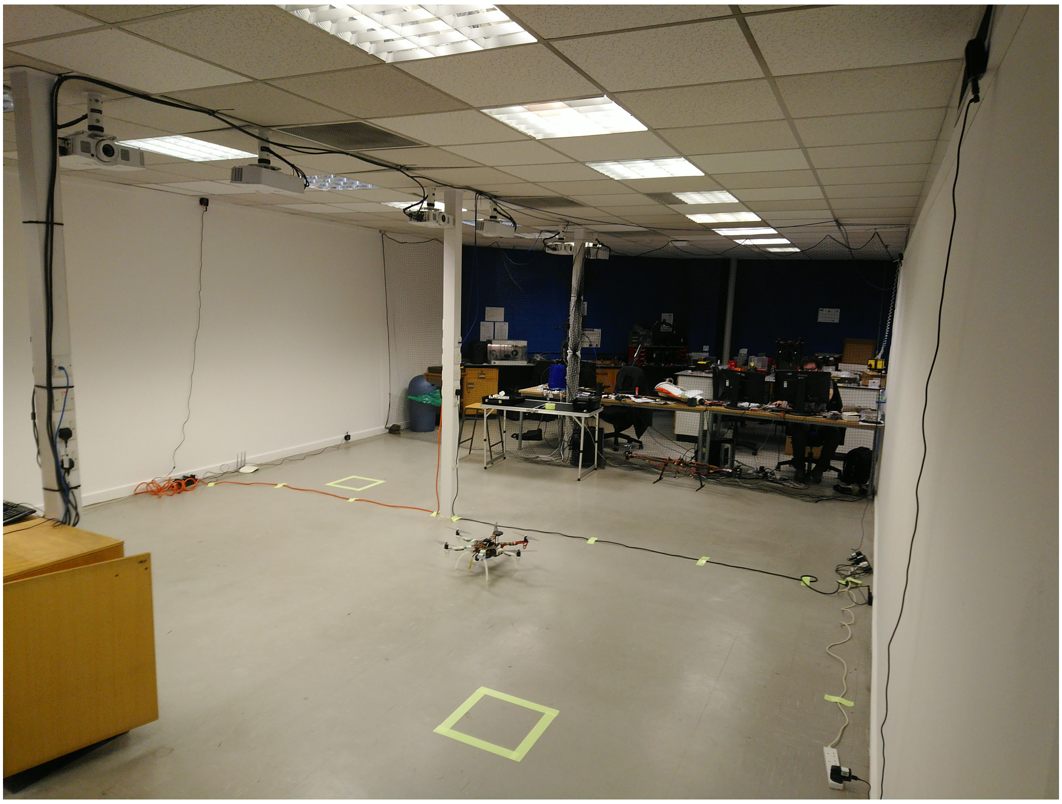


Figure 5.6: Space used for tuning of the UAV platform, netting seen in the background wraps around the flying space behind the camera position.

each axis direction. In this case the system was initially under damped, as seen in Figure 5.9. However, after tuning this was rectified as seen in Figure 5.10.

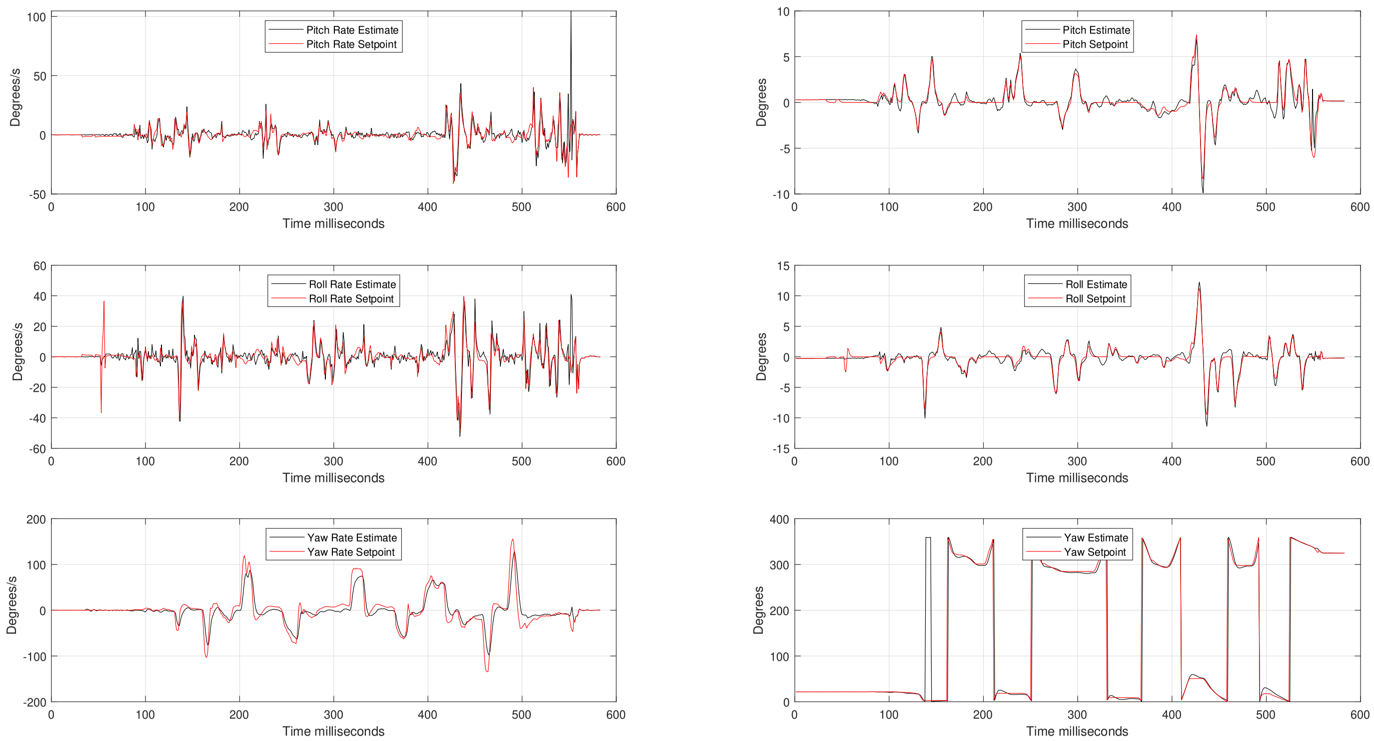


Figure 5.7: Prior to tuning Yaw. Set of graphs describing the recorded setpoint and estimate positions in Roll, Pitch, and Yaw during tuning. The left hand side describes the inner loop PID, with the outer loop PID presented on the right side.

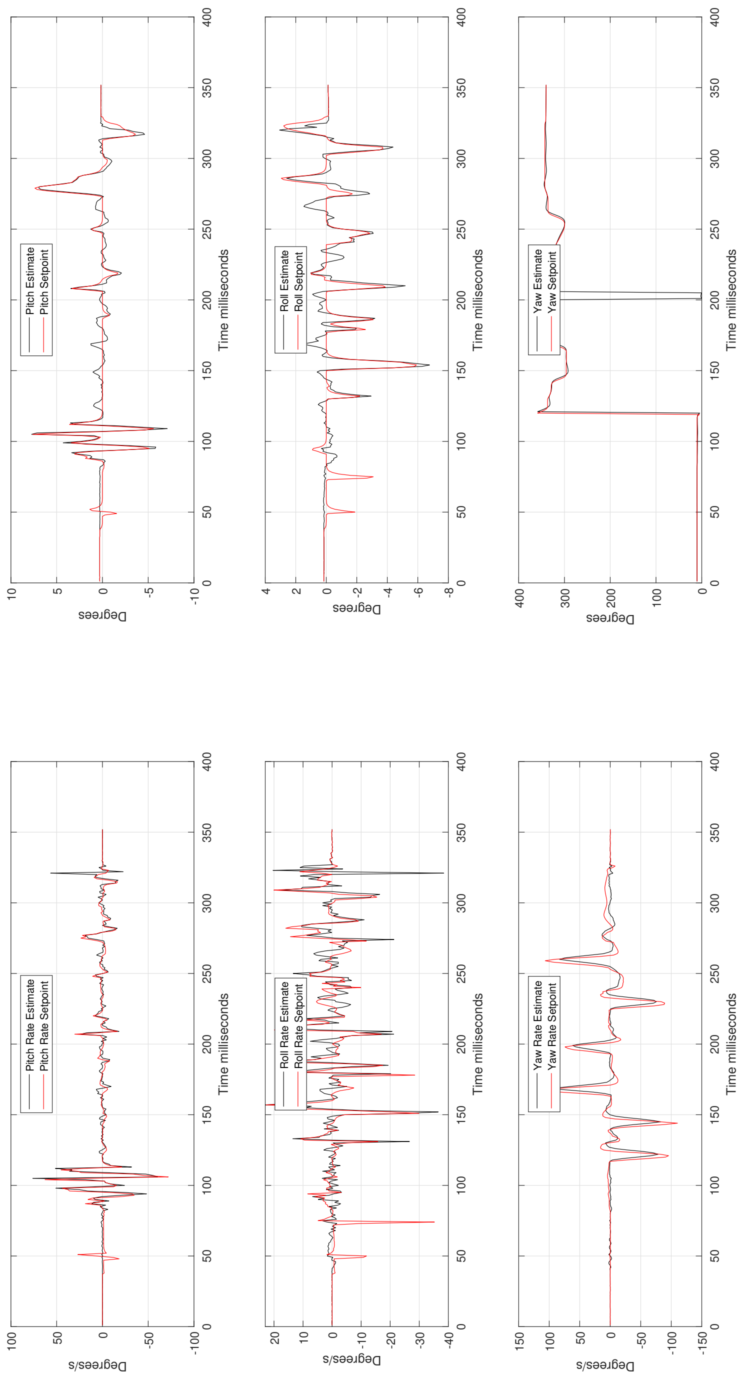


Figure 5.8: After tuning Yaw. Set of graphs describing the recorded setpoint and estimate positions in Roll, Pitch, and Yaw during tuning. The left hand side describes the inner loop PID, with the outer loop PID presented on the right side.

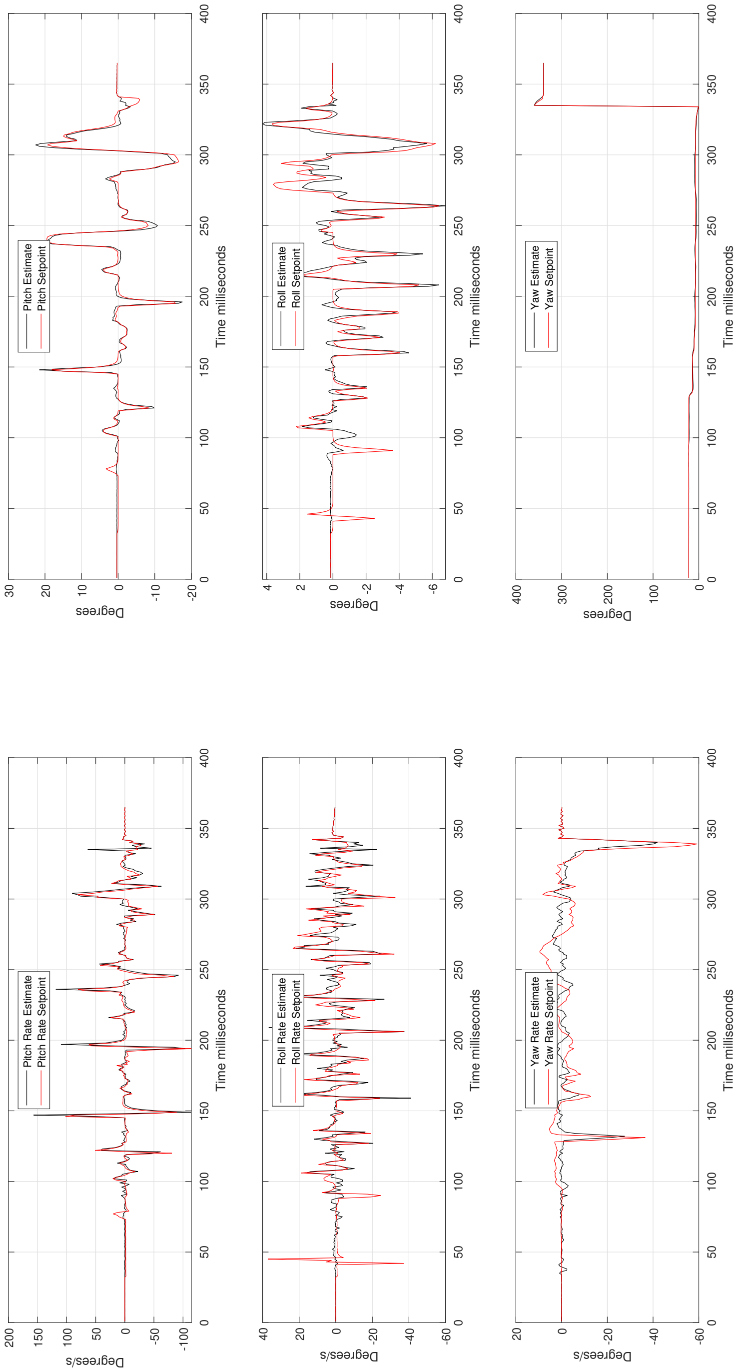


Figure 5.9: Prior to tuning Pitch and Roll. Set of graphs describing the recorded setpoint and estimate positions in Roll, Pitch, and Yaw during tuning. The left hand side describes the inner loop PID, with the outer loop PID presented on the right side.

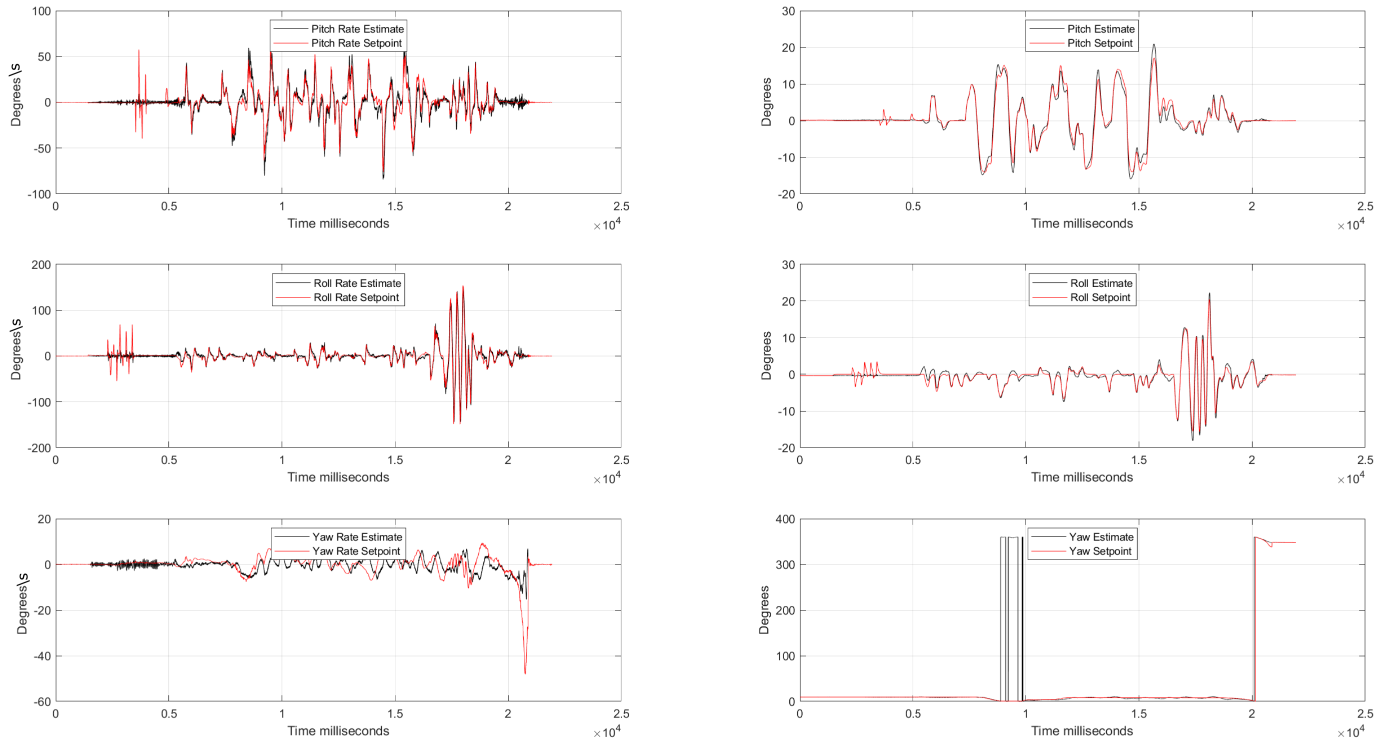


Figure 5.10: After to tuning Pitch and Roll. Set of graphs describing the recorded setpoint and estimate positions in Roll, Pitch, and Yaw during tuning. The left hand side describes the inner loop PID, with the outer loop PID presented on the right side.

5.4.4 Testing Space

The testing environment for this section of the project had a number of requirements that were not previously imposed upon the spaces used in earlier chapters. As a UAV would be flown as a part of the testing programme, the space needed to have sufficient z height to allow for safe operation and reasonable translation of the UAV . In this case, reasonable translation is taken on the same scale as that allowed in x and y. For this purpose the LJMU Lower James Parsons lecture theatre was chosen. As these tests were conducted during a break from teaching the space was available for long periods of time, allowing for the UWB anchors to be left in place for better comparison of their operation. The space has two entry points, as seen in the right of Figure 5.12 and off image to the left of Figure 5.13. Both of these entry points were easily controlled through the use of warning signs to bar entry except when authorised. The space also allowed for the piloting of the UAV from a position between the platform and an exit. This was desirable in case of an incident involving a LiPo battery fire. The space also allowed for the placement of the UWB anchors in significantly different z altitudes to one another. This is required as the 3D multilateration of an unknown point benefits from large differences in all three dimensions. The UWB anchors were powered from wall outlets, through the use of identical 240V to 5v 2A USB adapters.

The RTS was set up in a position that allowed for unobscured sight of the intended ground and aerial spaces, and mounted upon a surveying tripod. This was placed on the ground over a more supported section of the flooring. This was decided as during setup the operator's weight, or heavy equipment was found to be enough to alter the level of the RTS out of range after initial setup, when placed up to 2m away. In the new position, as seen in Figure 5.14, this effect was found to be negated.

The take-off area for the platform was designated in front of the platform and desk seen in Figure 5.11. This position was chosen as it allowed the most space for quick landing if that was required. Other constraints upon flight in this space were the low hanging projector towards the back, also seen in Figures 5.12 and 5.13.

Figure 5.11: View of the take-off area and test space.

Six UWB anchors were positioned around the perimeter of the space mounted on tripods, as seen in Figures 5.14 and 5.15. The positions of these anchors may be seen in Figure 5.16. As before, the RTS was used to establish the global reference frame and measure the anchor positions. These measurements were then used to set up the associated ROS node. As mentioned previously, the time given by the laptop clock is different from that of the time on the onboard computer. To avoid misalignment of the UWB and RTS measurements, the RTS measurement is republished by a node running on the onboard computer, which replaces the laptop time with that of the time registered on the Pi. This means that all measurements are published with regard to the same clock.

Figure 5.12: View of the lecture theatre testing space - Back

Figure 5.13: View of the lecture theatre testing space - Front

Figure 5.14: View of the UWB anchors setup - right side of the room

Figure 5.15: View of the UWB anchors setup - Left side of the room

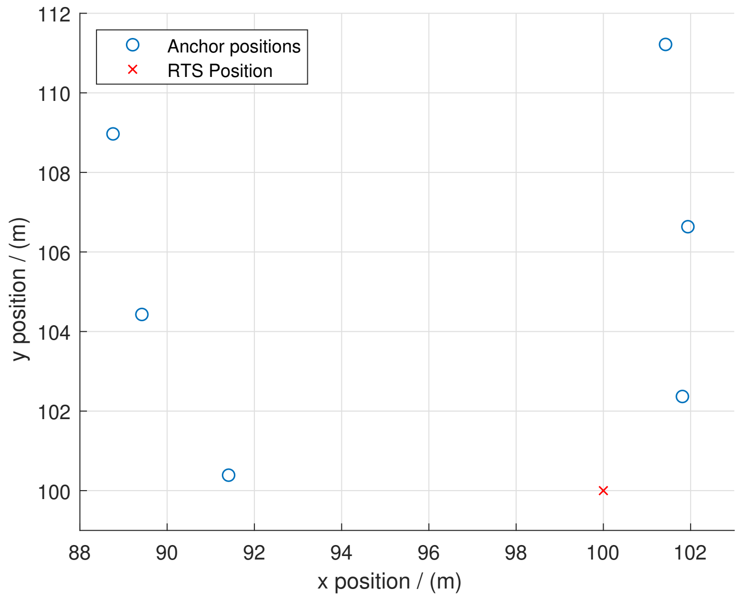


Figure 5.16: Plot of UWB anchor positions

5.4.5 Testing Procedure

The UAV was piloted from a position to the left of the take-off area, and armed when the initial setup of the onboard sensors was complete. This was done via a laptop connected to the onboard Pi computer via secure shell (SSH). This laptop was also used in previous experiments as the connection point for the RTS to the ROS network.

The UAV was flown in a variety of manoeuvres, and the data from each manoeuvre was logged through the use of the ROSBag function. The principle aim of the manoeuvres was to cover a large area of the space, with as much variety as possible in terms of ranges to anchors and changes in altitude. These manoeuvres took the form variously of circles, lines, and s shapes.

For the UGV portion of the testing, the area available was somewhat limited in comparison to previous tests, however still allowed for translations along the x axis of 8 m. The UGV was also tracked with the RTS to provide ground truth measurements. The Leica Mini prism was used for this. As the testing area is limited in this case, the trajectories available were also limited. The UGV was piloted in front of the main podium from the entrance archway, as seen on the left of Figure 5.12, and across towards the RTS.

5.5 Results and Analysis

This section will deal with the results and analysis of the experiments described in the previous section. First the results of the initial system testing and platform experiments will be presented, with the results of the indoor testing in the James Parsons lecture theatre presented last.

5.5.1 Initial System Testing

As the process of UAV flight in confined spaces with the required equipment is somewhat complex, initial testing was carried out to test the system architecture and reliability for later tests. During this initial testing of the UAV system, a number of challenges were encountered and noted for further development. Among these were connection issues to the wifi access point used for system networking. The onboard Pi often dropped its connection and attempted to connect to stronger networks in the area. This was solved by disabling the graphic user interface of the Ubuntu Mate OS.

Another issue that proved to be a much greater challenge was that of setup errors. The initial testing did not utilise an RTS system for ground truth and UWB anchor measurement, as the objective was to determine system faults, and not data quality. To this end, the anchor positions were measured using a tape measure. However, an error was made during the input of this data into the UWB ROS node. Instead of an anchor height of 2.4 m, a height of 24 m was entered. This was easily done as the measurements are entered in mm, and the file display format in use does not allow for easy identification of errors such as this. This is due to small screen display size and an awkward interface. To this end, a step was added to the setup of double checking the UWB anchor positions for this and other input errors. Due to this error, the UWB data was significantly noisy. This may be seen during a test flight during which the UAV was piloted to hover. The UWB data may be seen in Figure 5.17. Although a ground truth is not available, the position of the UAV was maintained within an area of roughly 1 m^2. An image of the flight may be seen in Figure 5.18.

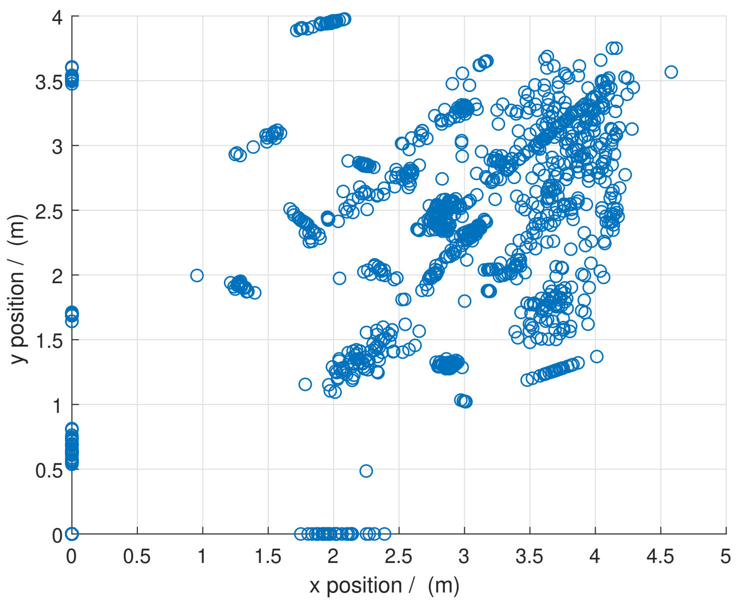


Figure 5.17: Plot showing UWB output during initial UAV testing in a hover.

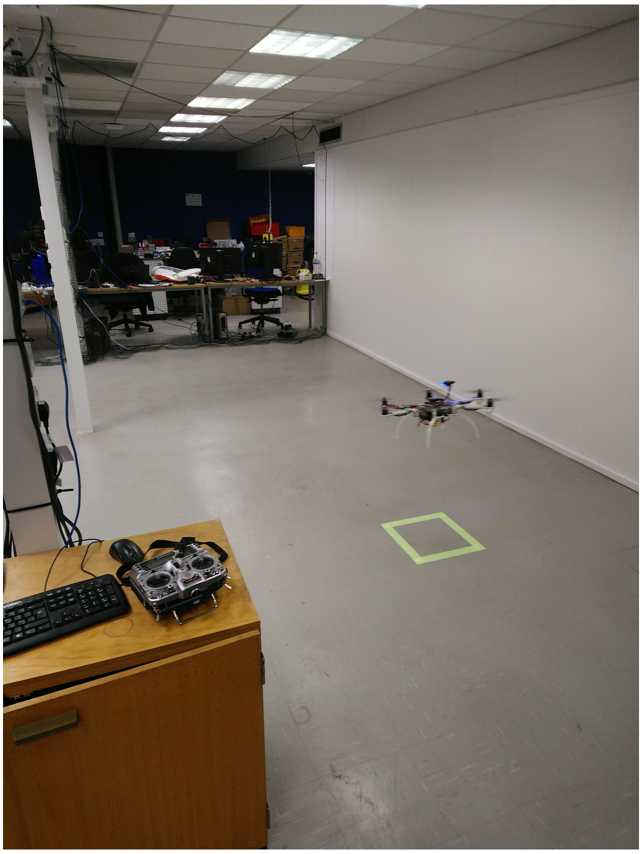


Figure 5.18: View of the initial UAV testing.

5.5.1.1 Effects of UAV Motor Operation on UWB Range Measurement

As has been noted in other works, the RF environment the UWB operates within is generally not a factor which has an effect upon the measurement reliability. However, as the UAV multirotor utilises BLDC motors for propulsion it is expected that an electromagnetic field will be induced. This is unlikely to have an impact the function of the UWB system, however it was prudent to demonstrate this is not the case, especially due to the short amount of time required to conduct such a study. Therefore, an experiment was devised to determine the extent of this effect.

The UWB system was set up as a single anchor rover node pair in a room similar to those used in previous and future tests. The UAV was placed upon a small stand 1 m above ground. The rover node was mounted downwards facing on the bottom surface of the UAV as intended. The UAV system was powered using a 4 cell Lipo battery. Range measurements were collected between the node pair using the UWB system for two minutes. The motors were then powered without propellers, and set to idle, so as not to change the position of the UAV . The UWB range measurements were then recorded for two minutes. A time period of two minutes was chosen as although there is no time limit for data collection while the drone motors are powered down, running the UAV motors stationary is not recommended. The BLDC motors used in multirotor propulsion depend upon the forced airflow cooling from prop wash. To determine whether placement of the UWB node would alter the effects, if any were noted, a second node was mounted 10cm above the frame of the system. This position was chosen as it is similar to the GNSS antenna mounting position. The second rover node also gathered range measurements from the anchor node, and was networked in the same manner as the first using ROS.

An issue found during testing showed that the dropout rate of the UWB range measurements significantly increased while both UWB nodes ran simultaneously. It is thought that the communication between one rover and the anchor disrupted the communication between the second rover and that anchor. The dropout rate of the measurements for both nodes was on average 50%. With successful measurements alternating between active rover nodes. This is a large increase from the expected rate of between 98.5% and 99.9% successful range measurements seen in other tests. The initial results of the tests with dropouts removed are shown in Figures 5.19 to 5.22. An unusual point to be seen in this data is that although the standard deviations of the data are roughly the same, there are a number of outliers that skew the results - hence the large x-axis for many of the histograms. In the case of this data there are no more than 10 outliers per experimental repetition. The data may be seen without these outliers in Figures 5.19 to 5.22.

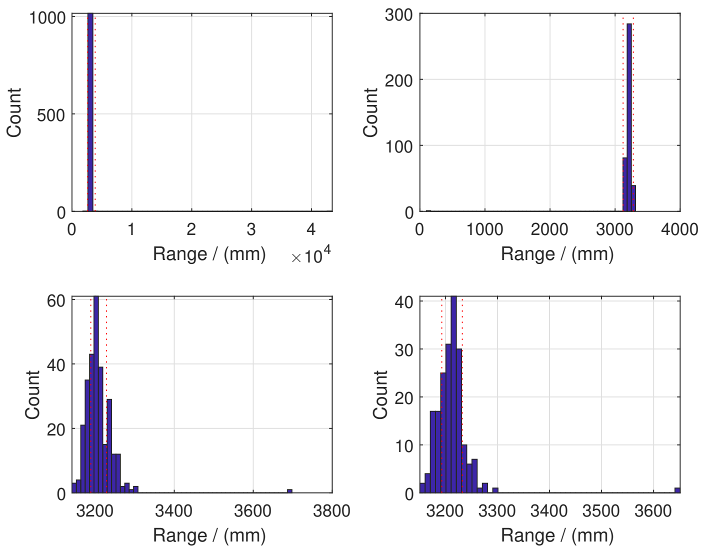


Figure 5.19: Histograms of UWB range measurements from node 1 with motors off. Each plot is one repetition of the experiment.

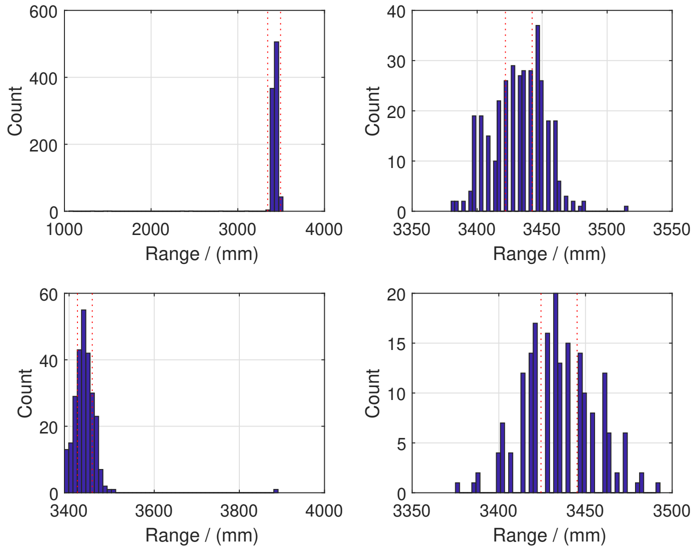


Figure 5.20: Histogram of UWB range measurements from node 2 with motors off. Each plot is one repetition of the experiment.

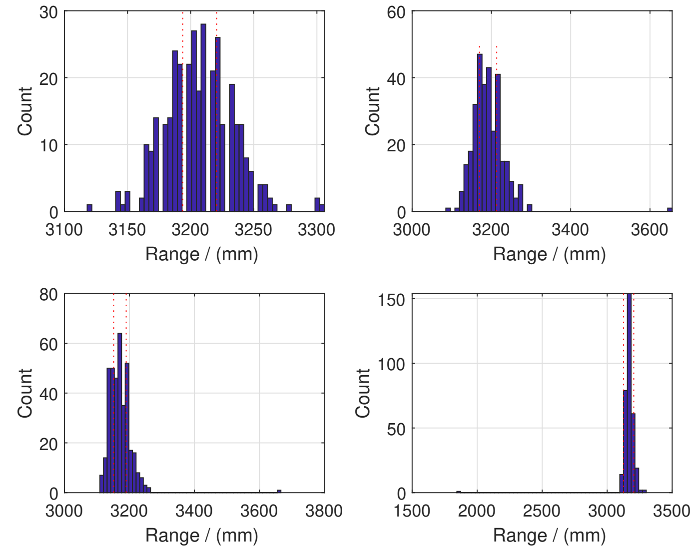


Figure 5.21: Histogram of UWB range measurements from node 1 with motors on. Each plot is one repetition of the experiment.

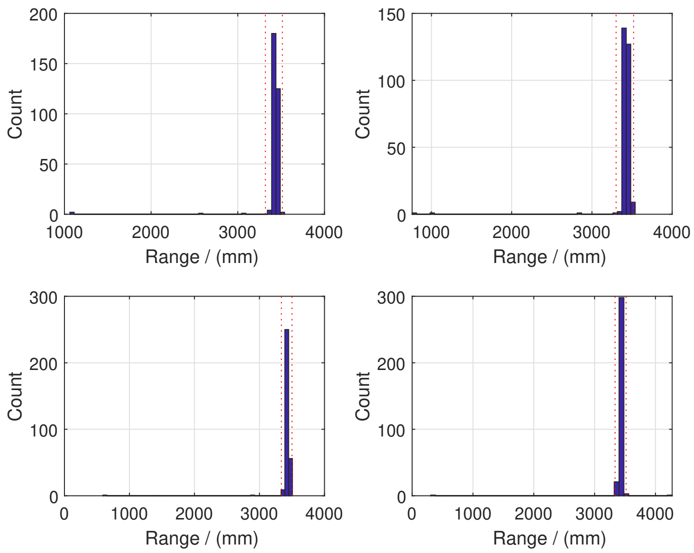


Figure 5.22: Histogram of UWB range measurements from node 2 with motors on. Each plot is one repetition of the experiment.

5.5.2 James Parsons Lecture Theatre Test

During the data collection described, a number of issues were encountered. Firstly, the BNO055 was found to drop its connection to the Pi, rendering data collection of certain trajectories incomplete; this was found to be due to a faulty jumper lead. Secondly, and most importantly, the UWB measurements were significantly more noisy when viewed as the x, y, z output from the Pozyx. This may be seen in Figure 5.23. An initial check of the setup showed no errors had been made, and all of the Pozyx anchors were functioning as expected, aside from decreased accuracy. The BNO055 was stable throughout testing. Overall, 11 complete bag files were captured over 14 flights. As the anchors on the 90 m x axis area are placed against a wall it is clear that the measurements made with an x reading of less than 90 m are multipath errors caused due to reflection of the signals from the left hand anchors.

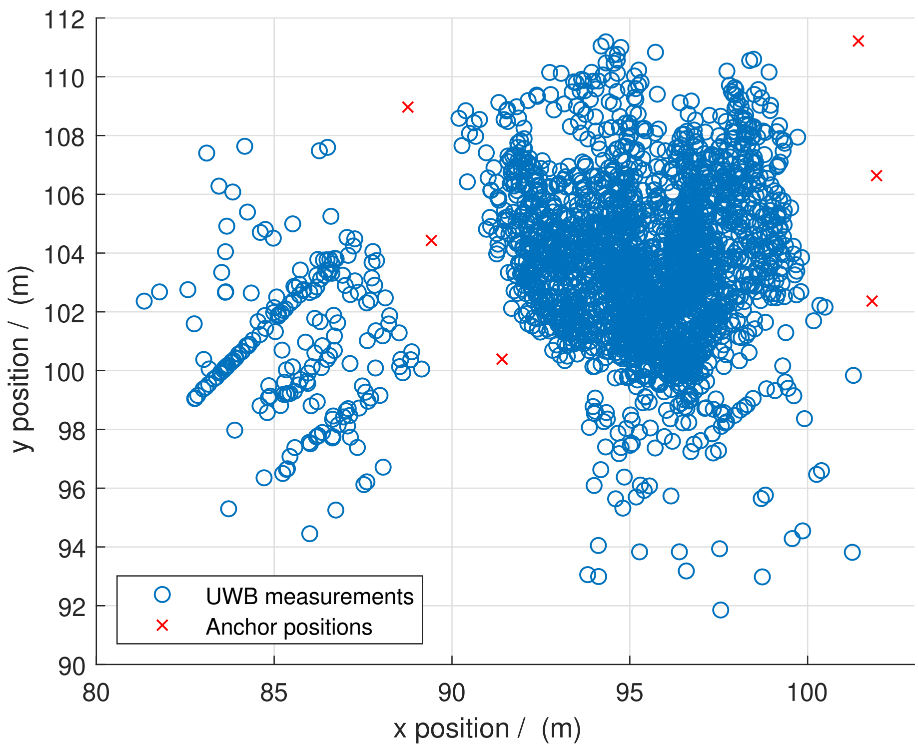


Figure 5.23: 2D plot of x, y measurements made by the UWB system with anchor positions marked.

To further investigate the cause of this noise, a closer inspection of the UWB data is needed. When viewing the x, y, z data there is a degree of post processing which may obscure the characteristics of the data. The range measurements were compared to the ground truth ranges which may be calculated given the known position of the UWB anchor nodes and the rover node at any given time. From this point it is a matter of calculating the euclidean distance between the rover node and the anchor node. This may be done in MATLab using the "pdist" function. The main challenge in this is the synchronisation of the RTS data to the UWB data and the question of the different data rates. As the RTS makes measurements at a rate of 1Hz, and the UWB system at a rate of 50Hz linear interpolation is used to align the data. Once this is done a direct comparison may be made, as seen in Figure 5.24. From this comparison it is clear that there is a large proportion of dropouts in the UWB data. It is also clear that there is an offset at the beginning of the recording. Throughout the test $16,296$ range measurements were gathered from all of the UWB anchors, and with 1755 of these being dropouts, this gives the dropout rate of 89.21%, meaning 10.79% of the measurements failed. This dropout rate is far above the usual range of 99.5%to97.5

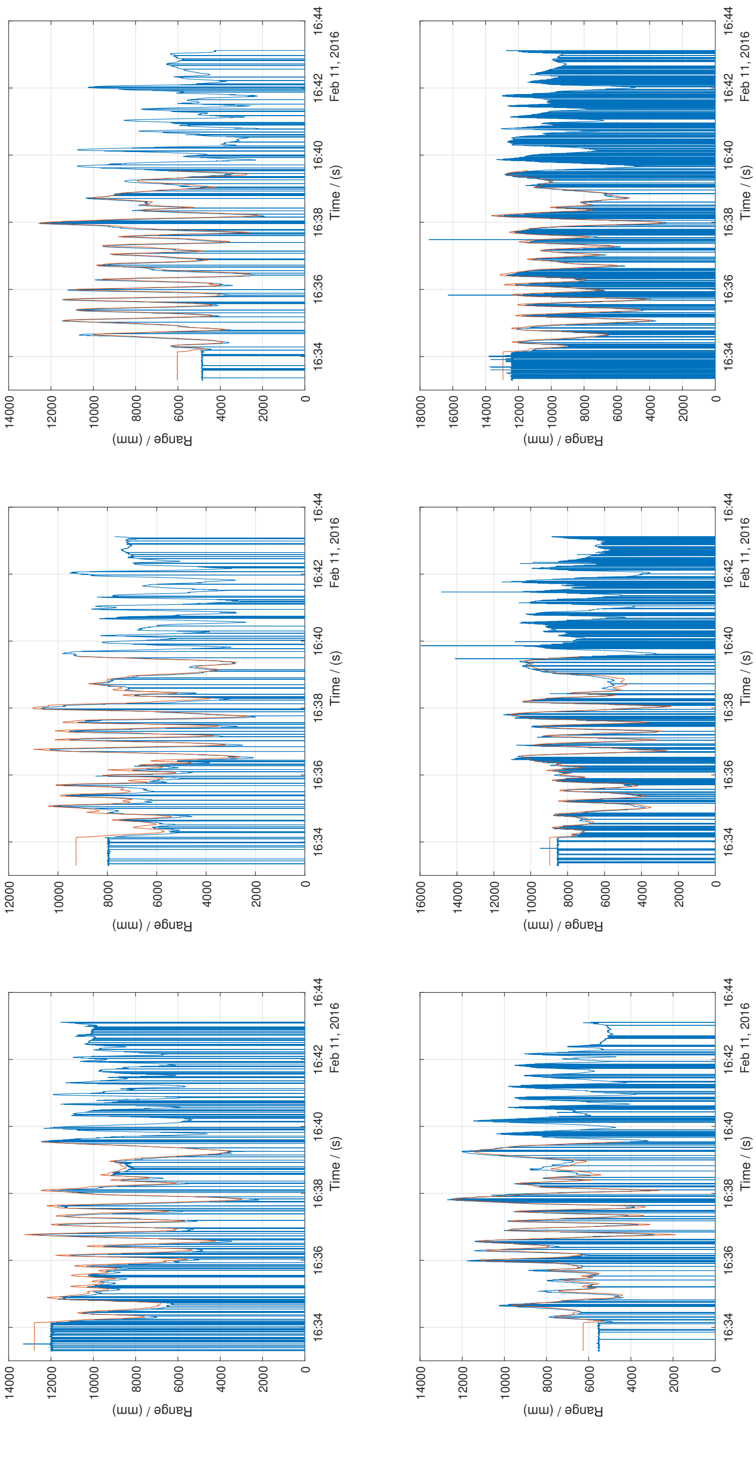


Figure 5.24: Comparison of UWB range measurements (blue) to ground truth measurements calculated from the RTS (orange). Anchors 1 to 6 clockwise from the top left.

Further examination of the UWB data demonstrates that the range measurements from individual anchors also presented with an increased degree of noise. This may be seen in Figure 5.25, where the histograms of the anchor-rover node range error measurements are presented. The multipath errors are evident in areas such as in anchor 6 of Figure 5.25. In this case there is a large spike between 10m and 15m, demonstrating a large number of erroneous readings. The range error is calculated by subtracting the calculated ground truth range seen in the previous figure, and the UWB ranges are then subtracted from the ground truth measurements; this may be seen in Figure 5.26. The increase in range error towards the end of the figure is due the loss of RTS track.

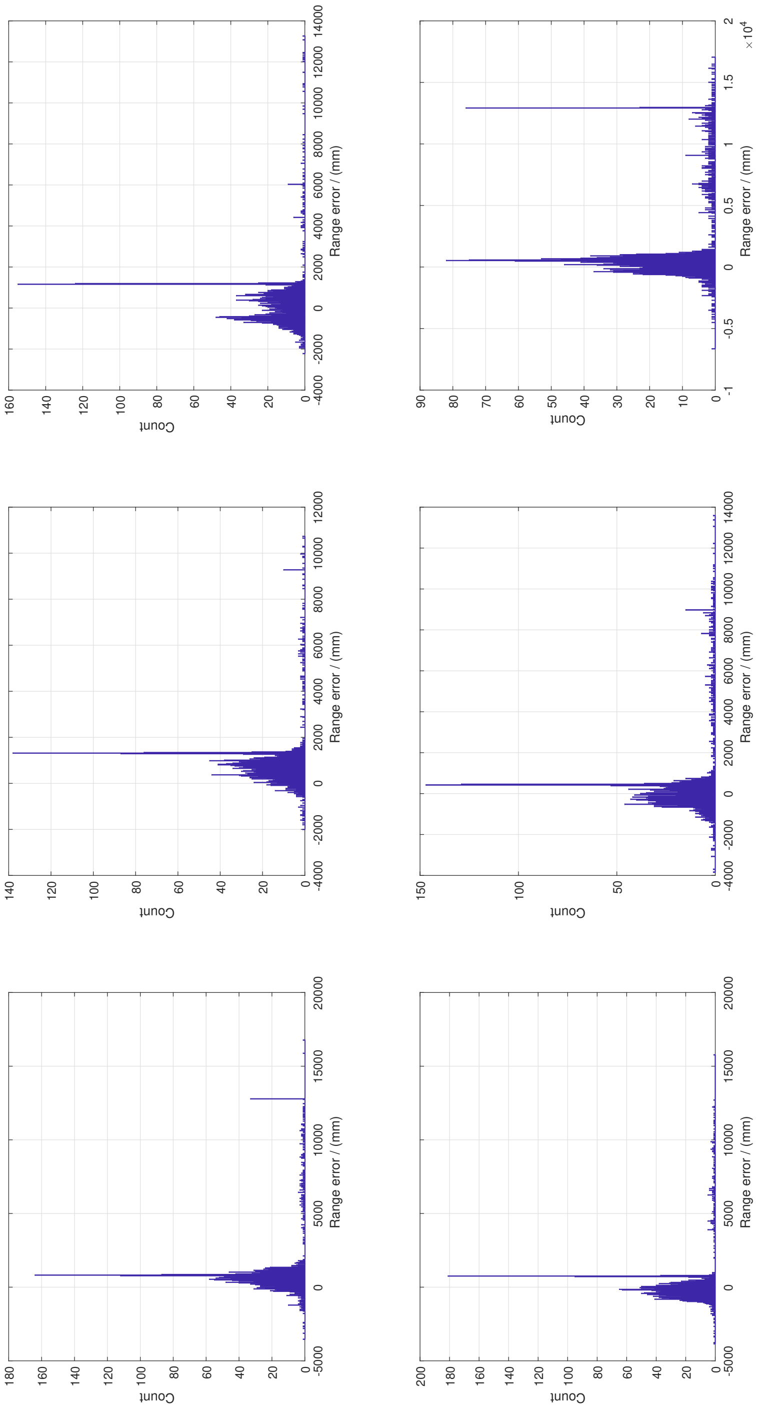


Figure 5.25: Range error histogram of the UWB measurements calculated from RTS ground truth. Anchors 1 to 6 clockwise from the top left.

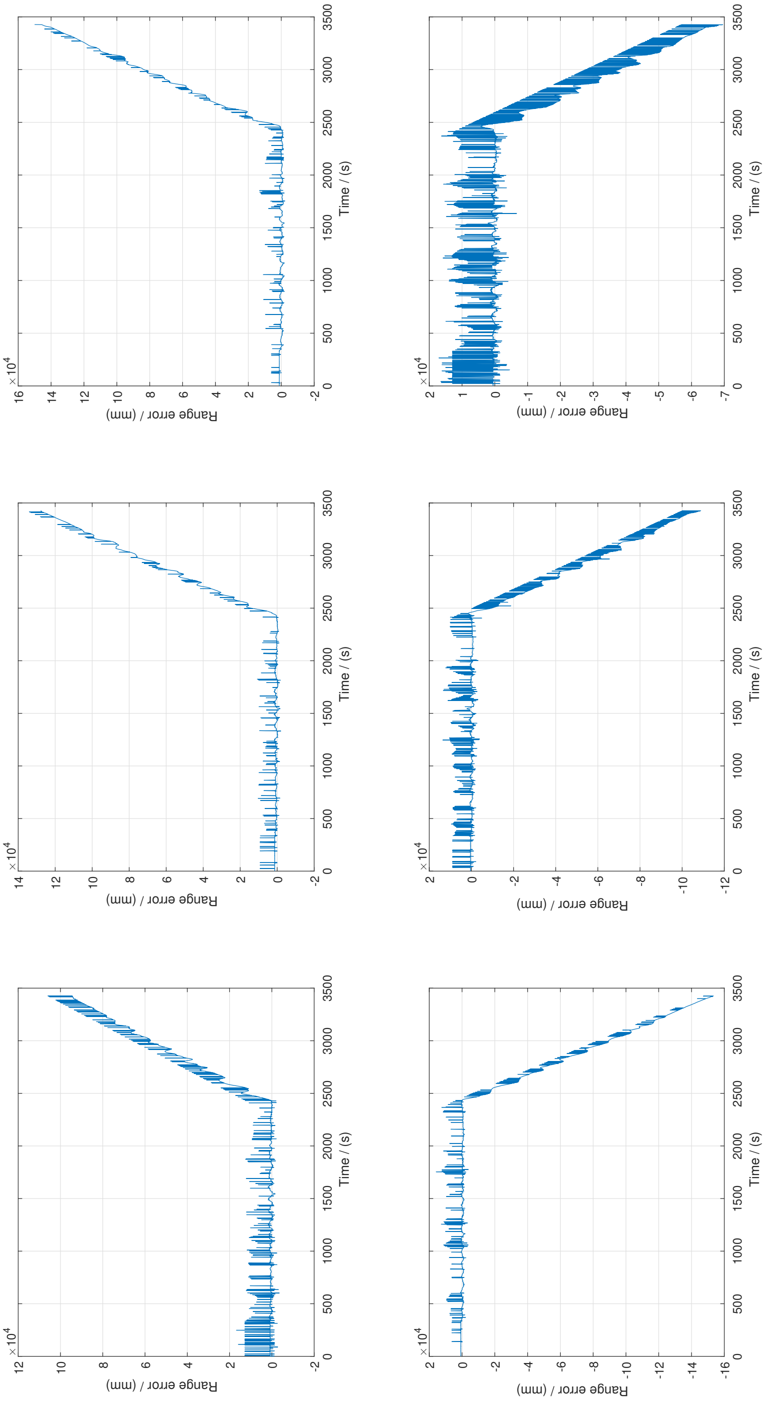


Figure 5.26: Range error of the UWB measurements calculated from RTS ground truth. Anchors 1 to 6 clockwise from the top left.

To determine whether this increased noise was due to the operation of equipment beyond the purview of the experiment, it was decided to repeat the test on a second day. This second series of tests produced 15 complete bag files, however although there was some improvement in the quality of the x, y, z track, this data was still significantly less accurate than in previous tests. This may be seen in Figure 5.27, where although there is a no obvious multipath error as seen in the previous testing, the UWB measurements were found not to be reliable.

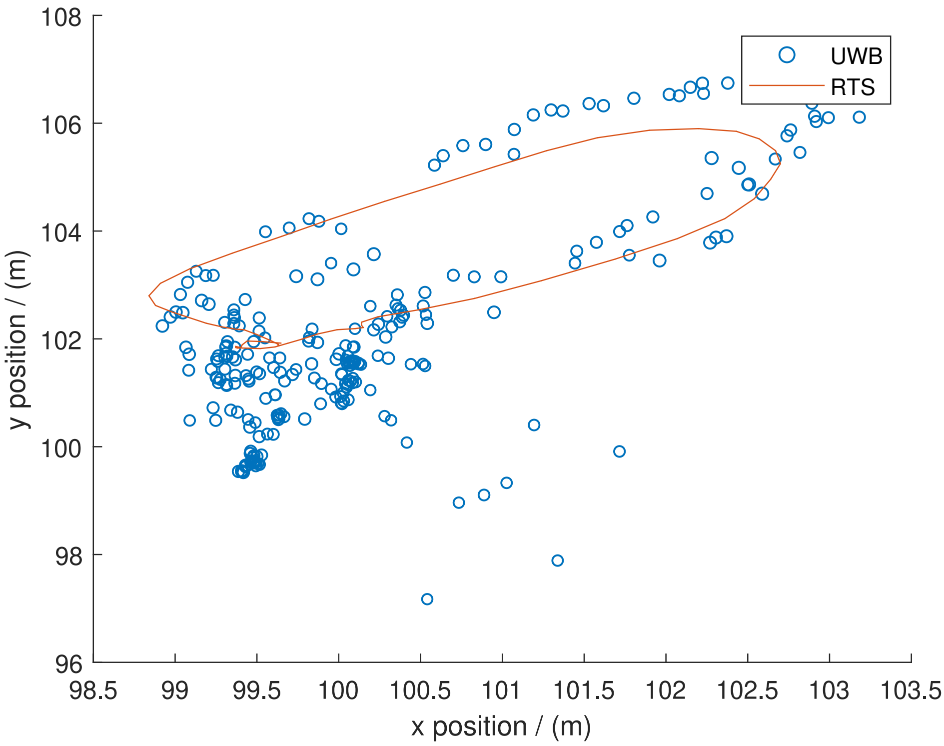


Figure 5.27: UWB and RTS results from the second round of testing in the James Parsons lower lecture theatre.

5.5.3 6DOF EKF Results

In the previous section, the results from the testing in the James Parsons lecture theatre were presented. In this section it is shown that the results were less than ideal, and demonstrated a significant degree of noise in comparison to the previous experiments. This is expected to drastically limit and decrease the degree to which the developed 6-DOF EKF will be able to track the position of the agent. With this in mind, the data collected was used as an input to the 6-DOF EKF . The results of three trajectories may be seen in Figures 5.28 to 5.30. Although the EKF is functioning in 3D space the results presented here are done using a 2D plot. This is less for analysis of the effectiveness of the algorithm, and more a demonstration of how the increased noise data affects the output. For clarity, a 3D plot of trajectory 1 may be seen in Figure 5.31. It may be seen from the EKF estimated position output that in certain circumstances the EKF is smoothing the UWB output data to provide an improved trajectory estimate. However, with increased UWB noise and outliers, the EKF generated output diverges from that reported by the RTS. It may be seen from Figure 5.31 that the EKF trajectory in z is more stable, although the trajectory overall is neither smooth nor stable.

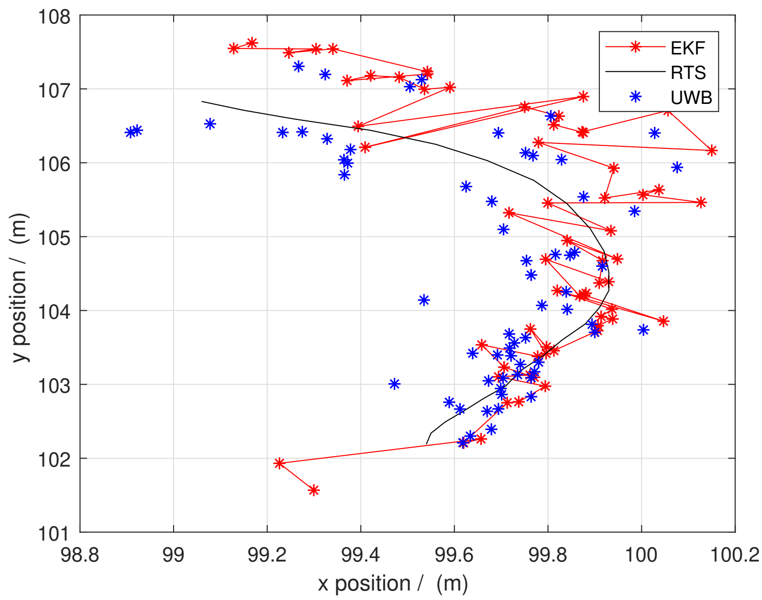


Figure 5.28: Plot showing RTS ground truth of trajectory 1 flown, UWB measurements and the EKF estimate in 2D

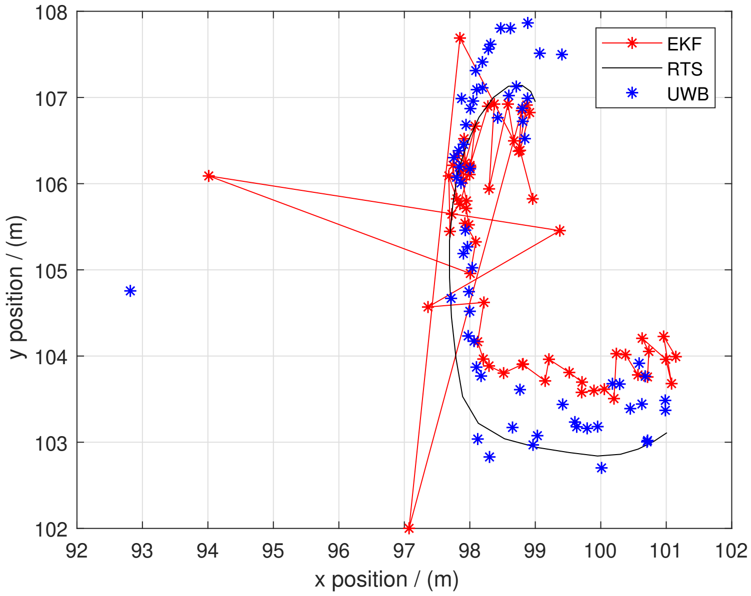


Figure 5.29: Plot showing RTS ground truth of trajectory 2 flown, UWB measurements and the EKF estimate in 2D

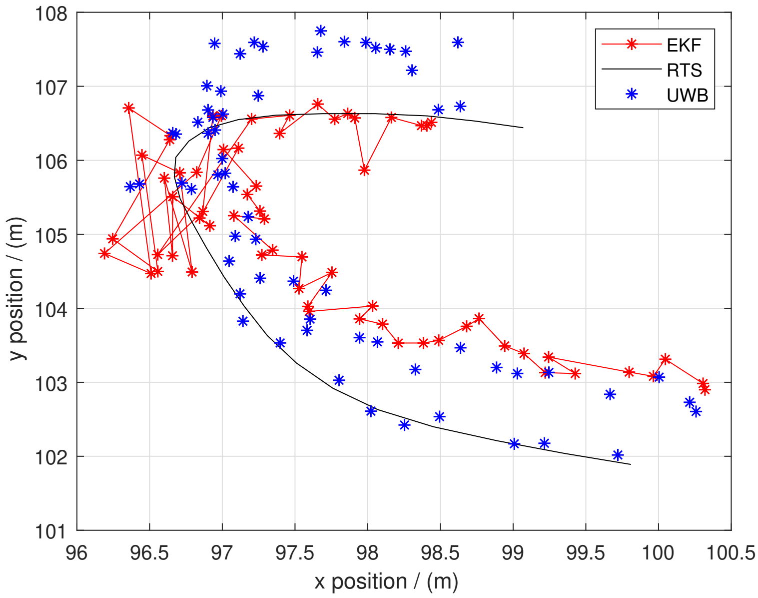


Figure 5.30: Plot showing RTS ground truth of trajectory 3 flown, UWB measurements and the EKF estimate in 2D

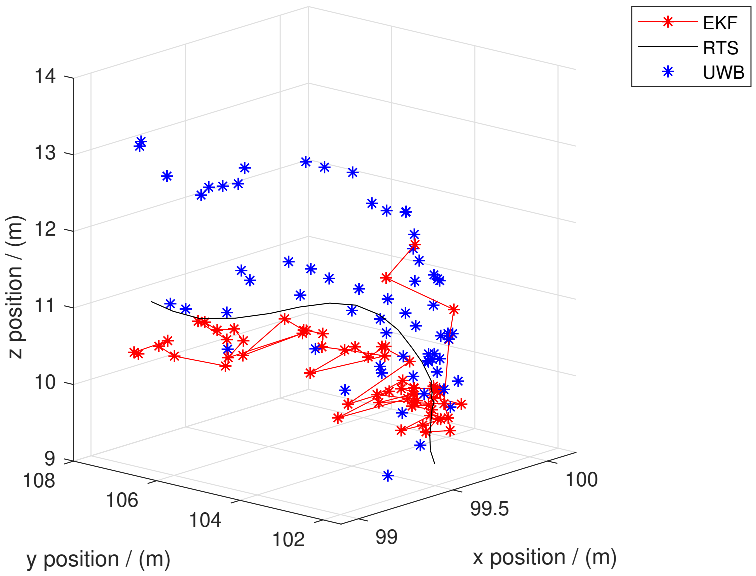


Figure 5.31: Plot showing RTS ground truth of trajectory 1 flown, UWB measurements and the EKF estimate in 3D

In the previous chapter we presented a comparison of the conventional EKF formulated in Chapter 3, and the same algorithm with the addition of a more full representation of the sensor measurement variance. In this case, the modified EKF was proved to be more resilient to increased measurement input noise. However, as seen here, the EKF does not function well with noisy data. Two points may be stated here; firstly, the measurement noise in this case is greater than seen in the previous chapter and secondly, the new EKF does not utilise wheel encoders as a control input. This is an important change in the algorithm. Wheel encoders directly measure the displacement of the platform, and when stationary do not drift. When used in conjunction with a sensor such as the UWB system this is ideal, as the UWB measurement may be considered reliable in terms of the x, y, z state components. However, in the case of the 6-DOF EKF , the only sensor capable of making a direct observation of the platforms x, y, z states is the UWB. Where the UWB observation is compromised as in the previous chapter, the system is still resilient, however in this case the system is not capable of drawing observations from the control inputs.

5.6 Conclusions

This chapter has demonstrated the continued applicability of the RTS system as a ground truth method for UAV platforms out to a range of 400m. This was confirmed through testing in an outdoor environment with several manoeuvres which may be expected in the use of a UAV . The effects of a UAV 's motors on the Pozyx have been tested to determine whether this needs to be a factor in the modelling of Pozyx measurement variance. However, this factor has been shown to cause no observable change in the operation of the Pozyx system measurements under these circumstances. This has been followed with the construction of a UAV platform that may be used in the same fashion as developed for the ground vehicles, within the existing infrastructure for data communication, synchronisation and storage.

The lessons of previous chapters have also been combined to construct an EKF that is applicable for a UAV , allowing for 6-DOF operation. The measurement update function was also modified to allow for updates of the staes in x, y and also z. This was done using the same workflow as previously demonstrated. A suitable testing environment was also selected. However, upon testing in the observation, inputs from the Pozyx proved to be too noisy for reliable estimation of the UAV position. This is an unexpected result as although in past experiments the Pozyx was noisy in indoor spaces, this was not thought to lead to issues, as demonstrated by the work published from the previous chapter [129]. However, in this case the UWB noise was found to be significantly higher than previously found this was confirmed upon further examination of the UWB range measurements of the UWB data.

The next chapter will investigate the increased noise, and endeavour to determine the cause or causes. The results from this section of experiments suggest the material used in the space increases the likelihood of signal reflections for the UWB system as discussed, leading to multipath errors. There is also an increased rate of dropouts in the range measurements, which suggests other factors may also contribute to this. The results of this investigation will be applied to a second experiment which will reassess the EKF formulated in this chapter.

Chapter 6

6 Degrees of Freedom for Ground and Aerial Platforms

In the previous chapter, a 6-DOF EKF was developed. The observation model for the UWB system in 3D was implemented and integrated alongside the system model presented in Chapter 2. The method used in previous chapters to collect the ground truth for the platform was also tested in 3D through the use of a multirotor platform out to 400m. This investigation demonstrated the manner in which the UAV platform must be flown to allow for robust tracking using the RTS, and proved that the system is capable of providing 3D tracking in real time. The intended testing platform was also constructed, and tested according to University and legal regulations.

Testing of this EKF formulation was conducted in an indoor area, using six UWB anchors placed about the perimeter of the space. This series of experiments led to an inconclusive result due to unexpected noise, and dropouts in the UWB system. In Chapter 4 the UWB system was set up in such a way as to decrease the system accuracy to test how well the EKF functioned under these conditions. However, the noise levels shown in the previous chapter are far in excess of that encountered previously. Due to this increased noise and dropout rate the formulated EKF was unable to function to an acceptable degree. The experiments conducted in this chapter were not included earlier in this project due to the scope defined. This research was not aimed at constructing an improved understanding of the performance of UWB systems. A better understanding may be gained during the fulfilment of its aims, however the scope has been to investigate the effects of including a more detailed representation of a sensors measurement variance in the state estimation process. To do this, as stated in Chapter 4, a single factor was identified. This procedure, as demonstrated thus far in this research is aimed at constructing a transferable method which may be applied to other observation inputs, not just UWB systems.

This chapter will focus on determining the causes of this increased noise and dropout rate, the resolution of these effects, if practical, and the re-testing of the algorithm. Lessons learned in the investigation will also be applied to strengthen the algorithm to reduce its susceptibility to such effects in the future.

6.1 UWB Noise Source Identification

During the course of the project the Pozyx system has presented an increased level of measurement noise for the x, y, z measurement in certain circumstances. This section aims to investigate the causes for this increased level of outliers and unreliability.

It was determined from the previous investigations, as presented in Chapter 4, that there are a finite number of factors which may individually or collectively be the cause of this increase in noise level.

Firstly, the manner in which the UWB anchors are powered may cause a change in reliability. As the testing and negation of these effects would not be labour intensive, this was tested first. In previous tests the UWB anchors have been powered through the use of AC - DC wall plugs, which convert the 220VAC mains supply to 5VDC. In the case of the initial testing, the anchors were powered from wall outlets from terraced houses on the University grounds. In the later tests the anchors were again powered from wall outlets, however this time in the main campus. This presents a change in the circumstances which may be a factor in the change in UWB measurement noise observed. The UWB anchors are connected to these supplies though the use of USB cables. In this investigation the UWB anchors were first powered using these wall adapters, then using LiPo batteries and DC - DC converters. The dropout rates and standard deviation will then be compared.

Secondly, the environment in terms of materials present in the testing space may be the cause in the increased noise. In the early work, where the Pozyx was shown to be most reliable, the environment was outdoors, with one large metal clad building on one side and conventional terraced houses on the other. The rest of the space was not enclosed. In later work the UWB system was tested in indoor spaces; this is likely to increase the probability of multipath errors, leading to outliers. There is also the factor of the space in terms of the electromagnetic environment which may be a cause. In the outdoor space the University WiFi was accessible, however access points were not located in the testing space. In the later experiments the testing space was located in main teaching spaces where access points were mounted. The University system operates on both the 2.4GHz and 5.8GHz bands, while the 2.4GHz is outside of the UWB operating band the 5.8Hz is within the 3.5GHz to 6.8GHz used. This may be an issue along with other unknown EM sources.

Therefore, having identified some key possible factors based upon the literature and experimental observation, this chapter will initially focus on investigating the degree to which these variables will modify the results.

6.1.1 Power source

To determine the effect of the method used to power the UWB nodes a small network was set up in an indoor space. Four nodes were set up in a small testing space as shown in Figure 6.1. The same configuration was used as in the previous chapter, with the onboard Raspberry Pi computer mounted on the UAV platform, connected to the UWB rover node via USB. This connection was also used to power to rover node.

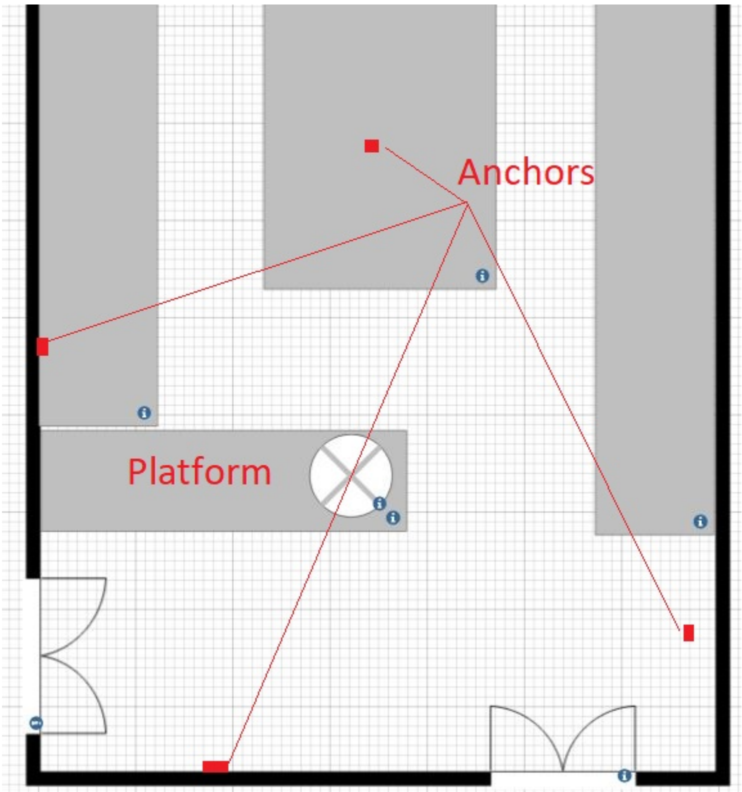


Figure 6.1: Layout of space used.

To begin with, the anchor nodes were powered from the wall outlets along with the Raspberry pi onboard the test platform drone. The drone was left in a stationary position with the anchor nodes facing inwards, towards the UAV and the UWB data was recorded for 20 minutes. This process was the repeated with the anchor and rover nodes powered in a variety of configurations as seen in Table 6.1. In all cases, the node placement was the same, and the equipment used to power each node for each configuration was constant.

Test number	**Anchor node power**	**Rover node power**
1	Wall	Wall
2	Battery	Battery
3	Wall	Battery
4	Battery	Wall

Table 6.1: Table describing the power source for anchor and rover nodes for each test.

It was also decided that investigating the effect of loads on the wall outlets during UWB operation would be prudent. To achieve this a high-power (1000W) battery charger was plugged into the socket next to one node, on the same circuit as the rest of the nodes. The battery charger was then set to charge a 4s LiPo batter at 6 amps, resulting in a maximum power draw W_{max} of 100.8W.

6.1.1.1 Results and Analysis

The results from the initial testing of a stationary rover node with the anchors powered in configurations 1 and 4 may be seen below in Figure 6.2. Shown here is the UWB pose output in x, y and z. Both tests were run for similar lengths, however to allow for more direct comparison extra samples from the longer tests were not plotted. Therefore all histograms shown in Figure 6.2 contain 5663 samples.

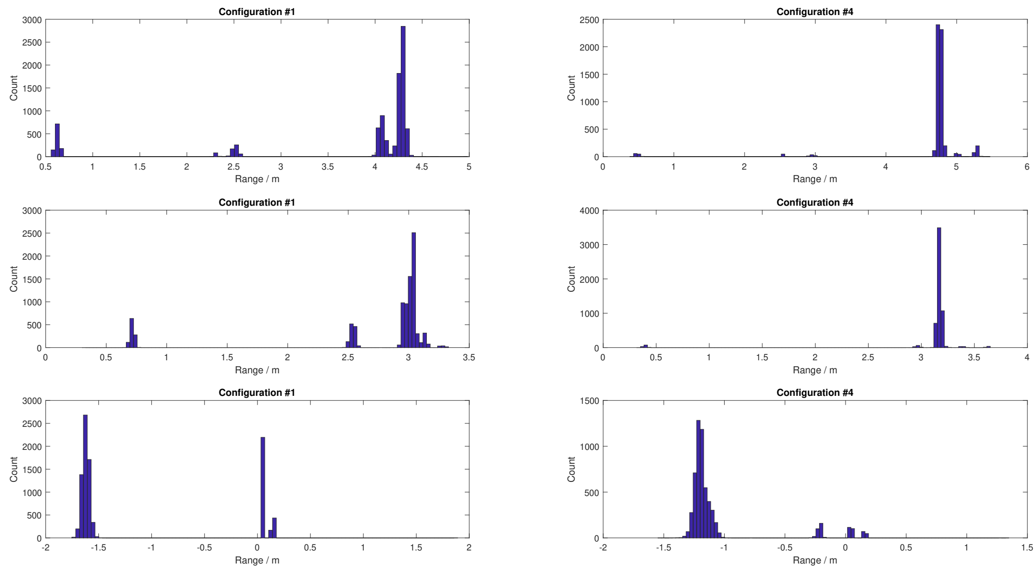


Figure 6.2: Histogram of x, y and z readings from UWB in two power configurations.

As has been mentioned previously, the x, y and z output from the UWB is often misleading, as it does not provide a full description of the sensor performance. However in this case, it does indicate an improvement over using battery powered anchors.

As described, the UWB system surveys the anchor nodes simultaneously, submitting range measurements for all of the anchors to the publisher as one observation. Therefore, for a network of six anchor nodes, a single dropout would result in the observation containing five range readings. Table 6.2 displays the percentage of observations to contain a number of readings as shown in the columns, for each power configuration, shown in rows.

To begin with, Table 6.2 clearly demonstrated the requirement of greater than minimum anchor nodes, as the percentage readings needed for 2D ranging (three) is only met in 67.0% of the observations on average across the

Configuration	1 readings	2 readings	3 readings	4 readings
1	99.5%	93.8%	65.3%	20.7%
2	99.5%	94.1%	69.4%	26.6%
3	99.7%	93.5%	65.3%	19.7%
4	99.2%	93.1%	67.9%	25.2%

Table 6.2: Table describing number of anchor range readings available in UWB measurement as a percentage of total measurements.

configurations. It is also clear that there is an improvement in dropout rate, as a function of the anchor power configuration. For example, an increase of 5 to 6% is seen in the number of observations containing four readings when the anchors are powered using batteries rather than mains adapters.

It may be seen from Figures 6.3 to 6.6 that the standard deviation and distribution of the range reading is not unusual, and within the expected order of magnitude found in the literature [97]. It may be seen that the principal change in UWB behaviour caused by the method of power supply is the rate of dropouts. Therefore, in future experiments UWB anchors used will be powered through the use of LiPo batteries and DC - DC converters.

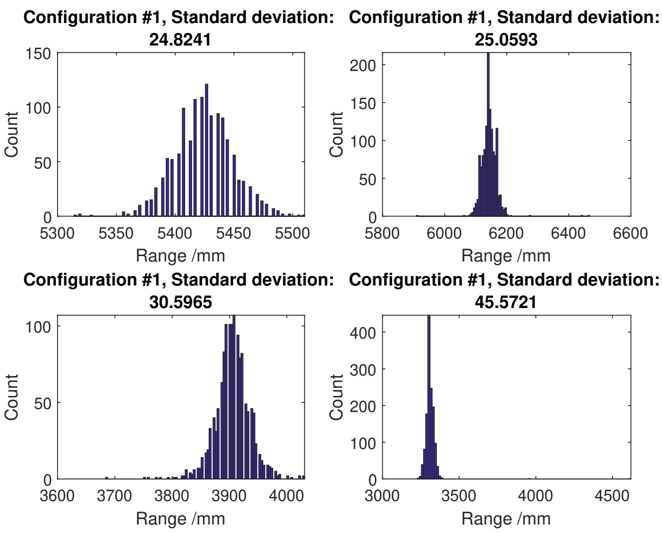


Figure 6.3: Histogram of anchor range readings in configuration 1.

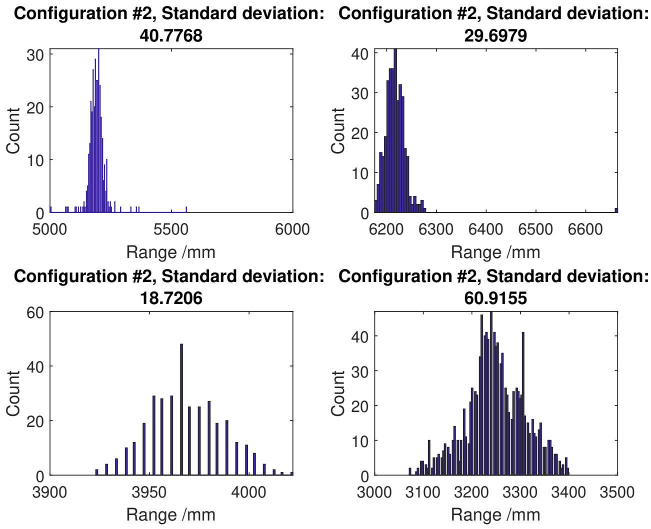


Figure 6.4: Histogram of anchor range readings in configuration 2.

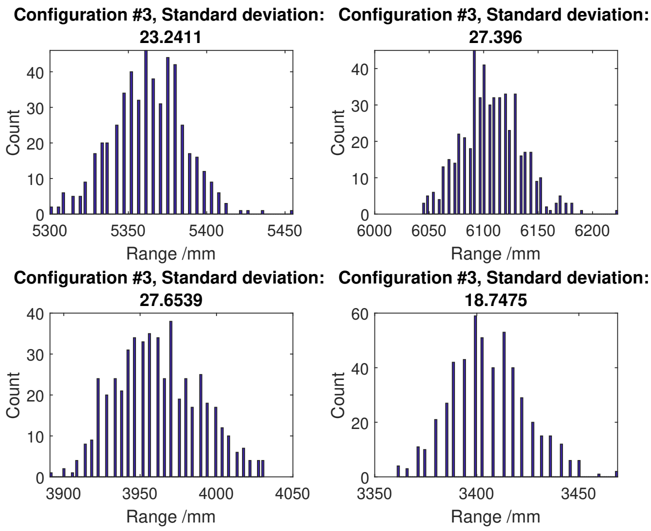


Figure 6.5: Histogram of anchor range readings in configuration 3.

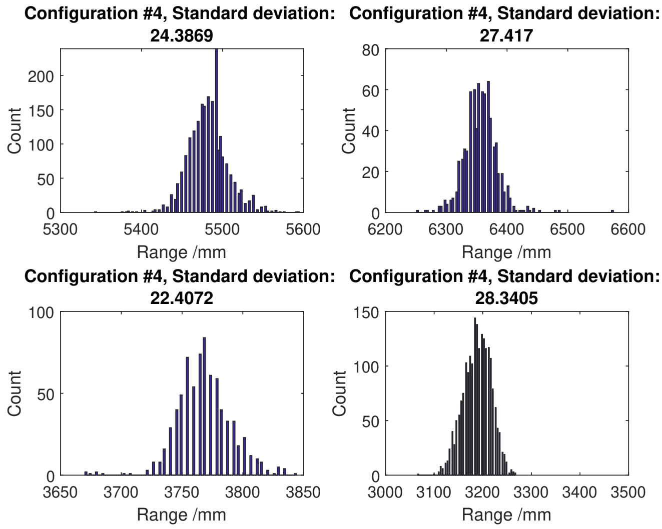


Figure 6.6: Histogram of anchor range readings in configuration 4.

6.1.2 Testing environment

With regards to the environmental conditions in terms of materials and degree of enclosure, it may be seen from the previous tests discussed in Chapter 4 that there is a worsening of the UWB tracking performance when deployed indoors. However, there could be other influential variables associated with the move to indoors spaces. As moving back outdoors would also remove these factors was decided that testing the other possible effects in an indoors setting is the most practical course. The principle factor which may be effecting the reliability in terms of RF systems for the UWB is the presence of WiFi networks. The UWB system operates between 3.4GHz and 6.8GHz, beyond that used for household wireless networks, however the University operates the EDUROAM network, which operates on both the 2.4GHz and 5.8GHz bands. This second 5.8GHz band sits within the operating range used and therefore may be a factor in the decreased performance.

To examine this, the UWB system was moved into domestic environment with 2.4GHz WiFi present. The chosen environment was mostly of plastered stone wall construction on three of the four walls, with the fourth being brick. The ceiling and floor were constructed from wood planks. Two tests were run, the first with four anchors in the space, and the second with six anchors in place. As demonstrated in previous work, battery power provided slight improvement in the UWB performance, therefore the UWB anchors were powered by battery.

The results from the testing in the domestic environment showed an improvement in the noise and dropout rates of the UWB system. The positioning error in x, y, z may be seen below in Figure 6.7. This plot shows the Gaussian distribution expected of the system, however there is still a significant spread of the data. When viewing the data as a whole in terms of the x, y, z error, as seen in Figure 6.8 the standard deviation may be found to be 0.15m. This is in line with the expected value. However, results do not demonstrate the Gaussian distribution modelled by the EKF . When viewing the range data from the four anchors used in this test this is not the case, as seen in Figure 6.9, the distribution of the range errors are strongly Gaussian. This is one of the main reasons for modelling the system in the chosen way. The range based EKF allows for the Gaussian assumption to be maintained, even when the EKF does not present with that type of data in some of its outputs. Similar results are seen when six anchors are used. The positional variation in error is again non Gaussian, as seen in Figure 6.10 and Figure 6.11. Again the standard deviation of the range error for the anchor measurements is within the expected range, as seen in Figure 6.12.

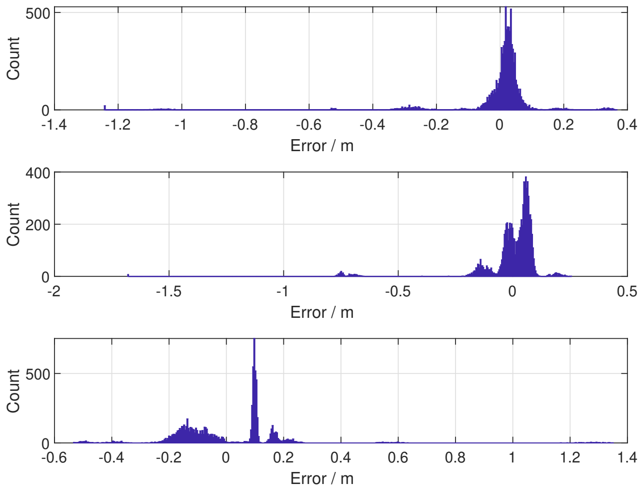


Figure 6.7: Histogram of Pozyx position estimate error in x, y, z axes, from the top to the bottom (four anchors).

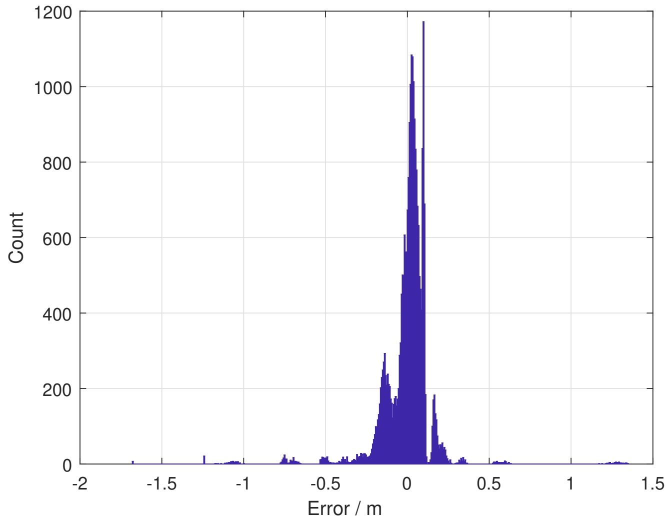


Figure 6.8: Histogram of combined position error (four anchors).

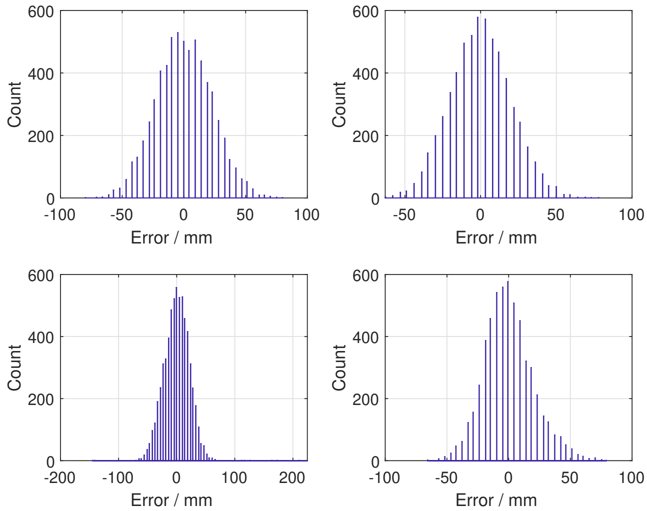


Figure 6.9: Histogram of individual anchor range error (four anchors).

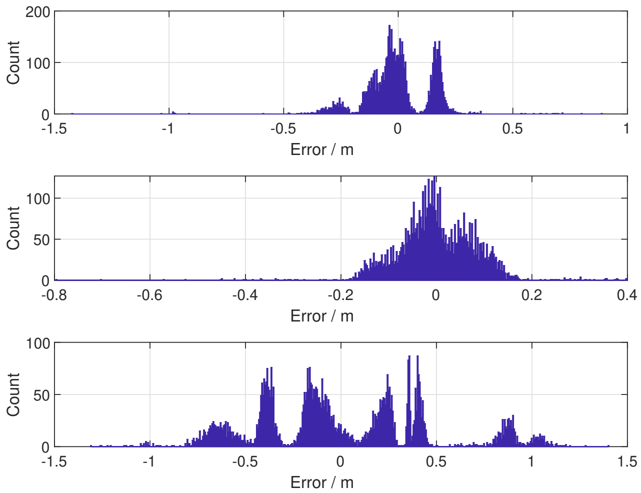


Figure 6.10: Histogram of Pozyx position estimate error in x, y, z axes, from the top to the bottom (six anchors).

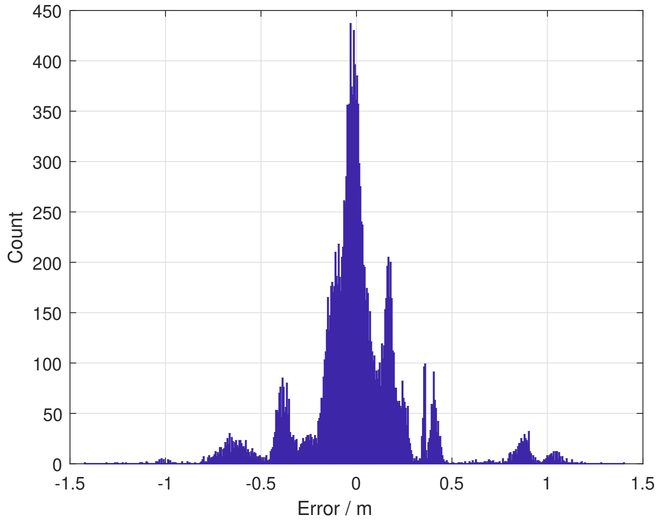


Figure 6.11: Histogram of combined position error (six anchors).

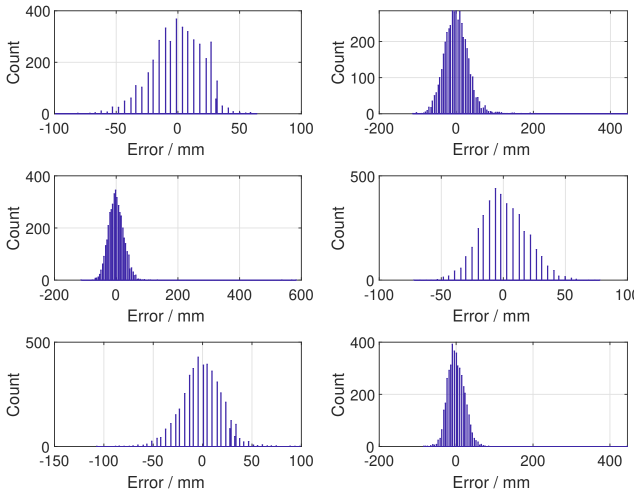


Figure 6.12: Histogram of individual anchor range error (six anchors).

As a final test before moving on from the domestic environment, the rover node was moved about the space in a series of trajectories. This was done to determine whether the results varied significantly about the space. For example, if when moved in an approximately circular path about the room the UWB system became more or less noisy this may indicate that the static readings were outliers. As seen in Figures 6.13 and 6.14 this was not the case. The UWB system performed well and demonstrated a marked improvement in terms of the path. Although no ground truth system was available for this test, it is apparent when the system functions erratically, in this case the x, y, z position readings were smooth and not discontinuous. In the case of the four anchor test, the readings are noisy, however this is to be expected for 3D ranging using four anchors.

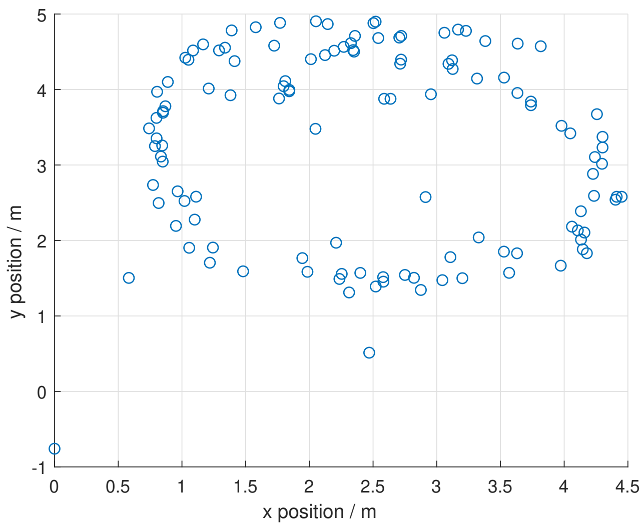


Figure 6.13: Raw Pozyx position estimates for a loop trajectory about the space (four anchors).

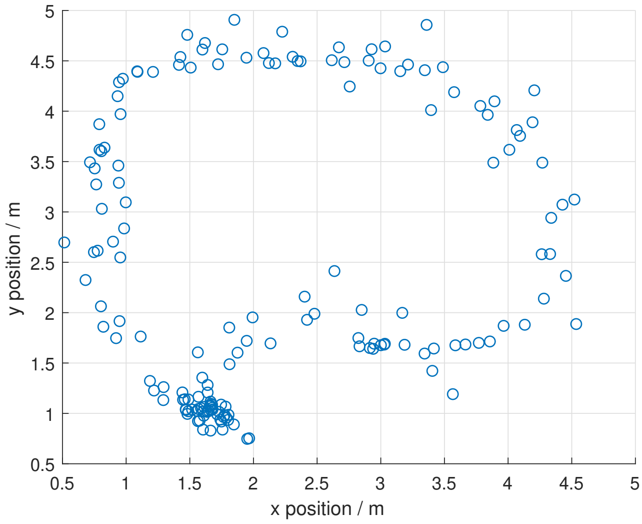


Figure 6.14: Raw Pozyx position estimates for a loop trajectory about the space (six anchors).

To further investigate the effects of the RF environment, and specifically the sections which may have changed when the testing moved to indoors spaces, the UWB system was set up in an EMC chamber. This chamber, seen in Figure 6.15 is designed to absorb all EM waves up to 6GHz. The same system was used, with the Pozyx anchors powered again from battery packs. Six anchors were used, and spread about the space as seen in Figure 6.15. During testing the room was sealed; the door, constructed in the same way as the walls (Figure 6.16), in order to block all external wireless signals. For this test a Taranis radio transmitter was also included in the test to determine if the use of a transmitter would cause a change in performance. Networking and logging was completed through ROS as before, and the communication was achieved through the use of the TPLink access point. The tools used may be seen set up in Figure 6.17.

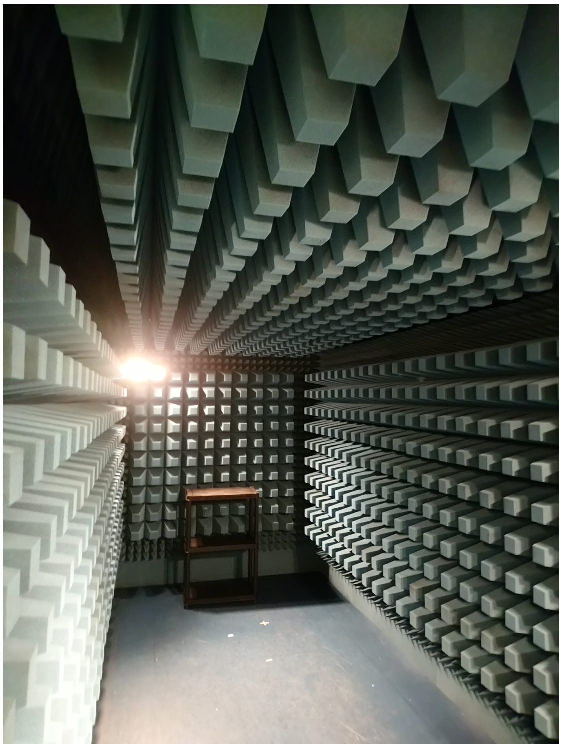


Figure 6.15: Interior view of the EMC chamber.

Figure 6.16: View of door construction for the EMC chamber.

Figure 6.17: View of the setup used for testing transmitter interference.

As the chamber is of a comparatively small size for this project, there is the issue of comparing the range error of varying node to node distances. However, this portion of the investigation is less concerned with the values output for the standard deviation of the range error, and more with the change in range error deviation with respect to the EM environment. Again, no ground truth is available for this test, therefore the range error is taken as the range with the mean subtracted, as the system has been previously demonstrated to have zero mean. Although this may lead to some errors, the main point of this section of the investigation is to determine standard deviation variance, not error variance. As may be seen from the results shown in Figures 6.18 and 6.19 there is a clear difference in range error when the Pozyx system is isolated from the normal EM environment found in the testing spaces used. This demonstrates that the system may be negatively affected by operating systems nearby. Figures 6.20 and 6.21 may be seen as a demonstration that although the Pozyx system range error is affected by the EM environment, the lack of change on the same scale as seen in Figures 6.18 and 6.19 suggests that the Taranis TX (operating at 2.4GHz) does not add significantly to this. The standard deviation values reinforce this concept; as shown in Table 6.3, the difference between the standard deviation of the Pozyx range error when the door is open and closed is a factor of 1000.

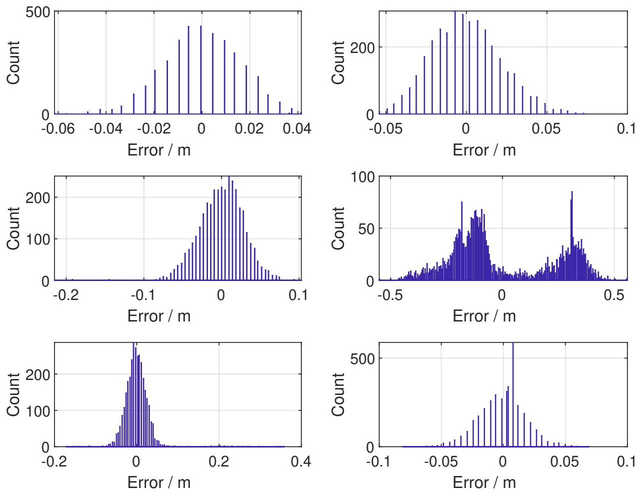


Figure 6.18: Histogram of anchor range error in EMC chamber with open door.

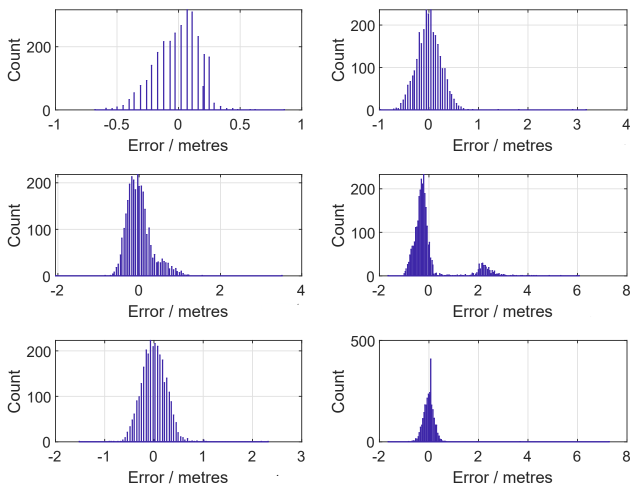


Figure 6.19: Histogram of anchor range error in EMC chamber with closed door.

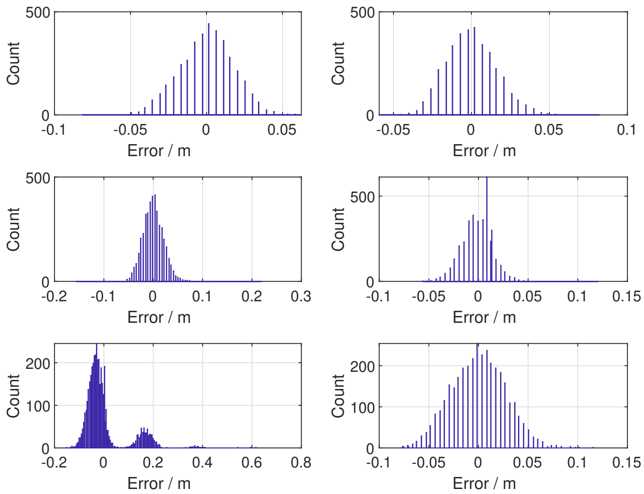


Figure 6.20: Histogram of anchor range error in EMC chamber with open door and Tx on.

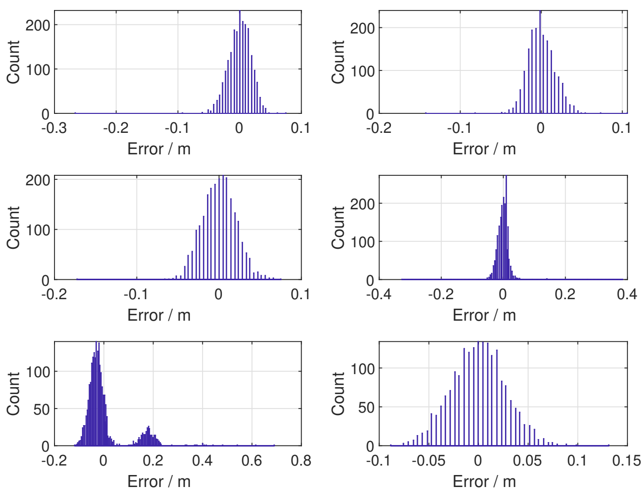


Figure 6.21: Histogram of anchor range error in EMC chamber with closed door and Tx on.

Anchor Number	Closed / m	Open / m	Closed with Tx on / m	Open with Tx on / m
1	1.7644e-05	0.0160	0.0189	0.0177
2	2.5697e-05	0.0216	0.0166	0.0160
3	3.2256e-05	0.0291	0.0206	0.0207
4	9.1204e-05	0.2343	0.0231	0.0150
5	2.5315e-05	0.0316	0.0836	0.0839
6	2.4843e-05	0.0172	0.0275	0.0274

Table 6.3: Standard deviation of range error for different conditions in the EMC chamber.

An unusual reading can be seen in Figure 6.18 for the fourth anchor. In this case a clear double peak may be seen. The overall range reading from this anchor may be seen in Figure 6.22. This second peak, with a mean of roughly 0.45m greater than the first, could be the result of a multipath error due to reflection from an object 0.225m behind the rover node with respect from the anchor. However, in the context of the other results, this value may be reasonably taken as an outlier.

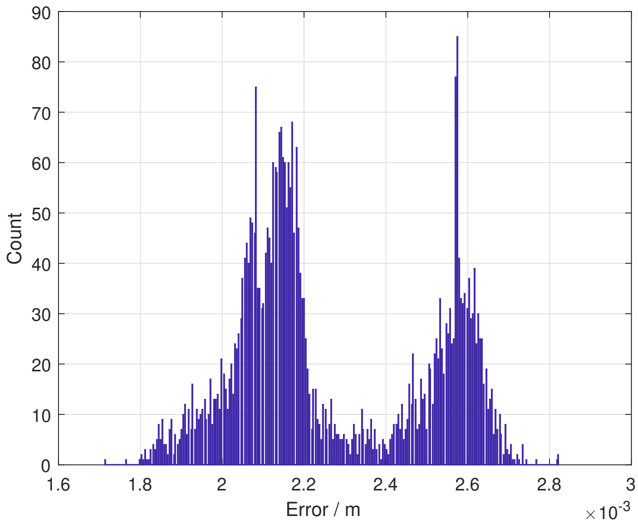


Figure 6.22: Histogram of anchor 4 range in EMC chamber with open door.

6.2 6DOF EKF testing

This section outlines the testing of the effectiveness of the 6-DOF MB-EKF in relation to the standard formulation of the EKF utilising a static sensor variance. The initial subsection outlines the methodology used, building upon the work described previously in the chapter on an improved set up of the UWB system. Next, it will describe and demonstrate the results from these tests, moving on to an analysis of these results.

6.2.1 Methodology

The testing procedure for this section will follow closely from that used in the previous chapter, however modifications to the experimental setup have been made to avoid the issues encountered in the previous chapter.

6.2.1.1 Experimental Setup

From the work conducted on determining the causes of the increased sensor noise found in the previous chapter, it is clear that several modifications to the experimental setup are prudent. Firstly, a different room is to be used as the testing space. The indoor testing space used in the previous set of tests seen in Chapter 5, contains multiple access points for the EDUROAM network, which may have contributed to the sensor noise. Secondly, the anchors used should be powered through the use of battery packs. One of the main features presented during testing of the initial noisy data was that the dropout rates increased. For the intended space nine anchors have been chosen, an increase of 50% on previous tests. Therefore thirdly, a step which may be taken to setup a robust sensor system is the inclusion of more anchor nodes. Furthermore, these anchors will be spread more evenly in the x, y, z axis, as seen in Figure 6.25. In the previous tests, it is clear that the anchors were not placed with enough variation in the z axis for accurate 3D localisation. Finally, the new testing space should be of a smaller volume; this will allow for operation of the rover node within the range of stability of the sensor variance model constructed in Chapter 4. This may not necessarily improve the functionality of the system, however it may lead to less erratic results. This is because in Chapter 4 the sensor range error is found to become less predictable beyond certain ranges.

Based upon these criteria, a new indoor testing space was selected. The space has no network access points present in the room, is large enough for

UAV flight and, although it has large windows on two of its four walls, these may be covered using fabric screens, as seen in Figure 6.23.

Figure 6.23: Screens used to cover the windows of the new indoors testing space.

The RTS was set up in the corner of the room to allow for maximum unimpeded coverage of the space, as seen in Figure 6.24. The Pozyx anchors were placed with variation in height about the room, and powered from the same DC-DC converters used in Section 6.1.1. An example of the range in anchor placements may be seen in Figure 6.25. The initial starting point for the UAV was chosen as the centre of the room. This allowed for safer take-off and landing, therefore all trajectories will begin and end at this location. The trajectory may be split during the EKF testing phase to examine alternate trajectories.

Figure 6.24: RTS setup used for monitoring the new indoors testing space.

(a)

(b)

Figure 6.25: Example anchor placement in (a) low position, and (b) high position.

6.2.1.2 Testing procedure

As described in Section 6.2.1.1, a number of modifications to the experimental setup have been made between this and the previous experiment. These modifications are not limited to the setup of the room, but also include the method of carrying out the experiments. In order to collect the most reliable UWB data possible for this setup the UAV will be flown at a lower velocity, to increase the density of UWB points in terms of points per distance of trajectory covered.

6.2.2 EKF Formulation Alterations

The previous work outlined in this chapter leads to the suggestion of an increased number of UWB anchors to improve the EKF performance and UWB stability. Therefore, as stated in Section 6.2.1.1, nine anchors was be used. This leads to the requirement for an alteration to the measurement phase of the EKF. Specifically, this change is made to the measurement Jacobian matrix. Currently the matrix is of dimensions 10 by 6. For this phase of testing this will need to be increased to 10 by 9 to accommodate the increased observation inputs provided by the 9 anchor ranges. As before, the Jacobian of the observation matrix is computed offline in MATLAB, this was incorperated into the EKF code seen in the appendix of this work.

6.2.3 Results and Analysis

This section will first describe the collected data from the latest set of experiments outlined thus far in the chapter. The data collected was then processed through the use of the 6-DOF EKF formulated in the previous chapter; a comparison is then be made between the performance of the standard filter and a filter incorporating the model based sensor variance as described in Chapter 4. This section will also demonstrate the effects of the alternate representation of anchor - rover node range observations on the performance of the EKF . Finally, the section will conclude with a comparison and analysis of the effects of the inclusion of a pre-filtering stage into the EKF algorithm.

6.2.3.1 Sensor data output

In total, 25 trajectories are recorded with 5 of these trajectories carried out using the UGV: 15 with the UAV and 5 with the UAV carried as initial testing. The UWB system performed well, with minimal outliers compared to all other tests shown previously, and good tracking of the platforms with respect to the RTS system. Three example plots may be seen in Figures 6.26 to 6.28, showing a trajectory taken with the UGV, and two with the UAV platform. These results show that the work conducted in this chapter to improve the UWB performance is successful, and results in a clear improvement to the UWB operation, showing a clear correlation between UWB reading and RTS ground truth.

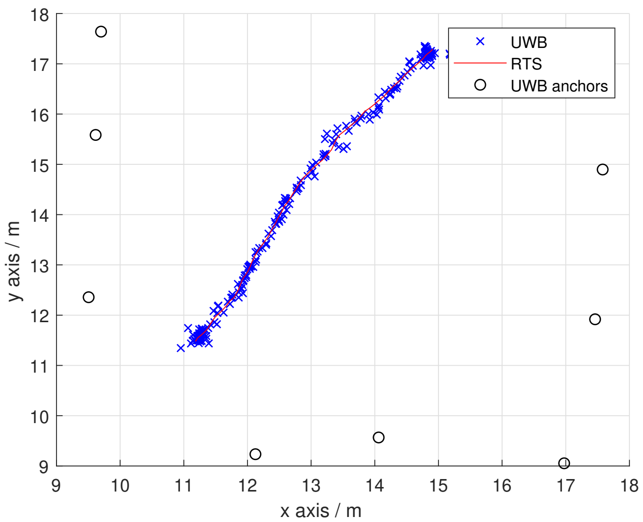


Figure 6.26: Plot showing UWB data and RTS ground truth for UGV trajectory.

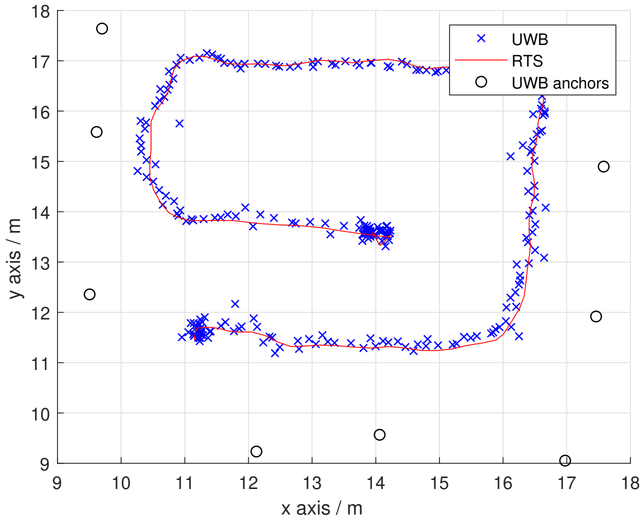


Figure 6.27: Plot showing UWB data and RTS ground truth for UAV trajectory.

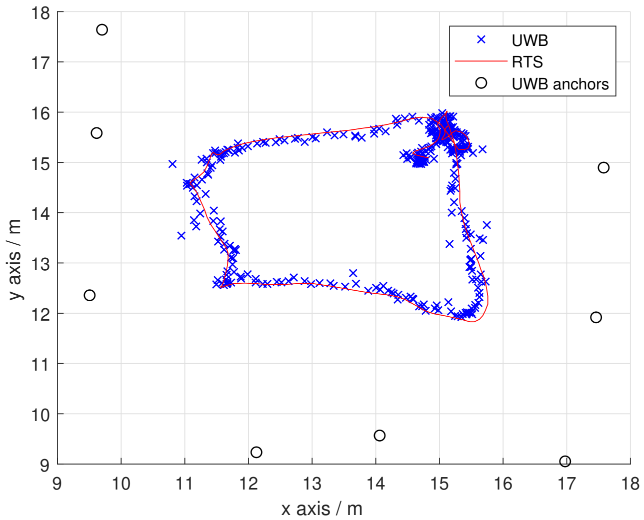


Figure 6.28: Plot showing UWB data and RTS ground truth for UAV trajectory.

A 3D plot may be seen in Figure 6.29, along with 2D plots of the data viewed in the x, y plane, x, z plane, and y, z plane shown in Figures 6.29 and 6.30. These plot demonstrates that the UWB system is particularly effective in the x, y dimensions, whilst less accurate in the z axis.

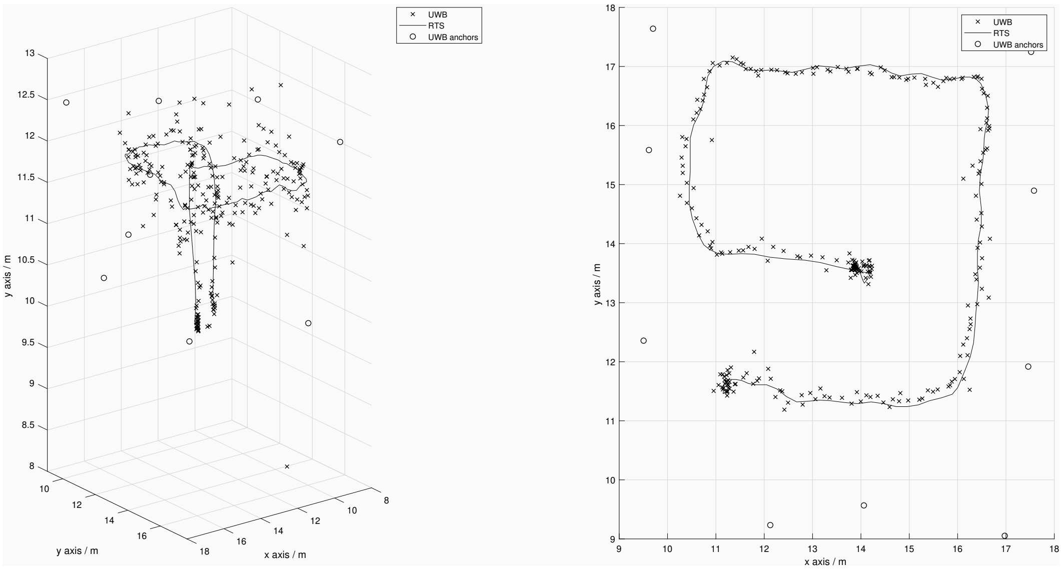


Figure 6.29: Plot showing UWB data and RTS ground truth for UAV trajectory.

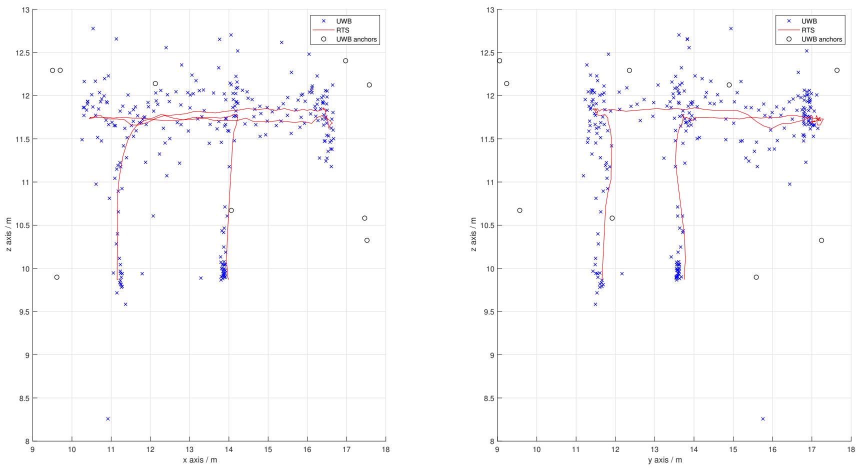


Figure 6.30: Plot showing UWB data and RTS ground truth for UAV trajectory.

6.2.3.2 6DOF EKF versus MBEKF

Having presented the results of the modifications to the UWB set up procedure and change in environment, this section will show the results of the EKF and MBEKF when applied to the datasets collected. First the EKF will be tuned using the standard sensor variance. Once this is complete a number of trajectory datasets will be tested with both the MBEKF and the standard formulation of the range based EKF first presented in Chapter 3. The tuning will take place using the standard formulation of the EKF to avoid over optimisation of the parameters. As this investigation aims to determine the effects of the inclusion of the model based sensor variance it is logical to construct a working filter in the standard manner. This filter would then be modified with the new sensor variance and the changes analysed.

Figure 6.31 shows an interesting feature in the EKF prior to tuning of the system variance matrix. It can be seen that the z axis estimate is highly erroneous to begin with, while eventually converging and stabilising near to the ground truth value for the trajectory. This effect is clearly visible mostly in the z axis, however may be seen in the x, y plane also. A reasonable conclusion which may be drawn is that due to the initial pause in the trajectory, a number of UWB measurements are gathered which do not directly agree with the prediction based upon the control input. In this case the Kalman gain would become erratic and may slip into the negative. This hypothesis is supported by Figure 6.32, which shows the z axis prediction plotted with the average Kalman gain against time step. It may be seen in this figure that the Kalman gain is initially erratic, sweeping into the negative during the same period as the z axis instability.

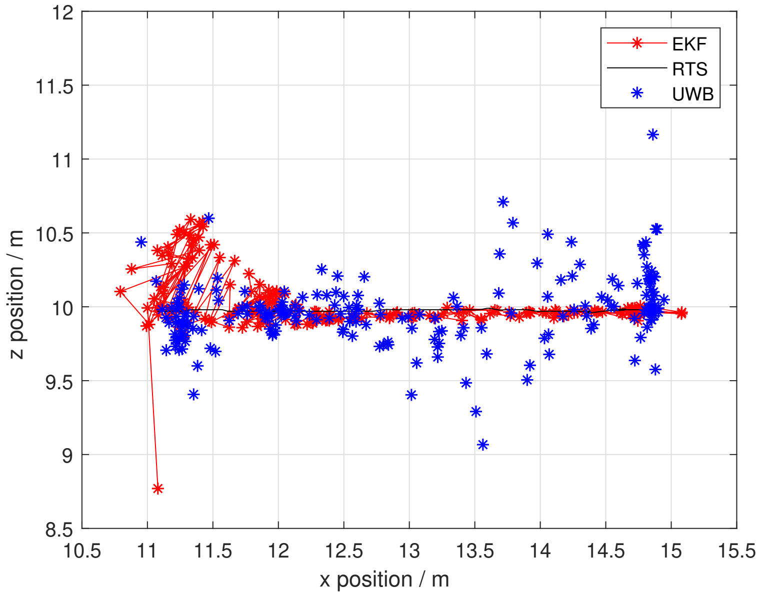


Figure 6.31: Initial trajectory running from left to right with RTS, UWB and prior to tuning MBEKF in position estimates in x, z plane.

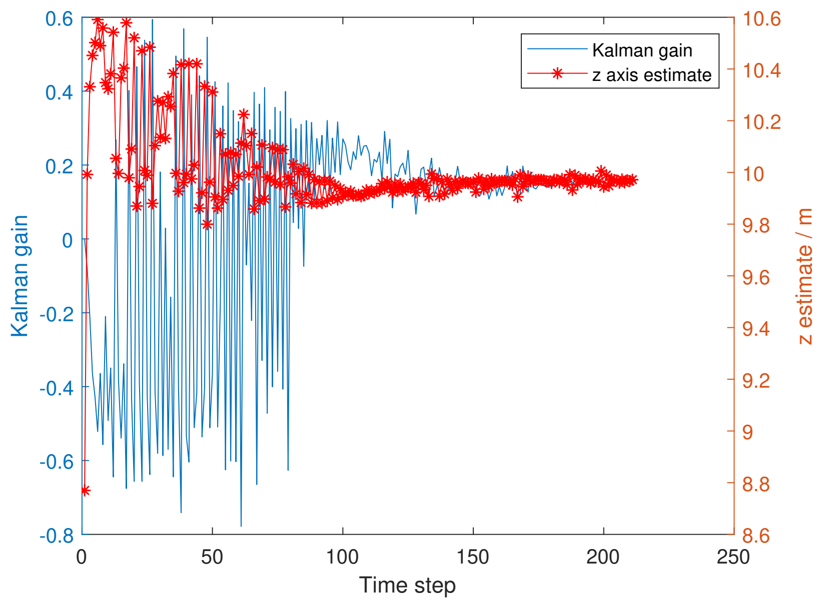


Figure 6.32: Mean Kalman gain and z estimate against time step.

Tuning of the MBEKF results in a clear improvement in the estimation of platform position in the z axis when using the MBEKF formulation as seen in Figure 6.33. In this case the MBEKF estimate for the z axis is significantly more stable than that of the standard formulation UWB, however the estimate is roughly the same as the UWB measurement in x and y. The MBEKF is more stable and reliable than the standard formulation EKF , as again demonstrated by Figures 6.33 to 6.36. It should be noted that in Figure 6.34 the leftmost EKF points diverge from the RTS ground truth more so than the rest of the trajectory; in this region the UWB experienced a temporary increase in outliers. This increase was handled more effectively by the MBEKF as demonstrated in previous chapters. The same outliers may be observed in Figure 6.35, indicating a region of poor UWB performance. The results presented here indicate the use of the MBEKF improved resilience to such regions due to the implementation of the sensor variance model.

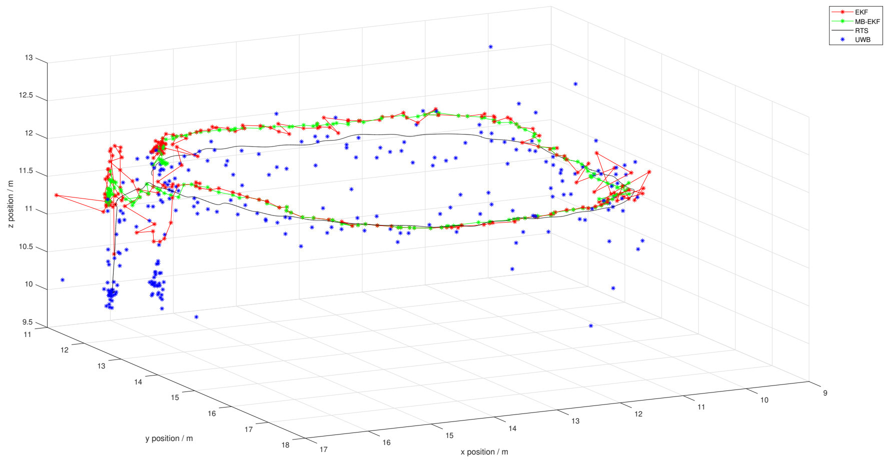


Figure 6.33: 3D plot showing the estimate outputs from the standard formulation of the EKF , the MBEKF, the UWB, and the RTS ground truth.

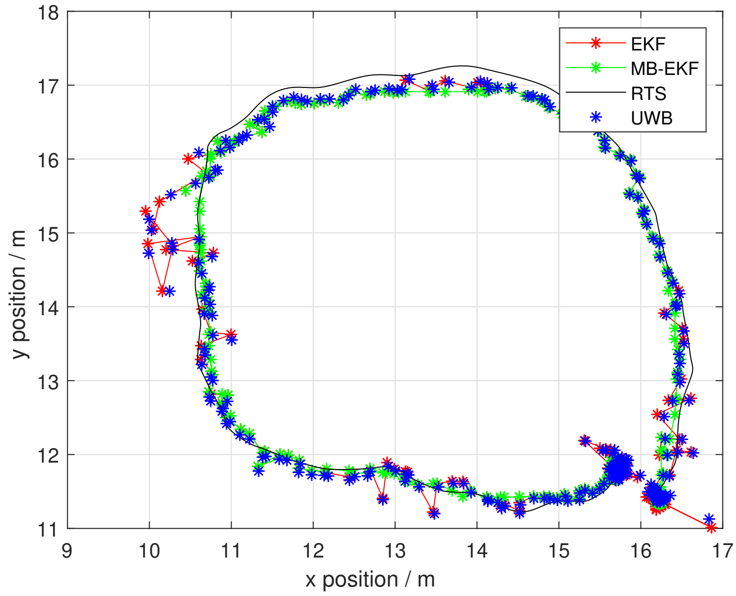


Figure 6.34: 2D plot showing x, y the estimate outputs from the standard formulation of the EKF , the MBEKF, the UWB, and the RTS ground truth.

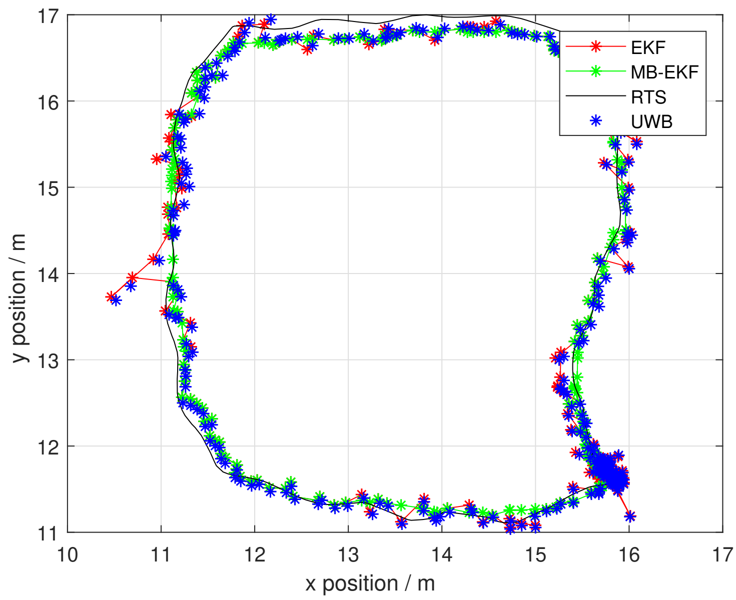


Figure 6.35: 2D plot showing x, y the estimate outputs from the standard formulation of the EKF , the MBEKF, the UWB, and the RTS ground truth.

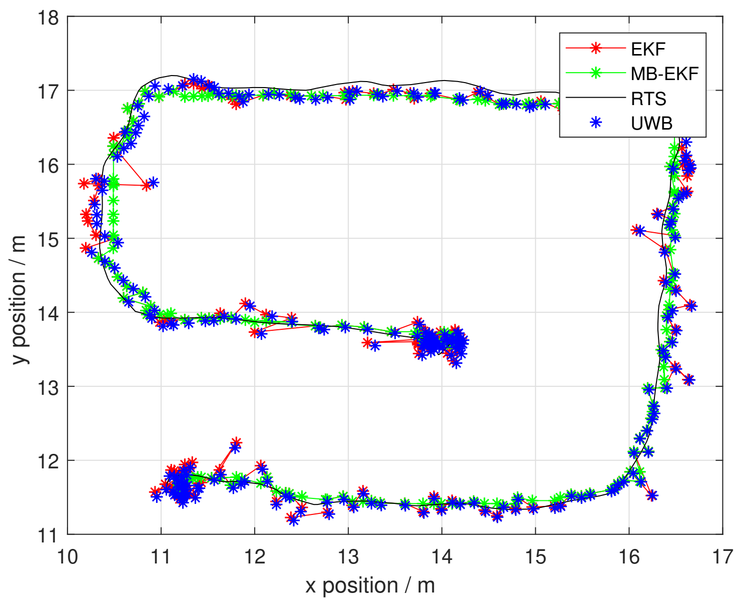


Figure 6.36: 2D plot showing x, y the estimate outputs from the standard formulation of the EKF , the MBEKF, the UWB, and the RTS ground truth.

Tables 6.4 and 6.5 describe the mean error, and standard deviation of the error in each axis for the three measurement sources (UWB, EKF , MBEKF), averaged across the trajectories presented here. Over the course of several of the trajectories, the UWB demonstrates significantly lower error than found in the previous experiments, this may be observed by the closer correlation between RTS ground truth and UWB observations. This is due to the time taken at the beginning of this chapter to optimise the set up process and configuration. Tables 6.4 and 6.5 show an improvement in the performance in terms of the average mean error across the trajectories tested when the model based sensor variance is included. The improvement mentioned in the z axis may be most clearly seen in the average standard deviation of the error across the trajectories seen in Table 6.5. Here there is a factor of ten decrease in the deviation in the z axis when the UWB data is processed through the EKF , and a decrease again when the sensor variance model is implemented. The increased mean error in the z axis seen in Table 6.4 for the UWB in comparison to the EKF and MBEKF is most likely due to an offset from the ground truth. The UWB observation is described in the literature and has been shown previously to be of zero mean, therefore the low values seen are to be expected. This noise may be seen demonstrated in Figure 6.37, where the 3D Euclidean distance between the estimates are plotted. The UWB is clearly seen to be the most volatile in terms of the change in observation position, with the MBEKF estimate being the most stable, in comparison to the UWB and conventional EKF estimates. When taking the estimates in terms of 2D x, y space, the UWB is seen to be significantly less noisy, as seen in Figure 6.38. The MBEKF is still the least volatile in terms of the distance between observations, however the EKF and UWB estimates and observations are seen to be more linked than in the 3D plot. This may be seen particularly in the left region of Figure 6.36, where the UWB observation deviates from the RTS. In this area of the trajectory the conventional EKF follows the UWB measurement, whereas the MBEKF remains closer to the ground truth. This is reasonable as the EKF primarily stabilises the observations in the z axis, with some improvement in the x and y axes, as seen in Table 6.4.

Source	x **axis / m**	y **axis / m**	z **axis / m**
UWB	0.0469	0.0211	0.0070
EKF	0.0679	0.0174	0.0966
MB-EKF	0.0371	0.0014	0.0605

Table 6.4: Table describing mean error in each axis for the UWB, EKF , and MBEKF across all of the tests.

Source	x **axis / m**	y **axis / m**	z **axis / m**
UWB	0.1608	0.1288	0.4833
EKF	0.1510	0.1429	0.0657
MB-EKF	0.1508	0.1275	0.0461

Table 6.5: Table describing mean standard deviation of the error in each axis for the UWB, EKF , and MBEKF across all of the tests.

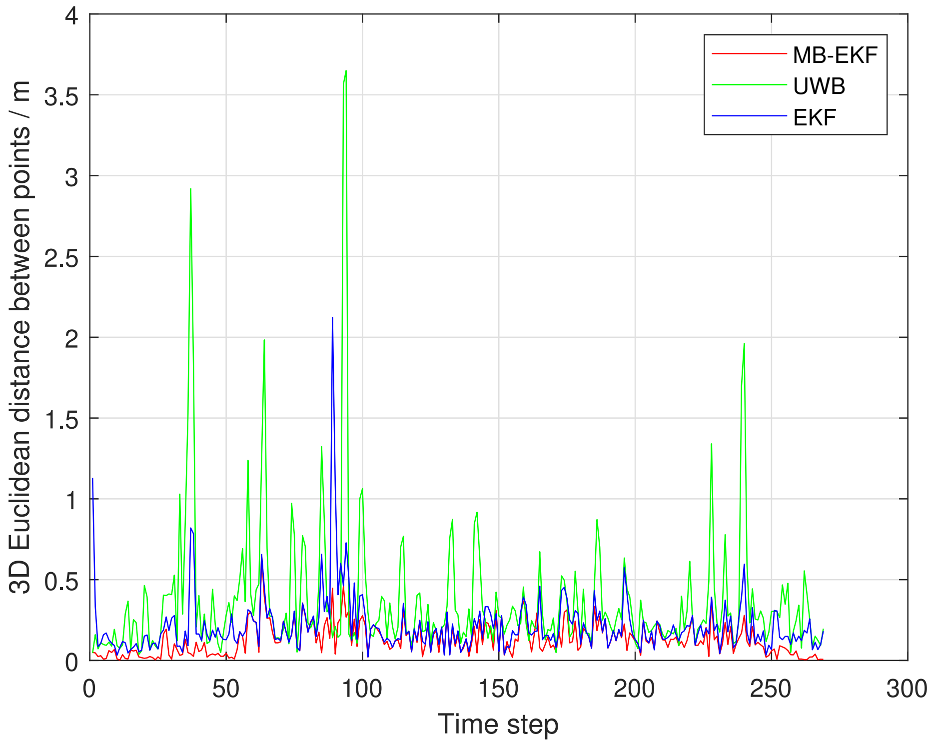


Figure 6.37: Euclidean distance between estimates and observations for EKF , MBEKF and UWB in 3D.

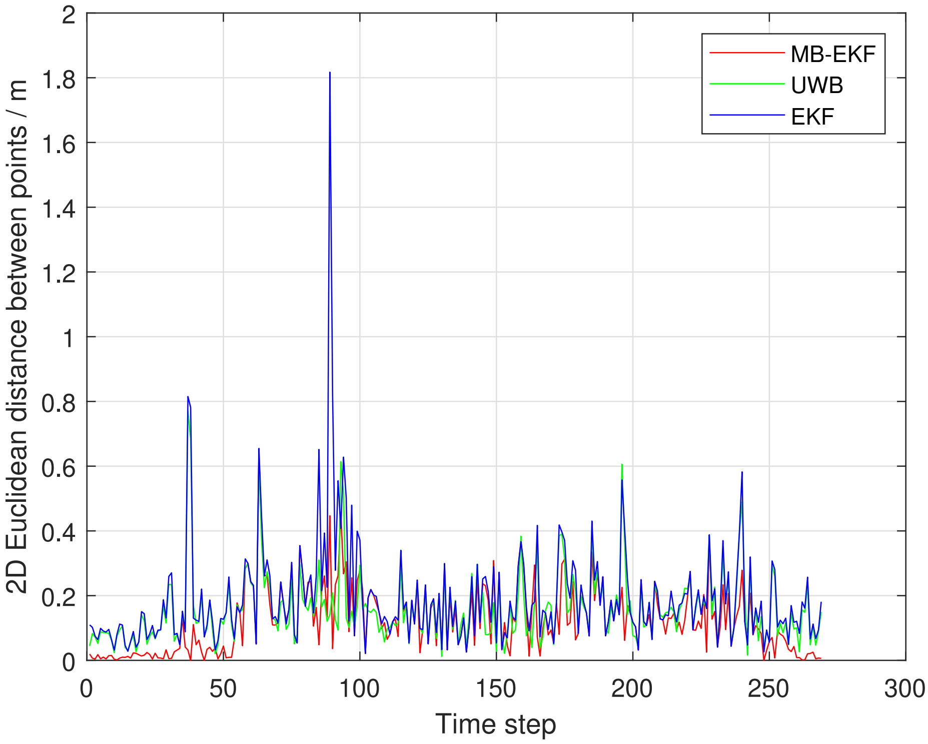


Figure 6.38: Euclidean distance between estimates and observations for EKF , MBEKF and UWB in 2D.

As seen in Figure 6.39, the sensor variance model varied throughout the trajectory seen in Figure 6.34. As mentioned previously, there is a section on the leftmost side of the path where the UWB estimate degrades in accuracy, which adversely affects the standard EKF formulation. The model based EKF is less affected by this increase, due to the altered sensor variance decreasing the trust placed upon observations in this region. As seen in Figure 6.39, the corresponding time steps (85 - 110) show an increased sensor variance in four of the six anchors.

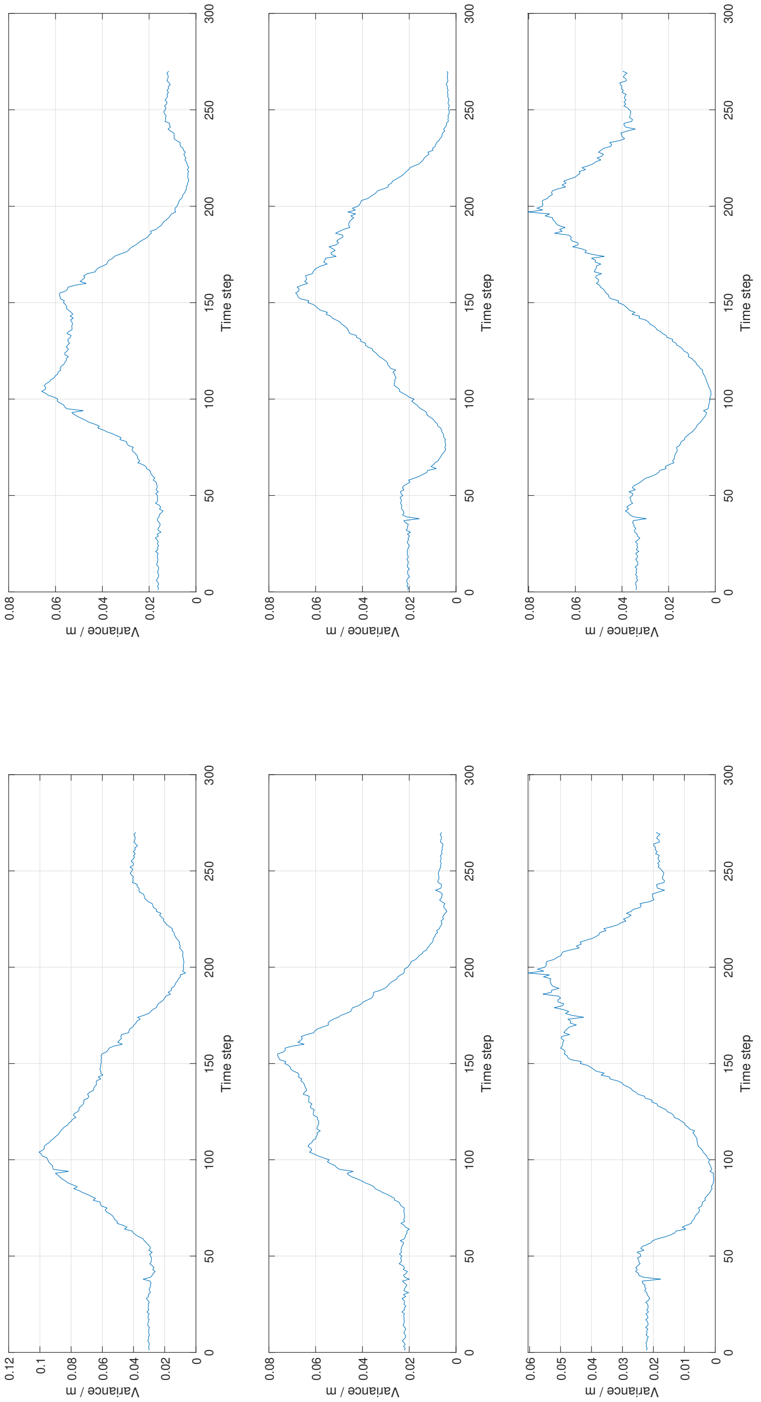


Figure 6.39: Plot showing the sensor variance estimates for the anchors 1 to 6 (clockwise from top left), throughout the testing trajectory shown in Figure 6.34.

6.3 Conclusions

This chapter set out to conclude the project by investigating and resolving the issues encountered in the previous chapter, which led to the inconclusive results in the testing of the MB-EKF on a UAV platform with 6-DOF. In this area the aim was achieved. The initial sections dealt with possible causes of the increased UWB error rate, and identified a number of practical measures which may be implemented to increase the robustness of the UWB network and improve performance. By utilising a different testing space, which allowed for a less noisy RF environment, increasing the number of anchors used, along with an improved distribution, the UWB system performed more reliably and stably than in previous chapters of the project. This allowed for a thorough investigation of the performance of the 6-DOF EKF both with and without the inclusion of the model based sensor variance.

It has been shown that the MB-EKF performs more stably than the EKF , and the basic observations made by the UWB system, filtering increased noise more effectively than the static sensor variance formulation.

The results shown in this chapter differ from the results seen in previous chapters for a number of possible reasons. Firstly, the new EKF utilises anIMU alone for the control input, and not wheel encoders. Therefore the system lacks a secondary measurement input of the platform position. Secondly, the UWB system has been optimised in terms of the set up and configuration as described in this chapter. Finally, the UWB system performs less effectively in the z axis as described previously, and need further improvement.

An important point reiterate is the aim of this section of the project. While the objectives do include a performance requirement, the overall aim is to investigate the effects of the incorporation of the improved sensor variance model. While alterations to the formulation of the filter and different or additional control and observation inputs may improve the performance, they could also obfuscate the changes in behaviour of the filter. The key point of interest is the degree of, if any, performance improvement which may be gained through the inclusion of the sensor variance model. Future work such as the inclusion of alternate or additional inputs will be discussed in the final chapter.

Chapter 7

Conclusions

As presented in the introduction of this thesis, the overall aim of this project was to demonstrate an alternate method for improving state estimator performance, through the use of an improved sensor variance characterisation. The goal is to determine the effectiveness of employing a general system model, and retaining low cost sensors, while still improving estimator performance. This aim, and the objectives found in Chapter 1 are the principle factors upon which success of this project is judged.

The research presented here has accomplished the principle aim of assessing the effectiveness of a sensor variance characterisation in improving estimator performance. Particularly, this has been accomplished when using a general system model, as opposed to the traditional method of further specifying the system. This project reviewed existing methods of representing variance for the sensor in question, and developed a novel representation within the EKF framework, informed by the empirical data collected. This was accomplished firstly, by demonstrating that an RTS system is an effective ground truth tool for both sensor characterisation and state estimator performance assessment. These results were then published, peer reviewed and demonstrated in Chapter 3.

This activity partially fulfilled the objective of formulating a platform specific state estimator, along with its deployment and assessment. Chapter 3 also partially fulfilled the objective to investigate of the use of more detailed sensor characterisation within a state estimator. The EKF formulated and experimentally validated in Chapter 3, was then used to assess the change in performance when employing a static in comparison to a model based sensor variance, the results of this experiment were also disseminated in a journal paper, and may be seen in Chapter 4. This section of the project completed the second and third objectives aimed at studying an alternate representation of the sensor model and comparing this to the traditional form. Furthermore, the work presented here demonstrates the benefits of incorporating a model based sensor variance, along with highlighting areas where this approach is limited. Such as in cases of unusual sensor noise, as demonstrated in Chapter 5. However, the incorporation of a model based sensor variance has been shown to improve performance in cases where sensor noise is higher than expected, as studied in Chapter 4. Chapter 5 further completed the objective set out to formulate a platform agnostic state estimator, with the testing of this algorithm continued in Chapter 6. Through the work conducted in both Chapters 5 and 6 it was found that the use of a generalised state estimator with a model based sensor variance was not only possible, but effective, with a clear improvement when using the model based sensor variance. Finally, this work accomplished the objective to explore the use of the developed sensor variance representation in a generalised, platform agnostic state estimator.

This work has developed a methodology which may be used for the characterisation the sensor uncertainty represented in state estimators. Traditionally, as stated in the earlier chapters, sensor variance is taken as a generalised value, usually given by the sensor manufacturer. However, as is the case here, the sensor variance is rarely constant, even within the same system. Through a thorough literature review, and experimental testing, it was demonstrated that through the use of an RTS the sensor variance may be more precisely determined, allowing for improved state estimation.

This same technique was also shown to be an improvement upon standard methods for examining state estimator performance in large, outdoor environments. Through the use of the RTS, a ground truth was measured of orders of magnitude greater accuracy than the sensor under assessment. This tool also allows for a degree of certification to be incorporated within the assessment, as the RTS used in this investigation is calibrated to an international standard. This gives the potential for the development of a framework for sensor assessment which may be used in the determination of safety in UAV operation.

With these tools and methodologies developed, the key finding, and development of this work was the use of a model-based sensor variance within the EKF framework. This work demonstrates that the improved characterisation of sensor variance, through the use of novel application of existing tools, in the intended environment, allows for a more robust, more accurate state estimator. Furthermore, this work demonstrates a formulation of the EKF which allows for the incorporation of the model-based sensor variance in a way that gives a generalised state estimator for use in both ground and aerial vehicles. This would not be practical through the use traditional state estimator frameworks, as the error in the sensor variance is made up for in a more specific state transition function. Within this, it is concluded that the use of the estimated sensor variance as the input to the sensor variance function is also appropriate, as it allows for a degree of resistance to outliers that may not be present when directly using the sensor measurements as the means for calculating the sensor variance.

During this work it was found that the UWB system was subject to a number of factors to a greater extent than documented in the relevant literature. Specifically, as seen in chapter 5, the EDUROAM wifi system caused a significant increase in measurement noise. In conjunction with an increased prevalence in multipath related errors due to the testing environment used in this segment of the testing the UWB system was rendered significantly less reliable than in previous work. Testing demonstrated there was a noticeable difference in the UWB performance when operating in an space isolated from the EDUROAM wifi network, in combination with the new space containing more access points to this network than previous test spaces it was concluded that this was a factor. Later tests, seen in chapter 6 support this hypothesis.

7.1 Future work

The research presented here has demonstrated the effectiveness of using a model based sensor variance, and the concept of a generalised state estimator. However, as this research was confined principally to the investigation of the use of a model based sensor variance to allow for the use of a generalised state estimator, there were some areas that fell outside of the scope of this research. First and foremost, is the application of this approach to sensors other than the UWB system. As was described throughout the project, the UWB system was not the focus, and the research is not primarily concerned with this system. It was chosen due to its affordable, self contained nature, and transparent measurement model, along with the available literature regarding its performance and application.

From this, the logical next stages of this work are to begin to test this methodology when applied to different sensors used in UAV navigation. For example, many UAV platforms operate through the use of GNSS systems. This system was not investigated in this project due to the substantial requirements of characterising the measurement variance of a GNSS as a function of the operating environment. GNSS accuracy is known to be subject to a number of factors, such as satellite constellation geometry, the number of satellites in use, both solar or planetary weather, and local interference to name but a few. However, for future work, this area of study would allow for a more full representation of the most ubiquitous measurement input for UAV navigation systems and, as demonstrated here, improve the reliability of UAV navigation. This would fulfil one of the requirements of CAP722 as outlined in Chapter 1.

A second avenue of investigation that would be beneficial to this field and is highlighted by this research, would be the modelling of the measurement variance of computer vision based navigation system inputs. Currently, visual navigation is a widespread tool for UAV positioning; however the reliability of the measurements are often dependant upon the tracking of feature points used for each camera frame, required in techniques such as monocular visual odometry. This in turn is also linked to factors such as lighting, colour, and rate of change between frames. The generation of a model of this in measurement variance would allow for a more reliable incorporation of such measurement inputs into state estimators.

An overall extension of this work may also be achieved through the integration of other, existing strategies such as that used by the Adaptive Kalman Filter. The current system allows for a better estimate of the sensor variance through empirical data collected prior to the state estimation. However, if the environment were to change, or in the case of the UWB system, multiple sensors were in use, the model used may not be fully representative of the sensor variance. However, through the use of the variance calculation component of the AKF, an ongoing estimate of the variance may be calculated and compared with the value provided by the sensor variance model. This may allow for an active update component whereby the adaptive variance estimate may be used to update and correct the model.

Bibliography

[1] J. P. Guimarães, T. L. Laura, A. S. Sanca, A. N. Schildt, M. S. De-Deus, P. J. Alsina, A. T. Da-Silva, and A. A. Medeiros, ”Fully autonomous quadrotor: A testbed platform for aerial robotics tasks,” *Proceedings - 2012 Brazilian Robotics Symposium and Latin American Robotics Symposium, SBR-LARS 2012*, pp. 68–73, 2012.

[2] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, ”The GRASP Multiple Micro UAV Testbed,” *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[3] Z. Pan, L. An, and C. Wen, ”Recent advances in fuel cells based propulsion systems for unmanned aerial vehicles,” *Applied Energy*, vol. 240, pp. 473–485, apr 2019.

[4] G. Loianno, Y. Mulgaonkar, C. Brunner, D. Ahuja, A. Ramanandan, M. Chari, S. Diaz, and V. Kumar, ”Smartphones power flying robots,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1256–1263, IEEE, sep 2015.

[5] E. Petritoli, F. Leccese, and L. Ciani, ”Reliability assessment of UAV systems,” in *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 266–270, IEEE, jun 2017.

[6] Civil Aviation Authority, ”Taking Flight: The Future of Drones in the UK - Moving Britain Ahead,” tech. rep., Department for Transport, London, 2018.

[7] Federal Aviation Administration, ”No TitleFAA Aerospace Forecast: Fiscal Years 2019-2039,” tech. rep., Federal Aviation Administration, Washington, 2019.

[8] G. Schmitz, T. Alves, R. Henriques, E. Freitas, and E. El’Youssef, ”A simplified approach to motion estimation in a UAV using two filters,” *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 325–330, 2016.

[9] C. F. M. A. Olivares-mendez and R. S.-f. P. Campoy, ”Monocular Visual-Inertial SLAM-Based Collision Avoidance Strategy for Fail-Safe UAV Using Fuzzy Logic Controllers Comparison of Two Cross-Entropy Optimization Approaches,” pp. 513–533, 2014.

[10] D. Mellinger and V. Kumar, ”Minimum Snap Trajectory Generation and Control for Quadrotors,” pp. 2520–2525, 2011.

[11] J. Q. Cui, S. Lai, X. Dong, and B. M. Chen, ”Autonomous Navigation of UAV in Foliage Environment,” *Journal of Intelligent & Robotic Systems*, vol. 84, pp. 259–276, 2015.

[12] M. Barczyk, M. Jost, D. R. Kastelan, A. F. Lynch, and K. D. Listmann, ”An experimental validation of magnetometer integration into a GPS-aided helicopter UAV navigation system,” in *Proceedings of the 2010 American Control Conference*, pp. 4439–4444, IEEE, jun 2010.

[13] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, ”Consistency of the EKF-SLAM Algorithm,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3562–3568, IEEE, oct 2006.

[14] J. C. Hodgson, S. M. Baylis, R. Mott, A. Herrod, and R. H. Clarke, ”Precision wildlife monitoring using unmanned aerial vehicles,” *Scientific Reports*, vol. 6, p. 22574, mar 2016.

[15] S. Baena, D. S. Boyd, and J. Moat, ”UAVs in pursuit of plant conservation - Real world experiences,” *Ecological Informatics*, vol. 47, pp. 2–9, sep 2018.

[16] N. Ratcliffe, D. Guihen, J. Robst, S. Crofts, A. Stanworth, and P. Enderlein, ”A protocol for the aerial survey of penguin colonies using UAVs,” *Journal of Unmanned Vehicle Systems*, vol. 3, pp. 95–101, sep 2015.

[17] G. J. Thapa, K. Thapa, R. Thapa, S. R. Jnawali, S. A. Wich, L. P. Poudyal, and S. Karki, ”Counting crocodiles from the sky: monitoring the critically endangered gharial (<i>Gavialis gangeticus</i>) population with an unmanned aerial vehicle (UAV),” *Journal of Unmanned Vehicle Systems*, vol. 6, pp. 71–82, jun 2018.

[18] G. Tomlins and Y. Lee, ”Remotely Piloted Aircraft - an Inexpensive Option for Large-Scale Aerial Photography in Forestry Applications,” *Canadian Journal of Remote Sensing*, vol. 9, pp. 76–85, dec 1983.

[19] G. F. Tomlins, ”Some Considerations in the Design of Low Cost Remotely-Piloted Aircraft for Civil Remote Sensing Applications,” *The Canadian Surveyor*, vol. 37, pp. 157–167, sep 1983.

[20] J. Paneque-Gálvez, M. McCall, B. Napoletano, S. Wich, and L. Koh, ”Small Drones for Community-Based Forest Monitoring: An Assessment of Their Feasibility and Potential in Tropical Areas,” *Forests*, vol. 5, pp. 1481–1507, jun 2014.

[21] L. P. Koh and S. A. Wich, ”Dawn of Drone Ecology: Low-Cost Autonomous Aerial Vehicles for Conservation,” *Tropical Conservation Science*, vol. 5, pp. 121–132, jun 2012.

[22] L. Wallace, A. Lucieer, C. Watson, and D. Turner, ”Development of a UAV-LiDAR System with Application to Forest Inventory,” *Remote Sensing*, vol. 4, pp. 1519–1543, may 2012.

[23] I. Colomina and P. Molina, ”Unmanned aerial systems for photogrammetry and remote sensing: A review,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 92, pp. 79–97, jun 2014.

[24] S. Candiago, F. Remondino, M. De Giglio, M. Dubbini, and M. Gattelli, ”Evaluating Multispectral Images and Vegetation Indices for Precision Farming Applications from UAV Images,” *Remote Sensing*, vol. 7, pp. 4026–4047, apr 2015.

[25] D. Hausamann, W. Zirnig, G. Schreier, and P. Strobl, ”Monitoring of gas pipelines - a civil UAV application,” *Aircraft Engineering and Aerospace Technology*, vol. 77, pp. 352–360, oct 2005.

[26] Y. Zhang, X. Yuan, Y. Fang, and S. Chen, ”UAV Low Altitude Photogrammetry for Power Line Inspection,” *ISPRS International Journal of Geo-Information*, vol. 6, p. 14, jan 2017.

[27] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, ”A UAV system for inspection of industrial facilities,” in *2013 IEEE Aerospace Conference*, pp. 1–8, IEEE, mar 2013.

[28] N. Metni and T. Hamel, ”A UAV for bridge inspection: Visual servoing control law with orientation limits,” *Automation in Construction*, vol. 17, pp. 3–10, nov 2007.

[29] C. Burke, P. R. McWhirter, J. Veitch-Michaelis, O. McAree, H. A. Pointon, S. Wich, and S. Longmore, ”Requirements and Limitations of Thermal Drones for Effective Search and Rescue in Marine and Coastal Areas,” *Drones*, vol. 3, p. 78, oct 2019.

[30] Her Majesty’s Inspectorate of Constabulary and Fire and Rescue Services, *Planes, drones and helicopters: An independent study of police air support*. No. November, Justice Inspectorates, 2017.

[31] P. G. Savage, ”Strapdown Inertial Navigation Integration Algorithm Design Part 1 : Attitude Algorithms,” *Journal of Guidance, Control and Dynamics*, vol. 21, no. 1, 1998.

[32] W. Ci and Y. Huang, ”A Robust Method for Ego-Motion Estimation in Urban Environment Using Stereo Camera.,” *Sensors (Basel, Switzerland)*, vol. 16, oct 2016.

[33] N. Kj, A. Sreejith, J. Mathew, M. Sarpotdar, A. Suresh, A. Prakash, M. Safonova, and J. Murthy, ”Noise modeling and analysis of an IMU-based attitude sensor: improvement of performance by filtering and sensor fusion,” p. 99126W, 2016.

[34] N. El-Sheimy, H. Hou, and X. Niu, ”Analysis and modeling of inertial sensors using allan variance,” *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 1, pp. 140–149, 2008.

[35] G. Araghi and R. J. Landry, ”Temperature compensation model of MEMS inertial sensors based on neural network,” in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 301–309, apr 2018.

[36] Z. Lin, Y. Xiong, H. Dai, and X. Xia, ”An Experimental Performance Evaluation of the Orientation Accuracy of Four Nine-Axis MEMS Motion

Sensors," in *2017 5th International Conference on Enterprise Systems (ES)*, pp. 185–189, 2017.

[37] J. Zhang, E. Edwan, J. Zhou, W. Chai, and O. Loffeld, "Performance investigation of barometer aided GPS/MEMS-IMU integration," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pp. 598–604, apr 2012.

[38] D. E. Bittner, J. A. Christian, and R. H. Bishop, "Fault Detection , Isolation , and Recovery Techniques for Large Clusters of Inertial Measurement Units," pp. 219–229.

[39] G. Ruzza, L. Guerriero, P. Revellino, and F. Guadagno, "Thermal Compensation of Low-Cost MEMS Accelerometers for Tilt Measurements," *Sensors*, vol. 18, no. 8, p. 2536, 2018.

[40] K. Schmid, P. Lutz, T. Tomic, E. Mair, and H. Hirschmüller, "Autonomous Vision-based Micro Air Vehicle for Indoor and Outdoor Navigation," *Journal of Field Robotics*, vol. 31, no. 4, 2014.

[41] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).* The MIT Press, 2005.

[42] Q. Quan, *Introduction to Multicopter Design and Control.* Singapore: Springer, Singapore, 2017.

[43] A. Johnson, *Process Dynamics Estimation and Control*, vol. 133. London: Peter Peregrinus Ltd, London, 1985.

[44] N. Correll, *Introduction to Autonomous Robots-Kinematics, Perception, Localization and Planning.* CreateSpace Independent Publishing Platform, 2015.

[45] A. G Quinchia, G. Falco, E. Falletti, F. Dovis, and C. Ferrer, ”A comparison between different error modeling of MEMS applied to GPS/INS integrated systems,” *Sensors (Basel, Switzerland)*, vol. 13, pp. 9549–9588, 2013.

[46] D. W. Allan, ”Statistics of atomic frequency standards,” *Proceedings of the IEEE*, vol. 54, pp. 221–230, feb 1966.

[47] P. G. Savage, ”Strapdown Inertial Navigation Integration Algorithm Design Part 2 : Velocity and Position Algorithms,” *Journal of guidance control and dynamics*, vol. 21, no. 2, pp. 208–221, 1998.

[48] J. Sola, ”Quaternion kinematics for the error-state Kalman filter,” *arXiv preprint arXiv:1711.02508*, 2017.

[49] M. Kassecker, B. T. Tomi, K. Schmid, P. Lutz, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, ”Research Platform for Indoor and Outdoor Urban Search and Rescue,” no. August, 2012.

[50] P. Crocoll, T. Caselitz, B. Hettich, M. Langer, and G. F. Trommer, ”Laser-aided Navigation with Loop Closure Capabilities for Micro Aerial Vehicles in Indoor and Urban Environments,” pp. 373–384.

[51] S. Huh, D. H. Shim, and J. Kim, ”Integrated Navigation System using Camera and Gimbaled Laser Scanner for Indoor and Outdoor Autonomous Flight of UAVs,” pp. 3158–3163, 2013.

[52] K. D. Sebesta and N. Boizot, ”A Real-Time Adaptive High-Gain EKF, Applied to a Quadcopter Inertial Navigation System,” *IEEE Transactions on Industrial Electronics*, vol. 61, pp. 495–503, jan 2014.

[53] M. B. Rhudy, T. Larrabee, H. Chao, Y. Gu, and M. Napolitano, ”UAV Attitude, Heading, and Wind Estimation Using GPS/INS and an Air Data System,” in *AIAA Guidance, Navigation, and Control (GNC) Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, aug 2013.

[54] V. Kučera, ”Rudolf E. Kalman: Life and Works,” *IFAC-PapersOnLine*, vol. 50, pp. 631–636, jul 2017.

[55] R. E. Kalman, ”A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 1960.

[56] K. Guo, Z. Qiu, C. Miao, A. H. Zaini, C.-L. Chen, W. Meng, and L. Xie, ”Ultra-Wideband-Based Localization for Quadcopter Navigation,” *Unmanned Systems*, vol. 04, no. 01, pp. 23–34, 2016.

[57] D. Cline, *Variational principles in classical mechanics.* Rochester: University of Rochester River Campus Librarie, 2017.

[58] P. Banelli, ”Non-Linear Transformations of Gaussians and Gaussian-Mixtures with implications on Estimation and Information Theory,” nov 2011.

[59] B. J. McLoughlin, H. A. Pointon, J. P. McLoughlin, A. Shaw, and F. A. Bezombes, ”Uncertainty characterisation of mobile robot localisation techniques using optical surveying grade instruments,” *Sensors (Switzerland)*, vol. 18, p. 2274, jul 2018.

[60] B. Barshan and H. Durrant-Whyte, ”Inertial navigation systems for mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 328–342, jun 1995.

[61] J. Engel, J. Sturm, and D. Cremers, ”Camera-Based Navigation of a Low-Cost Quadrocopter,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2815–2821, 2012.

[62] K. Frank, E. M. Diaz, P. Robertson, and F. J. F. S, ”Bayesian Recognition of Safety Relevant Motion Activities with Inertial Sensors and Barometer,”

[63] S. Pullen, J. Lee, and Ion, ”Guidance, Navigation, and Separation Assurance for Local-Area UAV Networks: Putting the Pieces Together,” *Proceedings of the Ion 2013 Pacific Pnt Meeting*, pp. 902–914, 2013.

[64] B. Osmanoglu, T. H. Dixon, and S. Wdowinski, ”Three-Dimensional Phase Unwrapping for Satellite Radar Interferometry, I: DEM Generation,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 1059–1075, feb 2014.

[65] Institute of Electrical and Electronics Engineers., *15th International Conference on Information Fusion : 9-12 July 2012, Singapore.* IEEE, 2012.

[66] The Office of the General Counsel, ”The Air Navigation Order 2016 and Regulations,” 2016.

[67] O. Policy and S. Series, ”Unmanned Aerial Vehicles in Humanitarian Response OCCASIONAL,” no. June, 2014.

[68] ”Pozyx - centimeter positioning for arduino.”

[69] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, ”Ultra wideband indoor

positioning technologies: Analysis and recent advances," *Sensors (Switzerland)*, vol. 16, p. 707, may 2016.

[70] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, vol. 43, pp. 55–81, jan 2015.

[71] M. Marron, J. Garcia, M. Sotelo, E. Lopez, and M. Mazo, "Fusing odometric and vision data with an EKF to estimate the absolute position of an autonomous mobile robot," in *EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.03TH8696)*, vol. 1, pp. 591–596, IEEE.

[72] Z. Jia, A. Balasuriya, and S. Challa, "Autonomous Vehicles Navigation with Visual Target Tracking: Technical Approaches," *Algorithms*, vol. 1, pp. 153–182, dec 2008.

[73] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, pp. 652–659, IEEE.

[74] M. Alatise, G. Hancke, M. B. Alatise, and G. P. Hancke, "Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter," *Sensors*, vol. 17, p. 2164, sep 2017.

[75] I. Wieser, A. V. Ruiz, M. Frassl, M. Angermann, J. Mueller, and M. Lichtenstern, "Autonomous robotic SLAM-based indoor navigation for high resolution sampling with complete coverage," in *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 945–951, 2014.

[76] J. Blankenbach, A. Norrdine, and H. Hellmers, ”A robust and precise 3D indoor positioning system for harsh environments,” in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, IEEE, nov 2012.

[77] H. Hellmers, A. Norrdine, J. Blankenbach, and A. Eichhorn, ”An IMU/magnetometer-based Indoor positioning system using Kalman filtering,” in *International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–9, IEEE, oct 2013.

[78] Y. XU, G. TIAN, and X. CHEN, ”Performance enhancement for INS/UWB integrated indoor tracking using distributed iterated extended Kalman filter,” in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, pp. 1–5, IEEE, mar 2018.

[79] J. Liu, J. Pu, L. Sun, and Z. He, ”An Approach to Robust INS/UWB Integrated Positioning for Autonomous Indoor Mobile Robots,” *Sensors*, vol. 19, no. 4, p. 950, 2019.

[80] T. Gigl, G. J. Janssen, V. Dizdarevic, K. Witrisal, and Z. Irahhauten, ”Analysis of a UWB Indoor Positioning System Based on Received Signal Strength,” *2007 4th Workshop on Positioning, Navigation and Communication*, vol. 2007, no. 1, pp. 97–101, 2007.

[81] J. Liu, J. Pu, L. Sun, and Z. He, ”An Approach to Robust INS/UWB Integrated Positioning for Autonomous Indoor Mobile Robots,” *Sensors*, vol. 19, p. 950, feb 2019.

[82] S. Zhang, R. Han, W. Huang, S. Wang, and Q. Hao, ”Linear Bayesian Filter Based Low-Cost UWB Systems for Indoor Mobile Robot Localization,” in *2018 IEEE SENSORS*, pp. 1–4, IEEE, oct 2018.

[83] M. J. Segura, V. A. Mut, and H. D. Patino, ”Mobile robot self-localization system using IR-UWB sensor in indoor environments,” in *2009 IEEE International Workshop on Robotic and Sensors Environments*, pp. 29–34, IEEE, nov 2009.

[84] J. Lima and P. Costa, ”Ultra-Wideband Time of Flight Based Localization System and Odometry Fusion for a Scanning 3 DoF Magnetic Field Autonomous Robot,” pp. 879–890, Springer, Cham, nov 2018.

[85] X. Li, Y. Wang, and K. Khoshelham, ”UWB/PDR Tightly Coupled Navigation with Robust Extended Kalman Filter for NLOS Environments,” *Mobile Information Systems*, vol. 2018, pp. 1–14, dec 2018.

[86] X. Li, Y. Wang, and K. Khoshelham, ”UWB/PDR Tightly Coupled Navigation with Robust Extended Kalman Filter for NLOS Environments,” *Mobile Information Systems*, vol. 2018, pp. 1–14, dec 2018.

[87] M. Saha, B. Goswami, and R. Ghosh, ”Two novel costs for determining the tuning parameters of the Kalman Filter,” *Advances in Control and Optimization of Dynamic Systems*, 2011.

[88] N. Kaur and A. Kaur, ”A Review on Tuning of Extended Kalman Filter using Optimization Techniques for State Estimation,” *International Journal of Computer Applications*, vol. 145, no. 15, pp. 1–5, 2016.

[89] J. González, J. Blanco, C. Galindo, A. Ortiz-de Galisteo, J. Fernández-Madrigal, F. Moreno, and J. Martínez, ”Mobile robot localization based on Ultra-Wide-Band ranging: A particle filter approach,” *Robotics and Autonomous Systems*, vol. 57, pp. 496–507, may 2009.

[90] A. Masiero, F. Fissore, A. Vettore, A. Masiero, F. Fissore, and A. Vettore, ”A Low Cost UWB Based Solution for Direct Georeferencing UAV Photogrammetry,” *Remote Sensing*, vol. 9, p. 414, apr 2017.

[91] P. Dabove, V. Di Pietra, M. Piras, A. A. Jabbar, and S. A. Kazim, ”Indoor positioning using Ultra-wide band (UWB) technologies: Positioning accuracies and sensors' performances,” in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 175–184, IEEE, apr 2018.

[92] J. Yang, T. Cheng, J. Teizer, P. Vela, and Z. Shi, ”A performance evaluation of vision and radio frequency tracking methods for interacting workforce,” *Advanced Engineering Informatics*, vol. 25, pp. 736–747, oct 2011.

[93] C. Roberts and P. Boorer, ”Kinematic positioning using a robotic total station as applied to small-scale UAVs,” *Journal of Spatial Science*, vol. 61, pp. 29–45, jan 2016.

[94] J. Li, Y. Bi, K. Li, K. Wang, F. Lin, and B. M. Chen, ”Accurate 3D Localization for MAV Swarms by UWB and IMU Fusion,” jul 2018.

[95] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, ”A Study of Vicon System Positioning Performance.,” *Sensors (Basel, Switzerland)*, vol. 17, jul 2017.

[96] Trimble, *Trimble S7 Total Station: User Guide.* No. A, Westminster: Trimble Navigation Limited, revision a ed., 2015.

[97] Pozyx, ”Creator - Documentation | Pozyx.”

[98] G. Aiello and G. Rogerson, ”Ultra-Wideband Wireless Systems,” *IEEE Microwave Magazine*, vol. 4, pp. 36–47, jun 2003.

[99] H. Liu, H. Darabi, P. Banerjee, and J. Liu, ”Survey of wireless indoor positioning techniques and systems,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.

[100] S. Gezici, Zhi Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, ”Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks,” *IEEE Signal Processing Magazine*, vol. 22, pp. 70–84, jul 2005.

[101] R. Fontana, ”Recent System Applications of Short-Pulse Ultra-Wideband (UWB) Technology,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, pp. 2087–2104, sep 2004.

[102] R. Bharadwaj, C. Parini, and A. Alomainy, ”Experimental Investigation of 3-D Human Body Localization Using Wearable Ultra-Wideband Antennas,” *IEEE Transactions on Antennas and Propagation*, vol. 63, pp. 5035–5044, nov 2015.

[103] M. Ridolfi, S. Vandermeeren, J. Defraye, H. Steendam, J. Gerlo, D. De Clercq, J. Hoebeke, and E. De Poorter, ”Experimental Evaluation of UWB Indoor Positioning for Sport Postures,” *Sensors*, vol. 18, no. 1, 2018.

[104] A. R. Jimenez Ruiz and F. Seco Granja, ”Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, pp. 2106–2117, aug 2017.

[105] Sweio, "UWB Technology for Precise Indoor Tracking | Sewio RTLS."

[106] Locatify, "Precise Indoor Positioning is finally here thanks to Ultra-wideband RTLS."

[107] L. Yao, Y.-W. A. Wu, L. Yao, and Z. Z. Liao, "An integrated IMU and UWB sensor based indoor positioning system," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, IEEE, sep 2017.

[108] T. Conceição, F. N. dos Santos, P. Costa, and A. P. Moreira, "Robot localization system in a hard outdoor environment," in *Advances in Intelligent Systems and Computing*, vol. 693, pp. 215–227, Springer, Cham, nov 2018.

[109] A. I. Eliazar and R. Parr, "Learning probabilistic motion models for mobile robots," in *Twenty-first international conference on Machine learning - ICML '04*, (New York, New York, USA), p. 32, ACM Press, 2004.

[110] C. Hajiyev and H. E. Soken, "Adaptive kalman filter with the filter gain correction applied to UAV flight dynamics," in *2009 17th Mediterranean Conference on Control and Automation*, pp. 892–897, IEEE, jun 2009.

[111] C. Hajiyev and H. E. Soken, "Robust Adaptive Kalman Filter for estimation of UAV dynamics in the presence of sensor/actuator faults," *Aerospace Science and Technology*, vol. 28, pp. 376–383, jul 2013.

[112] C. Hu, W. Chen, Y. Chen, and D. Liu, "Adaptive Kalman Filtering for Vehicle Navigation," *Journal of Global Positioning Systems*, vol. 2, pp. 42–47, jun 2003.

[113] A. L. da Silva and J. J. da Cruz, ”Fuzzy adaptive extended Kalman filter for UAV INS/GPS data fusion,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 38, pp. 1671–1688, aug 2016.

[114] K. D. Sebesta and N. Boizot, ”A Real-Time Adaptive High-Gain EKF, Applied to a Quadcopter Inertial Navigation System,” *IEEE Transactions on Industrial Electronics*, vol. 61, pp. 495–503, jan 2014.

[115] B. Cui, X. Chen, and X. Tang, ”Improved Cubature Kalman Filter for GNSS/INS Based on Transformation of Posterior Sigma-Points Error,” *IEEE Transactions on Signal Processing*, vol. 65, pp. 2975–2987, jun 2017.

[116] J. Liu, B.-g. Cai, and J. Wang, ”Cooperative Localization of Connected Vehicles: Integrating GNSS With DSRC Using a Robust Cubature Kalman Filter,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 2111–2125, aug 2017.

[117] I. Arasaratnam and S. Haykin, ”Cubature Kalman Filters,” *IEEE Transactions on Automatic Control*, vol. 54, pp. 1254–1269, jun 2009.

[118] Z. Gao, D. Mu, Y. Zhong, C. Gu, and C. Ren, ”Adaptively Random Weighted Cubature Kalman Filter for Nonlinear Systems,” *Mathematical Problems in Engineering*, vol. 2019, pp. 1–13, jan 2019.

[119] X.-C. Zhang and C.-J. Guo, ”Cubature Kalman filters: Derivation and extension,” *Chinese Physics B*, vol. 22, p. 128401, dec 2013.

[120] A. Ledergerber and R. D’Andrea, ”Ultra-wideband range measurement model with Gaussian processes,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1929–1934, IEEE, aug 2017.

[121] A. Ledergerber and R. D'andrea, "Calibrating Away Inaccuracies in Ultra Wideband Range Measurements: A Maximum Likelihood Approach," *IEEE Access*, vol. 6, pp. 78719–78730, 2018.

[122] H. Wymeersch, S. Marano, W. M. Gifford, and M. Z. Win, "A Machine Learning Approach to Ranging Error Mitigation for UWB Localization," *IEEE Transactions on Communications*, vol. 60, pp. 1719–1728, jun 2012.

[123] S. Boslaugh and P. A. Watters, *Statistics in a Nutshell, 2nd Edition*. O'Reilly Media, Inc, 2012.

[124] S. Greenland, S. J. Senn, K. J. Rothman, J. B. Carlin, C. Poole, S. N. Goodman, and D. G. Altman, "Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations.," *European journal of epidemiology*, vol. 31, no. 4, pp. 337–50, 2016.

[125] Civil Aviation Authority, "CAP 722," tech. rep., Civil Aviation Authority, 2015.

[126] S. Akahori, Y. Higashi, and A. Masuda, "Position estimation system with UWB, IMU and a distance sensor for quad-rotors," in *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 1992–1996, IEEE, nov 2017.

[127] F. Vanegas and F. Gonzalez, "Enabling UAV navigation with sensor and environmental uncertainty in cluttered and GPS-denied environments," *Sensors (Switzerland)*, vol. 16, no. 5, 2016.

[128] J. W. Langelaan, *State Estimation for Autonomous Flight in Cluttered Environments | Aerospace Robotics Lab*. PhD thesis, Stanford University, 2006.

[129] H. A. G. Pointon, B. J. McLoughlin, C. Matthews, and F. A. Bezombes, "Towards a Model Based Sensor Measurement Variance Input for Extended Kalman Filter State Estimation," *Drones*, vol. 3, no. 1, 2019.

[130] DJI, "DJI Mavic Pro - Specs, Tutorials & Guides - DJI."

7.2 Appendix

```python
#!/usr/bin/python

# Written by Harry Pointon 28/08/18

# This is the second form of the code after prototyping.

# ROS node which takes in the Trimble S7 RTS data from
    continuous topographic measurements.


# The data is output using a TSC3 controller using the "data
    output" function found on the second page of the instrument
# Section. Using serial connection the port on the TSC3 is set
    as "Controller port 1".
# Data is output as "pseudo NMEA" the format can be found on
    page 475-476 (sometimes 492)
# of the Trimble access general survey guide. Link below.
# https://www.geosoft.ee/sites/default/files/general_survey.pdf
# Northing, Easting and Altitude details may be found in
    entries 3,5 and 10 respectively
# This data is published by the node in raw format as a string
    and recoded as a pose stamped form intended for MavROS.

import serial, time
import rospy
from geometry_msgs.msg import PoseStamped
from std_msgs.msg import String

class RTS_Class(object):


```

```
    def __init__(self, Serial_Port, Baud):
        # Initialise the node
        rospy.init_node('RTS_Talker', anonymous=True)

        # Initialise the publishers
        self.String_pub = rospy.Publisher('/RTS', String, queue_size=10)
        self.Pose_pub = rospy.Publisher('/RTS_Pose', PoseStamped, queue_size=10)

        # Initialise the serial link at 115200 baudrate
        ser = serial.Serial(Serial_Port)
        ser.baudrate = Baud

        # Place to store the raw and decoded NMEA data
        self.RTS_String = None
        self.RTS_Northing = None
        self.RTS_Easting = None
        self.RTS_Altitude = None
        self.Serial_data = None
        self.x = None
        self.y = None
        self.Z = None

        rate = rospy.Rate(5)  # 10hz

        # Run the node while ROS is not shutdown.
        while not rospy.is_shutdown():
            # Read serial connection
            self.Serial_data = ser.readline()
            # Send data from serial to publisher
            self.string_publisher()
            # Decode the data for the pose
```

```
            self.seperate_RTS_String()
            # Send the decoded data to the publisher
            self.poseStamped_publisher()

    # Publish the raw serial data.
    # Removed the sequence thing from the Ben's custom msg.
    def string_publisher(self):
        nmea = self.Serial_data
        rospy.loginfo(nmea)
        self.String_pub.publish(nmea)

    # Publish the decoded pose data.
    def poseStamped_publisher(self):
        pose = PoseStamped()
        pose.header.stamp = rospy.get_rostime()
        pose.header.frame_id = "fcu"
        pose.pose.position.x = self.x
        pose.pose.position.y = self.y
        pose.pose.position.z = self.z

        # Leave this section blank, another node will add the
    orientation and republish for MavROS if needed
        pose.pose.orientation.x = 0
        pose.pose.orientation.y = 0
        pose.pose.orientation.z = 0
        pose.pose.orientation.w = 0

        self.Pose_pub.publish(pose)

    # Decodes the RTS serial data into x,y,z data in float form
    .
    def seperate_RTS_String(self):
        # Break full serial string up into required information
```

```
        split_String_List = self.Serial_data.split(',')
        Northing = split_String_List[2]
        Easting = split_String_List[4]
        Altitude = split_String_List[9]
        # Check if data is present
        rospy.loginfo(Easting)
        rospy.loginfo(Northing)
        rospy.loginfo(Altitude)
        if len(Easting) < 1:
            Easting = 0
        if len(Northing) < 1:
            Northing = 0
        if len(Altitude) < 1:
            Altitude = 0
        # Convert from string to float for publishing
        self.x = float(Easting)
        self.y = float(Northing)
        self.z = float(Altitude)

if __name__ == '__main__':
    Serial_Port = '/dev/ttyUSB0'
    Baud = 115200
    try:
        RTS_Class(Serial_Port, Baud)
    except rospy.ROSInterruptException:
        pass
```

Listing 7.1: RTS ROS node in Python

```
#!/usr/bin/python

# Written by Harry Pointon 28/08/18
import serial, time
import rospy
```

```
from std_msgs.msg import Int16
from trimble_rts.msg import Encoder

ser = serial.Serial('/dev/ttyACM0')
ser.baudrate = 115200

class Wheel_Encoder_Node(object):

  def __init__(self):
    rospy.init_node('Wheel_Encoder_Node')

    self.SerialDataIn = None
    self.SerialData = None
    self.LeftF_wheel = None
    self.RightF_wheel = None
    self.LeftB_wheel = None
    self.RightB_wheel = None

    # # Define Left publisher
    # self.lfwheel_pub = rospy.Publisher('/LFwheel', Encoder, queue_size=50)
    # self.lbwheel_pub = rospy.Publisher('/LBwheel', Encoder, queue_size=50)

    # # Encoder Right publisher
    # self.rfwheel_pub = rospy.Publisher('/RFwheel', Encoder, queue_size=50)
    self.encoderPub = rospy.Publisher('/encoders', Encoder, queue_size=50)

    while not rospy.is_shutdown():
      self.SerialDataIn = ser.readline()
      self.DataHandling()
```

```python


  def DataHandling(self):
    self.SerialData    =  self.SerialDataIn.split(',')
    self.LeftB_wheel   =  int(self.SerialData[0])
    self.LeftF_wheel   =  int(self.SerialData[1])
    self.RightB_wheel  =  int(self.SerialData[2])
    self.RightF_wheel  =  int (self.SerialData[3])
    self.encoderPub()
    #self.RFencoderPub()
    #self.LBencoderPub()
    #self.RBencoderPub()

  def WheelPub(self):
    enc = Encoder()
    enc.header.stamp = rospy.get_rostime()
    enc.lf = LeftF_wheel
    enc.rf = RightF_wheel
    enc.rb = RightB_wheel
    enc.lb = LeftB_wheel
    self.encoderPub.publish(enc)

if __name__ == '__main__':
  WEN = Wheel_Encoder_Node()
```

Listing 7.2: Wheel encoder ROS node in Python

Generation of the EKF Functions for Simulation:

Definition of Variables:

```
syms q1 q2 q3 q4 'real'
syms x y z 'real'
syms xPrime yPrime zPrime 'real'

syms DeltaX DeltaY DeltaZ 'real'
syms DeltaXBias DeltaYBias DeltaZBias 'real'

syms Deltai Deltaj Deltak 'real'
syms DeltaiBias DeltajBias DeltakBias 'real'
syms g deltaTime 'real'
syms f Gx Gu StateTransition
```

Define Equations:

Positon and Velocities are NED referenced. Orientation is a quarternion rotation from navigation frame to body frame.

```
Quat = [q1; q2; q3; q4]            % Orientation
```

Quat =

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}$$

```
Pos = [x; y; z]                    % Position Components NED
```

Pos =

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

```
Vel = [xPrime; yPrime; zPrime]     % Velocity Components NED
```

Vel =

$$\begin{pmatrix} \text{xPrime} \\ \text{yPrime} \\ \text{zPrime} \end{pmatrix}$$

```
% TBN = [q1^2+q2^2-q3^2-q4^2, 2*(dot(q2, q3)-dot(q1, q4)), 2*(dot(q2, q4)+dot(q1, q3)); ...
%        2*(dot(q2, q3)+dot(q1, q4)), q1^2+q2^2+q3^2-q4^2, 2*(dot(q3, q4)-dot(q1, q2)); ...
```

```
%        2*(dot(q2, q4)-dot(q1, q3)), 2*(dot(q3, q4)+dot(q1, q2)), q1^2-q2^2-q3^2+q4^2]

TBN = [q1^2 + q2^2 - q3^2 - q4^2, 2*(q2*q3 - q1*q4), 2*(q2*q4 + q1*q3); ...
       2*(q2*q3 + q1*q4), q1^2 - q2^2 + q3^2 - q4^2, 2*(q3*q4 - q2*q2); ...
       2*(q2*q4-q1*q3), 2*(q3*q4 + q1*q2), q1^2 - q2^2 - q3^2 + q4^2];



G = [0; 0; g]
```

G =

$$\begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix}$$

```
X = [Quat; Vel; Pos]               % State
```

X =

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ \mathrm{xPrime} \\ \mathrm{yPrime} \\ \mathrm{zPrime} \\ x \\ y \\ z \end{pmatrix}$$

State Transition Equations:

Velocities:

```
DeltaVelMeasured = [DeltaX; DeltaY; DeltaZ]         % X Y Z not NED
```

DeltaVelMeasured =

$$\begin{pmatrix} \mathrm{DeltaX} \\ \mathrm{DeltaY} \\ \mathrm{DeltaZ} \end{pmatrix}$$

```
DeltaVelBias = [DeltaXBias; DeltaYBias; DeltaZBias] %
```

DeltaVelBias =

$$\begin{pmatrix} \text{DeltaXBias} \\ \text{DeltaYBias} \\ \text{DeltaZBias} \end{pmatrix}$$

```
TrueDeltaVel = DeltaVelMeasured - DeltaVelBias      % X Y Z not NED
```

TrueDeltaVel =

$$\begin{pmatrix} \text{DeltaX} - \text{DeltaXBias} \\ \text{DeltaY} - \text{DeltaYBias} \\ \text{DeltaZ} - \text{DeltaZBias} \end{pmatrix}$$

Orientations:

```
DeltaAngMeasured = [Deltai; Deltaj; Deltak]
```

DeltaAngMeasured =

$$\begin{pmatrix} \text{Deltai} \\ \text{Deltaj} \\ \text{Deltak} \end{pmatrix}$$

```
DeltaAngBias = [DeltaiBias; DeltajBias; DeltakBias]
```

DeltaAngBias =

$$\begin{pmatrix} \text{DeltaiBias} \\ \text{DeltajBias} \\ \text{DeltakBias} \end{pmatrix}$$

```
TrueDeltaAng = (DeltaAngMeasured - DeltaAngBias) * deltaTime
```

TrueDeltaAng =

$$\begin{pmatrix} \text{deltaTime}\ (\text{Deltai} - \text{DeltaiBias}) \\ \text{deltaTime}\ (\text{Deltaj} - \text{DeltajBias}) \\ \text{deltaTime}\ (\text{Deltak} - \text{DeltakBias}) \end{pmatrix}$$

```

DeltaQuat = [1; TrueDeltaAng/2]
```

DeltaQuat =

$$\begin{pmatrix} 1 \\ \frac{\text{deltaTime}\ (\text{Deltai} - \text{DeltaiBias})}{2} \\ \frac{\text{deltaTime}\ (\text{Deltaj} - \text{DeltajBias})}{2} \\ \frac{\text{deltaTime}\ (\text{Deltak} - \text{DeltakBias})}{2} \end{pmatrix}$$

Update Orientation:

```
NextQuatState = quatmultiply(Quat', DeltaQuat')
```

NextQuatState =

$$\left(q_1 - \frac{\text{deltaTime}\, q_2\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_3\, (\text{Deltaj} - \text{DeltajBias})}{2} - \frac{\text{deltaTime}\, q_4\, (\text{Deltak} - \text{DeltakBias})}{2} \quad q_2\right.$$

```
NextQuatState = NextQuatState'
```

NextQuatState =

$$\begin{pmatrix} q_1 - \frac{\text{deltaTime}\, q_2\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_3\, (\text{Deltaj} - \text{DeltajBias})}{2} - \frac{\text{deltaTime}\, q_4\, (\text{Deltak} - \text{DeltakBias})}{2} \\ q_2 + \frac{\text{deltaTime}\, q_1\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_4\, (\text{Deltaj} - \text{DeltajBias})}{2} + \frac{\text{deltaTime}\, q_3\, (\text{Deltak} - \text{DeltakBias})}{2} \\ q_3 + \frac{\text{deltaTime}\, q_1\, (\text{Deltaj} - \text{DeltajBias})}{2} + \frac{\text{deltaTime}\, q_4\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_2\, (\text{Deltak} - \text{DeltakBias})}{2} \\ q_4 - \frac{\text{deltaTime}\, q_3\, (\text{Deltai} - \text{DeltaiBias})}{2} + \frac{\text{deltaTime}\, q_2\, (\text{Deltaj} - \text{DeltajBias})}{2} + \frac{\text{deltaTime}\, q_1\, (\text{Deltak} - \text{DeltakBias})}{2} \end{pmatrix}$$

Update Velocities:

True change in velocity is rotated from body frame to NED via TBN

```
NextVelState = Vel + (TBN *TrueDeltaVel) + (G * deltaTime) % Now in NED
```

NextVelState =

$$\left(\begin{matrix} \text{xPrime} + (\text{DeltaX} - \text{DeltaXBias})\,(q_1{}^2 + q_2{}^2 - q_3{}^2 - q_4{}^2) - (\text{DeltaY} - \text{DeltaYBias})\,(2\,q_1\,q_4 - 2\,q_2\,q_3) + (\text{DeltaZ} \\ \text{yPrime} + (\text{DeltaY} - \text{DeltaYBias})\,(q_1{}^2 - q_2{}^2 + q_3{}^2 - q_4{}^2) + (\text{DeltaZ} - \text{DeltaZBias})\,(2\,q_3\,q_4 - 2\,q_2{}^2) + (\text{DeltaX} - \\ \text{zPrime} + (\text{DeltaZ} - \text{DeltaZBias})\,(q_1{}^2 - q_2{}^2 - q_3{}^2 + q_4{}^2) + \text{deltaTime}\, g - (\text{DeltaX} - \text{DeltaXBias})\,(2\,q_1\,q_3 - 2\,q_2\,q_4) + (\text{I} \end{matrix}\right.$$

Update Position:

```
NextPosState = Pos + Vel * deltaTime                % Always was in NED
```

NextPosState =

$$\begin{pmatrix} x + \text{deltaTime xPrime} \\ y + \text{deltaTime yPrime} \\ z + \text{deltaTime zPrime} \end{pmatrix}$$

State Updated

```
StateTransition = [NextQuatState; NextVelState; NextPosState]
```

StateTransition =

$$\left(\begin{array}{c} q_1 - \frac{\text{deltaTime}\, q_2\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_3\, (\text{Deltaj} - \text{DeltajBias})}{2} - \frac{\text{deltaTime}\, q_4\, (\text{De}}{} \\ q_2 + \frac{\text{deltaTime}\, q_1\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_4\, (\text{Deltaj} - \text{DeltajBias})}{2} + \frac{\text{deltaTime}\, q_3\, (\text{De}}{} \\ q_3 + \frac{\text{deltaTime}\, q_1\, (\text{Deltaj} - \text{DeltajBias})}{2} + \frac{\text{deltaTime}\, q_4\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_2\, (\text{De}}{} \\ q_4 - \frac{\text{deltaTime}\, q_3\, (\text{Deltai} - \text{DeltaiBias})}{2} + \frac{\text{deltaTime}\, q_2\, (\text{Deltaj} - \text{DeltajBias})}{2} + \frac{\text{deltaTime}\, q_1\, (\text{De}}{} \\ \text{xPrime} + (\text{DeltaX} - \text{DeltaXBias})\, (q_1^2 + q_2^2 - q_3^2 - q_4^2) - (\text{DeltaY} - \text{DeltaYBias})\, (2\, q_1\, q_4 - 2\, q_2\, q_3) + (\text{DeltaZ} \\ \text{yPrime} + (\text{DeltaY} - \text{DeltaYBias})\, (q_1^2 - q_2^2 + q_3^2 - q_4^2) + (\text{DeltaZ} - \text{DeltaZBias})\, (2\, q_3\, q_4 - 2\, q_2^2) + (\text{DeltaX} - \\ \text{zPrime} + (\text{DeltaZ} - \text{DeltaZBias})\, (q_1^2 - q_2^2 - q_3^2 + q_4^2) + \text{deltaTime}\, g - (\text{DeltaX} - \text{DeltaXBias})\, (2\, q_1\, q_3 - 2\, q_2\, q_4) + (\\ x + \text{deltaTime xPrime} \\ y + \text{deltaTime yPrime} \\ z + \text{deltaTime zPrime} \end{array} \right.$$

```
U = [DeltaAngMeasured; DeltaVelMeasured]
```

U =

$$\begin{pmatrix} \text{Deltai} \\ \text{Deltaj} \\ \text{Deltak} \\ \text{DeltaX} \\ \text{DeltaY} \\ \text{DeltaZ} \end{pmatrix}$$

```
f(U) = StateTransition
```

f(Deltai, Deltaj, Deltak, DeltaX, DeltaY, DeltaZ) =

$$\left(\begin{array}{c} q_1 - \frac{\text{deltaTime}\, q_2\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_3\, (\text{Deltaj} - \text{DeltajBias})}{2} - \frac{\text{deltaTime}\, q_4\, (\text{De}}{} \\ q_2 + \frac{\text{deltaTime}\, q_1\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_4\, (\text{Deltaj} - \text{DeltajBias})}{2} + \frac{\text{deltaTime}\, q_3\, (\text{De}}{} \\ q_3 + \frac{\text{deltaTime}\, q_1\, (\text{Deltaj} - \text{DeltajBias})}{2} + \frac{\text{deltaTime}\, q_4\, (\text{Deltai} - \text{DeltaiBias})}{2} - \frac{\text{deltaTime}\, q_2\, (\text{De}}{} \\ q_4 - \frac{\text{deltaTime}\, q_3\, (\text{Deltai} - \text{DeltaiBias})}{2} + \frac{\text{deltaTime}\, q_2\, (\text{Deltaj} - \text{DeltajBias})}{2} + \frac{\text{deltaTime}\, q_1\, (\text{De}}{} \\ \text{xPrime} + (\text{DeltaX} - \text{DeltaXBias})\, (q_1^2 + q_2^2 - q_3^2 - q_4^2) - (\text{DeltaY} - \text{DeltaYBias})\, (2\, q_1\, q_4 - 2\, q_2\, q_3) + (\text{DeltaZ} \\ \text{yPrime} + (\text{DeltaY} - \text{DeltaYBias})\, (q_1^2 - q_2^2 + q_3^2 - q_4^2) + (\text{DeltaZ} - \text{DeltaZBias})\, (2\, q_3\, q_4 - 2\, q_2^2) + (\text{DeltaX} - \\ \text{zPrime} + (\text{DeltaZ} - \text{DeltaZBias})\, (q_1^2 - q_2^2 - q_3^2 + q_4^2) + \text{deltaTime}\, g - (\text{DeltaX} - \text{DeltaXBias})\, (2\, q_1\, q_3 - 2\, q_2\, q_4) + (\\ x + \text{deltaTime xPrime} \\ y + \text{deltaTime yPrime} \\ z + \text{deltaTime zPrime} \end{array} \right.$$

Static Process Model

```
syms MagN MagE MagD MagX MagY MagZ 'real'
syms VwindN VwindE 'real'
SPM = [DeltaAngBias; DeltaVelBias; MagN; MagE; MagD; MagX; MagY; MagZ; VwindN; VwindE]
```

SPM =

$$\begin{pmatrix} \mathrm{DeltaiBias} \\ \mathrm{DeltajBias} \\ \mathrm{DeltakBias} \\ \mathrm{DeltaXBias} \\ \mathrm{DeltaYBias} \\ \mathrm{DeltaZBias} \\ \mathrm{MagN} \\ \mathrm{MagE} \\ \mathrm{MagD} \\ \mathrm{MagX} \\ \mathrm{MagY} \\ \mathrm{MagZ} \\ \mathrm{VwindN} \\ \mathrm{VwindE} \end{pmatrix}$$

Generate Jacobians:

```
Gx = jacobian(f,X)
```

Gx(Deltai, Deltaj, Deltak, DeltaX, DeltaY, DeltaZ) =

$$\left(\begin{array}{ccccccc}
1 & -\sigma_4 & -\sigma_3 & -\sigma_2 & 0 & 0 & 0 \\
\sigma_4 & 1 & \sigma_2 & -\sigma_3 & 0 & 0 & 0 \\
\sigma_3 & -\sigma_2 & 1 & \sigma_4 & 0 & 0 & 0 \\
\sigma_2 & \sigma_3 & -\sigma_4 & 1 & 0 & 0 & 0 \\
\sigma_{10} - \sigma_8 + \sigma_7 & \sigma_1 & \sigma_{11} & \sigma_9 - \sigma_5 - \sigma_6 & 1 & 0 & 0 \\
\sigma_6 + \sigma_5 & \sigma_{13} - \sigma_{12} - 4\,q_2\,(\mathrm{DeltaZ} - \mathrm{DeltaZBias}) & \sigma_1 & \sigma_{10} - \sigma_8 + \sigma_7 & 0 & 1 & 0 \\
\sigma_{11} & \sigma_6 + \sigma_5 - \sigma_9 & \sigma_8 - \sigma_{10} - \sigma_7 & \sigma_1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & \mathrm{deltaTime} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathrm{deltaTime} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathrm{deltaTime}
\end{array}\right.$$

where

$$\sigma_1 = 2\,q_2\,(\mathrm{DeltaX} - \mathrm{DeltaXBias}) + 2\,q_3\,(\mathrm{DeltaY} - \mathrm{DeltaYBias}) + 2\,q_4\,(\mathrm{DeltaZ} - \mathrm{DeltaZBias})$$

$$\sigma_2 = \frac{\mathrm{deltaTime}\,(\mathrm{Deltak} - \mathrm{DeltakBias})}{2}$$

$$\sigma_3 = \frac{\mathrm{deltaTime}\,(\mathrm{Deltaj} - \mathrm{DeltajBias})}{2}$$

$$\sigma_4 = \frac{\mathrm{deltaTime}\,(\mathrm{Deltai} - \mathrm{DeltaiBias})}{2}$$

$$\sigma_5 = 2\,q_4\,(\mathrm{DeltaX} - \mathrm{DeltaXBias})$$

$$\sigma_6 = 2\,q_1\,(\mathrm{DeltaY} - \mathrm{DeltaYBias})$$

$$\sigma_7 = 2\,q_3\,(\mathrm{DeltaZ} - \mathrm{DeltaZBias})$$

$$\sigma_8 = 2\,q_4\,(\mathrm{DeltaY} - \mathrm{DeltaYBias})$$

$$\sigma_9 = 2\,q_2\,(\mathrm{DeltaZ} - \mathrm{DeltaZBias})$$

$$\sigma_{10} = 2\,q_1\,(\mathrm{DeltaX} - \mathrm{DeltaXBias})$$

$$\sigma_{11} = \sigma_{12} - \sigma_{13} + 2\,q_1\,(\mathrm{DeltaZ} - \mathrm{DeltaZBias})$$

$$\sigma_{12} = 2\,q_2\,(\mathrm{DeltaY} - \mathrm{DeltaYBias})$$

$$\sigma_{13} = 2\,q_3\,(\mathrm{DeltaX} - \mathrm{DeltaXBias})$$

```
Gu = jacobian(f,U)
```

```
Gu(Deltai, Deltaj, Deltak, DeltaX, DeltaY, DeltaZ) =
```

$$
\begin{pmatrix}
\sigma_3 & \sigma_2 & \sigma_1 & 0 & 0 & 0 \\
\frac{\mathrm{deltaTime}\, q_1}{2} & \sigma_1 & \frac{\mathrm{deltaTime}\, q_3}{2} & 0 & 0 & 0 \\
\frac{\mathrm{deltaTime}\, q_4}{2} & \frac{\mathrm{deltaTime}\, q_1}{2} & \sigma_3 & 0 & 0 & 0 \\
\sigma_2 & \frac{\mathrm{deltaTime}\, q_2}{2} & \frac{\mathrm{deltaTime}\, q_1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & {q_1}^2 + {q_2}^2 - {q_3}^2 - {q_4}^2 & 2\, q_2\, q_3 - 2\, q_1\, q_4 & 2\, q_1\, q_3 + 2\, q_2\, q_4 \\
0 & 0 & 0 & 2\, q_1\, q_4 + 2\, q_2\, q_3 & {q_1}^2 - {q_2}^2 + {q_3}^2 - {q_4}^2 & 2\, q_3\, q_4 - 2\, {q_2}^2 \\
0 & 0 & 0 & 2\, q_2\, q_4 - 2\, q_1\, q_3 & 2\, q_1\, q_2 + 2\, q_3\, q_4 & {q_1}^2 - {q_2}^2 - {q_3}^2 + {q_4}^2 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

where

$$\sigma_1 = -\frac{\mathrm{deltaTime}\, q_4}{2}$$

$$\sigma_2 = -\frac{\mathrm{deltaTime}\, q_3}{2}$$

$$\sigma_3 = -\frac{\mathrm{deltaTime}\, q_2}{2}$$

Generate Function:

```
%f = symfun(StateTransition, U)
matlabFunction(f, 'File', 'StateTransitionFunctionGEN');
matlabFunction(Gx, 'File', 'StateJacobianGEN');
matlabFunction(Gu, 'File', 'ControlJacobianGEN');
```

Observation Model:

Gps: Trivial as direct observations

Magnetomer:

```
syms MagX MagY MagZ MagBiasX MagBiasY MagBiasZ 'real'
MagMeasured = [MagX; MagY; MagZ]
```

```
MagMeasured =
```

$$\begin{pmatrix} \mathrm{MagX} \\ \mathrm{MagY} \\ \mathrm{MagZ} \end{pmatrix}$$

```
MagNED = [MagN; MagE; MagD]
```

```
MagNED =
```

$$\begin{pmatrix} \mathrm{MagN} \\ \mathrm{MagE} \\ \mathrm{MagD} \end{pmatrix}$$

```
MagBias = [MagBiasX; MagBiasY; MagBiasZ]
```

```
MagBias =
```

$$\begin{pmatrix} \mathrm{MagBiasX} \\ \mathrm{MagBiasY} \\ \mathrm{MagBiasZ} \end{pmatrix}$$

```
MagMeasured = (TBN * (MagNED + MagBias))
```

```
MagMeasured =
```

$$\begin{pmatrix} (2\,q_1\,q_3 + 2\,q_2\,q_4)\;(\mathrm{MagD} + \mathrm{MagBiasZ}) - (2\,q_1\,q_4 - 2\,q_2\,q_3)\;(\mathrm{MagE} + \mathrm{MagBiasY}) + (\mathrm{MagN} + \mathrm{MagBiasX})\;\left({q_1}^2 + {q_2}^2 - \right. \\ \left(2\,q_3\,q_4 - 2\,{q_2}^2\right)\;(\mathrm{MagD} + \mathrm{MagBiasZ}) + (2\,q_1\,q_4 + 2\,q_2\,q_3)\;(\mathrm{MagN} + \mathrm{MagBiasX}) + (\mathrm{MagE} + \mathrm{MagBiasY})\;\left({q_1}^2 - {q_2}^2 + \right. \\ (2\,q_1\,q_2 + 2\,q_3\,q_4)\;(\mathrm{MagE} + \mathrm{MagBiasY}) - (2\,q_1\,q_3 - 2\,q_2\,q_4)\;(\mathrm{MagN} + \mathrm{MagBiasX}) + (\mathrm{MagD} + \mathrm{MagBiasZ})\;\left({q_1}^2 - {q_2}^2 - \right. \end{pmatrix}$$

UWB:

```
syms ('xb1', 'yb1','xb2', 'yb2', 'xb3', 'yb3', 'xb4', 'yb4', 'xb5', 'yb5', 'xb6', 'yb6','xb7',

h = [sqrt((x - xb1)^2 + (y - yb1)^2 + (z - zb1)^2); ...
     sqrt((x - xb2)^2 + (y - yb2)^2 + (z - zb2)^2); ...
     sqrt((x - xb3)^2 + (y - yb3)^2 + (z - zb3)^2); ...
     sqrt((x - xb4)^2 + (y - yb4)^2 + (z - zb4)^2); ...
     sqrt((x - xb5)^2 + (y - yb5)^2 + (z - zb5)^2); ...
     sqrt((x - xb6)^2 + (y - yb6)^2 + (z - zb6)^2); ...
     sqrt((x - xb7)^2 + (y - yb7)^2 + (z - zb7)^2); ...
     sqrt((x - xb8)^2 + (y - yb8)^2 + (z - zb8)^2); ...
     sqrt((x - xb9)^2 + (y - yb9)^2 + (z - zb9)^2)]
```

```
h =
```

$$\begin{pmatrix} \sqrt{(x-\mathrm{xb}_1)^2+(y-\mathrm{yb}_1)^2+(z-\mathrm{zb}_1)^2} \\ \sqrt{(x-\mathrm{xb}_2)^2+(y-\mathrm{yb}_2)^2+(z-\mathrm{zb}_2)^2} \\ \sqrt{(x-\mathrm{xb}_3)^2+(y-\mathrm{yb}_3)^2+(z-\mathrm{zb}_3)^2} \\ \sqrt{(x-\mathrm{xb}_4)^2+(y-\mathrm{yb}_4)^2+(z-\mathrm{zb}_4)^2} \\ \sqrt{(x-\mathrm{xb}_5)^2+(y-\mathrm{yb}_5)^2+(z-\mathrm{zb}_5)^2} \\ \sqrt{(x-\mathrm{xb}_6)^2+(y-\mathrm{yb}_6)^2+(z-\mathrm{zb}_6)^2} \\ \sqrt{(x-\mathrm{xb}_7)^2+(y-\mathrm{yb}_7)^2+(z-\mathrm{zb}_7)^2} \\ \sqrt{(x-\mathrm{xb}_8)^2+(y-\mathrm{yb}_8)^2+(z-\mathrm{zb}_8)^2} \\ \sqrt{(x-\mathrm{xb}_9)^2+(y-\mathrm{yb}_9)^2+(z-\mathrm{zb}_9)^2} \end{pmatrix}$$

```
UWB = jacobian(h,X)
```

UWB =

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x - 2\,\mathrm{xb}_1}{\sigma_9} & \frac{2\,y - 2\,\mathrm{yb}_1}{\sigma_9} & \frac{2\,z - 2\,\mathrm{zb}_1}{\sigma_9} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x - 2\,\mathrm{xb}_2}{\sigma_8} & \frac{2\,y - 2\,\mathrm{yb}_2}{\sigma_8} & \frac{2\,z - 2\,\mathrm{zb}_2}{\sigma_8} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x - 2\,\mathrm{xb}_3}{\sigma_7} & \frac{2\,y - 2\,\mathrm{yb}_3}{\sigma_7} & \frac{2\,z - 2\,\mathrm{zb}_3}{\sigma_7} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x - 2\,\mathrm{xb}_4}{\sigma_6} & \frac{2\,y - 2\,\mathrm{yb}_4}{\sigma_6} & \frac{2\,z - 2\,\mathrm{zb}_4}{\sigma_6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x - 2\,\mathrm{xb}_5}{\sigma_5} & \frac{2\,y - 2\,\mathrm{yb}_5}{\sigma_5} & \frac{2\,z - 2\,\mathrm{zb}_5}{\sigma_5} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x - 2\,\mathrm{xb}_6}{\sigma_4} & \frac{2\,y - 2\,\mathrm{yb}_6}{\sigma_4} & \frac{2\,z - 2\,\mathrm{zb}_6}{\sigma_4} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x - 2\,\mathrm{xb}_7}{\sigma_3} & \frac{2\,y - 2\,\mathrm{yb}_7}{\sigma_3} & \frac{2\,z - 2\,\mathrm{zb}_7}{\sigma_3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x - 2\,\mathrm{xb}_8}{\sigma_2} & \frac{2\,y - 2\,\mathrm{yb}_8}{\sigma_2} & \frac{2\,z - 2\,\mathrm{zb}_8}{\sigma_2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\,x - 2\,\mathrm{xb}_9}{\sigma_1} & \frac{2\,y - 2\,\mathrm{yb}_9}{\sigma_1} & \frac{2\,z - 2\,\mathrm{zb}_9}{\sigma_1} \end{pmatrix}$$

where

$$\sigma_1 = 2\,\sqrt{(x - \mathrm{xb}_9)^2 + (y - \mathrm{yb}_9)^2 + (z - \mathrm{zb}_9)^2}$$

$$\sigma_2 = 2\,\sqrt{(x - \mathrm{xb}_8)^2 + (y - \mathrm{yb}_8)^2 + (z - \mathrm{zb}_8)^2}$$

$$\sigma_3 = 2\,\sqrt{(x - \mathrm{xb}_7)^2 + (y - \mathrm{yb}_7)^2 + (z - \mathrm{zb}_7)^2}$$

$$\sigma_4 = 2\,\sqrt{(x - \mathrm{xb}_6)^2 + (y - \mathrm{yb}_6)^2 + (z - \mathrm{zb}_6)^2}$$

$$\sigma_5 = 2\,\sqrt{(x - \mathrm{xb}_5)^2 + (y - \mathrm{yb}_5)^2 + (z - \mathrm{zb}_5)^2}$$

$$\sigma_6 = 2\,\sqrt{(x - \mathrm{xb}_4)^2 + (y - \mathrm{yb}_4)^2 + (z - \mathrm{zb}_4)^2}$$

$$\sigma_7 = 2\,\sqrt{(x - \mathrm{xb}_3)^2 + (y - \mathrm{yb}_3)^2 + (z - \mathrm{zb}_3)^2}$$

$$\sigma_8 = 2\,\sqrt{(x - \mathrm{xb}_2)^2 + (y - \mathrm{yb}_2)^2 + (z - \mathrm{zb}_2)^2}$$

$$\sigma_9 = 2\,\sqrt{(x - \mathrm{xb}_1)^2 + (y - \mathrm{yb}_1)^2 + (z - \mathrm{zb}_1)^2}$$

```
matlabFunction(UWB, 'File', 'MeasurementJacobianGEN');
```

Main EKF script used to estimate the pose of a UAV

```
Code describe here is part of a PhD conducted at Liverpool John Moores University
by Harry A.G. Pointon 2016-2020


Ground truth is established using a Trimble S7 total station.


This state estimation system is defined as a testbed for model based
uncertainty experimentation. The aim is to investigate a practical
improvement on the current systems of static uncertainty or adaptive
uncertainty modelling as used in the Pixhawk EKF systems.


Sensors used:
    - IMU                   BNO-055
    - Pozyx UWB             Pozyx with described number of anchors



State Space: X
    q1
    q2
    q3
    q4
    xPrime
    yPrime
    zPrime
    x
    y
    z
Control Input: U
    deltaI
    deltaJ
    deltaK
    deltaX
    deltaY
    deltaZ
```

Setup

```
% ------- Model-Based Variance is used -------

clear

path = 'E:\University\UAV - EKF\QuatV3 9 Anchors\Data Generation\'
```

```
path =
'E:\University\UAV - EKF\QuatV3 9 Anchors\Data Generation\'
```

```
%name = 'SecondFlightW9A';
%name = 'SecondFlightW9A1';
```

```
%name = 'SecondFlightW9A2';
name = 'SecondFlightW9A3';
%name = 'SlowWalk9A';
%name = 'SlowWalk9A1';
%name = 'Wiggle9A';
%name = 'Wiggle9A1';
%name = 'Line9A1';
type = '.mat';
file = strcat(path,name,type);
load(file)


%load('E:\University\UAV - EKF\QuatV1.0 WORKING\Data Generation\BNOandUWBwithRTS.mat')

% Initial Parameters
StartPoint = 1;
QuartInitialParameter = eul2quat([120,0,0])';
%QuartInitialParameter = [-0.4481; 0; 0; 0.8940];
VelocityInitialParameter = [0; 0; 0];
PositionInitialParameter = [data_final(StartPoint,16); data_final(StartPoint,17);...
                            data_final(StartPoint,18)];
%PositionInitialParameter = [99.6; 102.83; 14];

% Initial State - X
X = [QuartInitialParameter; VelocityInitialParameter; PositionInitialParameter];

% % Anchor Locations - insert anchor locations here...
% anchor_pos_1 = [x1,y1,z1];
% anchor_pos_2 = [x2,y2,z2];
% anchor_pos_3 = [x3,y3,z3];
% anchor_pos_4 = [x4,y4,z4];
% anchor_pos_5 = [x5,y5,z5];
% anchor_pos_6 = [x6,y6,z6];
% anchors = [ anchor_pos_1; anchor_pos_2; anchor_pos_3;...
%             anchor_pos_4; anchor_pos_5; anchor_pos_6];
%anchors = anchor_pos;
anchors = AnchorPositions;


% Initial Uncertainty - P
P = eye(10);
P(1,1) = 0.8;
P(2,2) = 0.8;
P(3,3) = 0.8;
P(4,4) = 0.8;
P(5,5) = 0;
P(6,6) = 0;
P(7,7) = 0;
P(8,8) = 0;
P(9,9) = 0;
P(10,10) = 0;

% Process Noise Covariance - Q
```

```
Q = eye(10);
AttiQ = 0.001;
VelQ = 0.007;
Q(1,1) = AttiQ;
Q(2,2) = AttiQ;
Q(3,3) = AttiQ;
Q(4,4) = AttiQ;
Q(5,5) = VelQ;
Q(6,6) = VelQ;
Q(7,7) = VelQ;
Q(8,8) = 0.99;
Q(9,9) = 0.99;
Q(10,10) = 0.99;

% Bias Matrix (If constant biases) - B
% B(1:2:3) = Accelerometer biases
% B(4:5:6) = Gyro biases
%B = [-0.457121, -0, -0.486211 , -0.0041889, -0.00206168, -0.00251885];

B = GetBNOBias();

I = eye(10);
flag = 1;
deltaTime = 1/100; % 100Hz control input rate

%
Logged_H_Values = [];
Logged_State_Vector = [];
Logged_Kalman_gain = [];
Logged_State_Uncertainty = [];
Logged_R_Values = [];
```

Main EKF Loop

```
loggedStatePredict = [];
loggedState = [];

data_final(:,4) = data_final(:,4);
data_final(:,5) = data_final(:,5);
data_final(:,6) = data_final(:,6);
Updates = 0;
Predicts = 0;

for i=StartPoint:1:length(data_final)

    % -------------------------- Prediction Phase ------------------------------
    if isnan(data_final(i,7)) && isnan(data_final(i,8)) && isnan(data_final(i,9))
        U = [data_final(i,1), data_final(i,2), data_final(i,3), data_final(i,4),...
             data_final(i,5), data_final(i,6)]; % U is control input

        % Predict State
        %X = StateTransitionFunction(X, U, B, deltaTime);
```

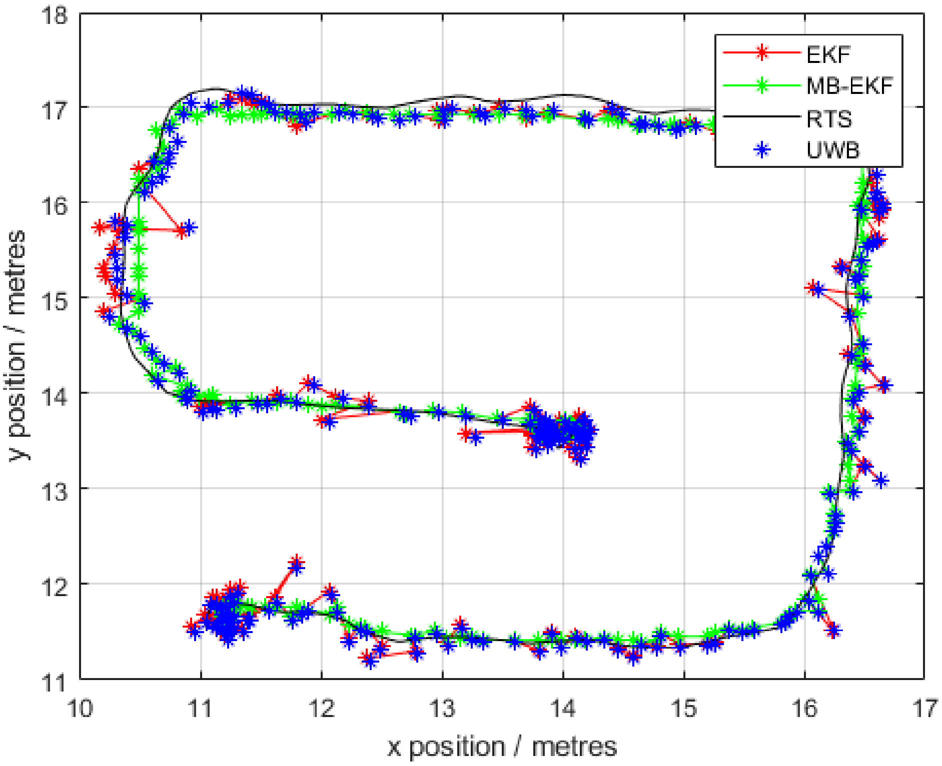
```
%X = HandMadeStateTransition(X, U, B, deltaTime);

g = 9.81;
Deltai = U(1) * -1;
Deltaj = U(2) * -1;
Deltak = U(3);

DeltaX = U(4) * g * -1;
DeltaY = U(5) * g * -1;
DeltaZ = U(6) * g;

XBias = B(1);
YBias = B(2);
ZBias = B(3);
IBias = B(4);
JBias = B(5);
KBias = B(6);

q1 = X(1);
q2 = X(2);
q3 = X(3);
q4 = X(4);
xDot = X(5);
yDot = X(6);
zDot = X(7);
x = X(8);
y = X(9);
z = X(10);

Quart = [q1; q2; q3; q4];

Vel = [xDot; yDot; zDot];

Pos = [x; y; z];

deltaI = ((Deltai - IBias)) * deltaTime;
deltaJ = ((Deltaj + JBias)) * deltaTime;
deltaK = ((Deltak + KBias)) * deltaTime;

deltaxDot = (DeltaX + XBias) * deltaTime;
deltayDot = (DeltaY + YBias) * deltaTime;
deltazDot = (DeltaZ + ZBias) * deltaTime;

AngularChange = [deltaI; deltaJ; deltaK];
VelocityChange = [deltaxDot; deltayDot; deltazDot];

ChangeInQuart = eul2quat([AngularChange(1,1), AngularChange(2,1),...
                          AngularChange(3,1)]);

NextQuart = quatmultiply(Quart.', ChangeInQuart);

NextQuart = NextQuart.';
```

```
        DCM = quat2dcm(Quart.');

        NextVel = Vel + ((DCM * (VelocityChange * deltaTime)) - ([0; 0; g] * deltaTime));

        NextPos = Pos + (Vel * deltaTime);

        State = [NextQuart; NextVel; NextPos];

        X = State;


        loggedStatePredict = horzcat(loggedStatePredict, X);

        %Predict State Uncertainty Ahead
        % Calculate the jacobian of state for covariance
        Gx = StateJacobian(U,B,deltaTime,X);
        % Gu is used to calculate the process noise of the system, as i am
        % tuning this i will not need Gu, instead i've just got Q defined
        % above.
        % Gu = ControlJacobian(U); % Calculate the jacobian of the control input
        P  = (Gx * P *  transpose(Gx)) + Q; % predicted state covariance
        Predicts = Predicts + 1;


        % -------------------------- If measurement is available ------------------------
    elseif isnan(data_final(i,7)) == 0
        % Get measurements
        measurements = [data_final(i,7), data_final(i,8), data_final(i,9),...
            data_final(i,10), data_final(i,11),data_final(i,12), ...
            data_final(i,13), data_final(i,14),data_final(i,15)];
        measurements = measurements / 1000; % mm to m

        % Calculate resudual and sensor measurement variance
        [Y,RVal] = MeasurementResidual(X, measurements, anchors); % Y is residual R is SMN
        R = RVal / 1000; % R val is in mm, EKF working in m
        Logged_R_Values = cat(3, Logged_R_Values, RVal);


        % Calclate measurement Jacobian
        % Jacobian of the equations used to convert measurements to state
        H = MeasurementJacobian(X, anchors);
        Logged_H_Values = cat(3, Logged_H_Values, H);

        % Calculate Kalman Gain
        K = ((P * transpose(H)) * ( (((H * P)*transpose(H)) + (R))^-1 ));
        Logged_Kalman_gain = cat(3, Logged_Kalman_gain, K);
%         if K(1,1) ~= 0
%             K = K + 0.005;
%         end
        if isnan(K(1,1)) % freakout detection
            disp('NaN detected at Kalman Gain update')
            break
        end
```

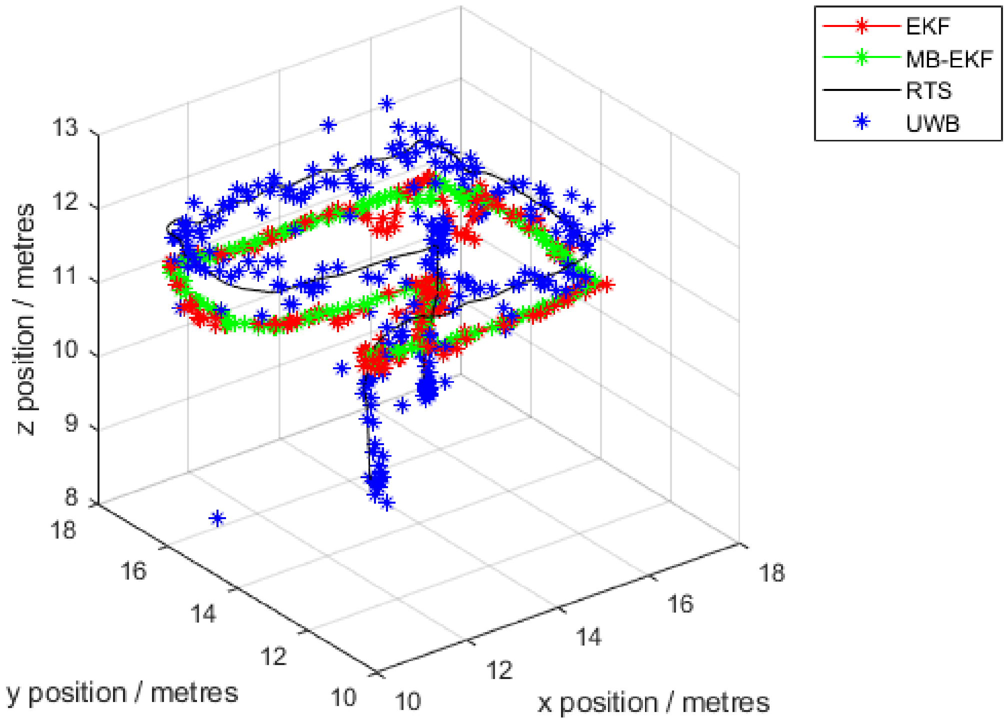
```
        % Update state uncertainty
        P = (I - (K*H)) * P;%* (transpose(I - (K*H)) + (K * R * transpose(K)) );
        % Log state uncertainty
        Logged_State_Uncertainty = cat(3, Logged_State_Uncertainty, P);
        if isnan(P(1,1)) % freakout detection
            disp('NaN detected at State uncertainty update')
            break
        end

        if X(8) == 0
            ZeroLogMeasurements = horzcat(ZeroLogMeasurements, measurements');
        end
        X = X + (K * Y);
        loggedState = horzcat(loggedState, X);

        % Log data to outut 1,2,3 are x,y,z
        output(i,1) = X(8); % X
        output(i,2) = X(9); % Y
        output(i,3) = X(10); % Z
        Updates = Updates + 1;

    end


    Logged_State_Vector = cat(3, Logged_State_Vector, X);

end
```

Analysis of results:

```
%loggedState(10,:) = loggedState(10,:)+0.5;

Sigma = 0.5;
N = 10;


figure(2)
plot(loggedState(8,:), loggedState(9,:),'r*-')
hold on
plot(MBEKFX, MBEKFY, 'g*-')
plot(ComparisonDataMatrix(:,4), ComparisonDataMatrix(:,5), 'k')
plot(ComparisonDataMatrix(:,1), ComparisonDataMatrix(:,2), 'b*')
legend('EKF', 'MB-EKF', 'RTS', 'UWB')
grid on
ylabel('y position / metres')
xlabel('x position / metres')
hold off
```




```
figure
plot3(loggedState(8,:), loggedState(9,:), loggedState(10,:), 'r*-')
hold on
plot3(MBEKFX, MBEKFY, MBEKFZ, 'g*-')
plot3(ComparisonDataMatrix(:,4), ComparisonDataMatrix(:,5), ComparisonDataMatrix(:,6), 'k')
%plot3(data_final(:,16), data_final(:,17), data_final(:,18),'b*')
plot3(ComparisonDataMatrix(:,1), ComparisonDataMatrix(:,2), ComparisonDataMatrix(:,3), 'b*')
legend('EKF', 'MB-EKF', 'RTS', 'UWB')
grid on
ylabel('y position / metres')
xlabel('x position / metres')
zlabel('z position / metres')
hold off
```




```
UWBErrorX = ComparisonDataMatrix(:,1) - ComparisonDataMatrix(:,4);
UWBErrorY = ComparisonDataMatrix(:,2) - ComparisonDataMatrix(:,5);
UWBErrorZ = ComparisonDataMatrix(:,3) - ComparisonDataMatrix(:,6);

MBEKFErrorX = MBEKFX' - ComparisonDataMatrix(:,4);
MBEKFErrorY = MBEKFY' - ComparisonDataMatrix(:,5);
MBEKFErrorZ = MBEKFZ' - ComparisonDataMatrix(:,6);
MBEKF = [MBEKFX;MBEKFY;MBEKFZ];

EKFErrorX = loggedState(8,:)' - ComparisonDataMatrix(:,4);
EKFErrorY = loggedState(9,:)' - ComparisonDataMatrix(:,5);
EKFErrorZ = loggedState(10,:)' - ComparisonDataMatrix(:,6);

MBEKFErrorMeans = [mean(MBEKFErrorX), mean(MBEKFErrorY), mean(MBEKFErrorZ)]
```

```
MBEKFErrorMeans = 1×3
    0.0182   -0.0445   -0.1643
```

```
EKFErrorMeans = [mean(EKFErrorX), mean(EKFErrorY), mean(EKFErrorZ)]
```

```
EKFErrorMeans = 1×3
    0.0120   -0.0428   -0.1853
```

```
UWBErrorMeans = [mean(UWBErrorX), mean(UWBErrorY), mean(UWBErrorZ)]
```

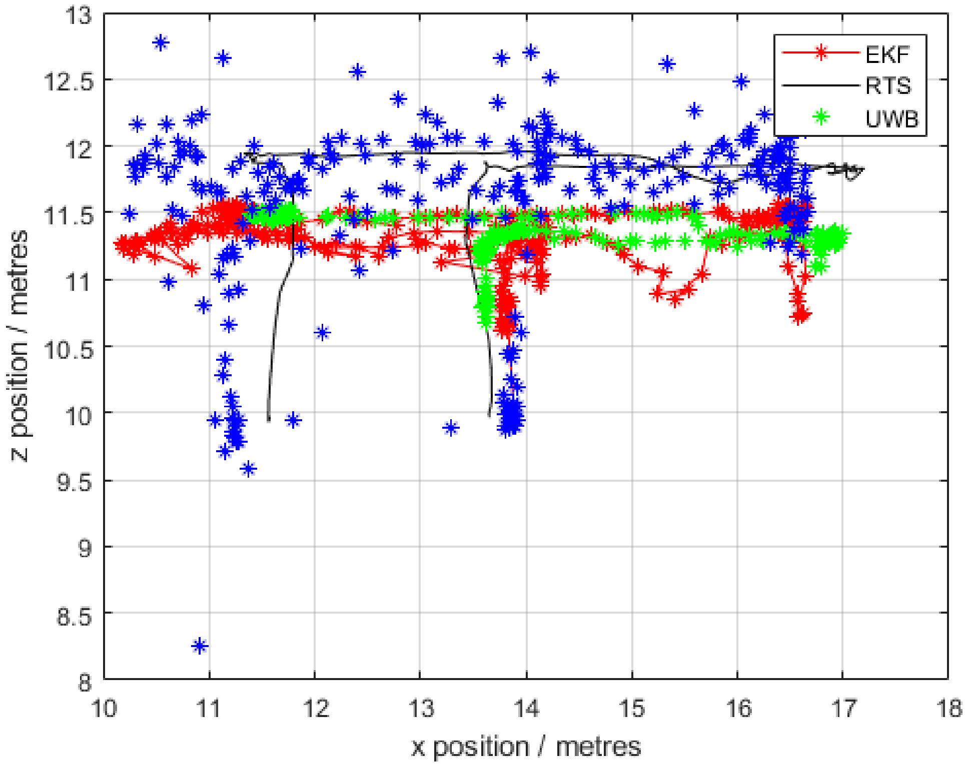
```
UWBErrorMeans = 1×3
```

```
    0.0448   -0.0654   -0.0477
```

```
Means = [MBEKFErrorMeans; EKFErrorMeans; UWBErrorMeans];

figure
plot(loggedState(8,:), loggedState(10,:), 'r*-')
hold on
plot(ComparisonDataMatrix(:,5), ComparisonDataMatrix(:,6), 'k')
plot(MBEKFY, MBEKFZ,'g*')
plot(UWB_X, UWB_Z, 'b*')
legend('EKF','RTS','UWB')
grid on
ylabel('z position / metres')
xlabel('x position / metres')
hold off

save(name, 'ComparisonDataMatrix', 'MBEKF', 'loggedState', 'Means')

% figure()
% for i = 1:1:length(Logged_Kalman_gain)
%     K_Mean(i) = mean(mean(Logged_Kalman_gain(:,:,i)));
% end
% yyaxis left
% plot(K_Mean)
% ylabel("Kalman gain")
% yyaxis right
% plot(loggedState(10,:)-1.2, 'r*-')
% ylabel("z estimate / m")
% legend("Kalman gain", "z axis estimate")
% xlabel("Time step")
% QuantAnalysisEKFV2


% LoggedQuat = loggedStatePredict(1:4,:);
% for i = 1:1:length(LoggedQuat)
%     [LoggedEul(1,i), LoggedEul(2,i), LoggedEul(3,i)] = quat2angle(LoggedQuat(:,i)');
% end
%
% figure(3)
% subplot(4,2,1)
% plot(LoggedQuat(1,:))
% subplot(4,2,3)
% plot(LoggedQuat(2,:))
% subplot(4,2,5)
% plot(LoggedQuat(3,:))
% subplot(4,2,7)
% plot(LoggedQuat(4,:))
% subplot(4,2,2)
% plot(data_final(:,1))
% subplot(4,2,4)
% plot(data_final(:,2))
% subplot(4,2,6)
% plot(data_final(:,3))
```

```
% hold off
%
% figure(4)
% subplot(3,1,1)
% plot(rad2deg(LoggedEul(1,:)))
% subplot(3,1,2)
% plot(rad2deg(LoggedEul(2,:)))
% subplot(3,1,3)
% plot(rad2deg(LoggedEul(3,:)))
% hold off
%
% figure(5)
% title('Velocity')
% subplot(3,2,1)
% plot(loggedStatePredict(5,:))
% subplot(3,2,3)
% plot(loggedStatePredict(6,:))
% subplot(3,2,5)
% plot(loggedStatePredict(7,:))
% subplot(3,2,2)
% plot(loggedState(5,:))
% subplot(3,2,4)
% plot(loggedState(6,:))
% subplot(3,2,6)
% plot(loggedState(7,:))
% hold off
%
% % subplot(3,2,2)
% % plot(data_final(:,4))
% % subplot(3,2,4)
% % plot(data_final(:,5))
% % subplot(3,2,6)
% % plot(data_final(:,6))
% % hold off
%
%
% figure(6)
% title('Positional comparison x, y, z.')
% subplot(3,1,1)
% plot(loggedStatePredict(8,:), 'b')
% hold on
% plot(data_final(:,13), 'r')
% plot(data_final(:,16), 'k')
% subplot(3,1,2)
% plot(loggedStatePredict(9,:), 'b')
% hold on
% plot(data_final(:,14), 'r')
% plot(data_final(:,17), 'k')
% subplot(3,1,3)
% plot(loggedStatePredict(10,:), 'b')
% hold on
% plot(data_final(:,15), 'r')
% plot(data_final(:,18), 'k')
```

```
%
% figure
% subplot(3,1,1)
% plot(loggedState(8,:), 'k')
% hold on
% %plot(data_final(:,13))
% %plot(UWB_X(1:170),'r')
%
% subplot(3,1,2)
% plot(loggedState(9,:),'k')
% hold on
% %plot(data_final(:,14))
% %plot(UWB_Y(1:170),'r')
%
% subplot(3,1,3)
% plot(loggedState(10,:),'k')
% hold on
% %plot(data_final(:,15))
% %plot(UWB_Z(1:170),'r')
%
%
% figure(8)
% for i=1:1:6
%     subplot(3,2,i)
%     plotData = squeeze(Logged_R_Values(i,i,:));
%     plot(plotData)
%     hold on
% end


hold off
```


13
12.5
12
11.5
11
10.5
10
9.5
9
8.5
8
10
11
12
13
14
15
16
17
18
z position / metres
x position / metres
EKF
RTS
UWB

```
function [Y, R] = MeasurementResidual(X, M, A)
%MEASUREMENTRESIDUAL Calculation of UWB measurement related
 information.
%   A: Anchors
%   M: Measurement
%   X: Current state estimate
%   R: Sensor Measurement Noise
%   Y: Measurement residuals

% Clear sensor measurement uncertainty
R = [];
R = eye(9);

% Anchor locations
xb1 = A(1,1);
yb1 = A(1,2);
zb1 = A(1,3);

xb2 = A(2,1);
yb2 = A(2,2);
zb2 = A(2,3);

xb3 = A(3,1);
yb3 = A(3,2);
zb3 = A(3,3);

xb4 = A(4,1);
yb4 = A(4,2);
zb4 = A(4,3);

xb5 = A(5,1);
yb5 = A(5,2);
zb5 = A(5,3);

xb6 = A(6,1);
yb6 = A(6,2);
zb6 = A(6,3);

xb7 = A(7,1);
yb7 = A(7,2);
zb7 = A(7,3);

xb8 = A(8,1);
yb8 = A(8,2);
zb8 = A(8,3);

xb9 = A(9,1);
yb9 = A(9,2);
zb9 = A(9,3);

% Predicted location
x = X(8);
```

```
y = X(9);
z = X(10);

% Range to anchors
zm_1 = M(1);
zm_2 = M(2);
zm_3 = M(3);
zm_4 = M(4);
zm_5 = M(5);
zm_6 = M(6);
zm_7 = M(7);
zm_8 = M(8);
zm_9 = M(9);


% % Predictive uncertainty calculation:
% R(1,1) = PredictiveUncertainty(zm_1)^2;
% R(2,2) = PredictiveUncertainty(zm_2)^2;
% R(3,3) = PredictiveUncertainty(zm_3)^2;
% R(4,4) = PredictiveUncertainty(zm_4)^2;
% R(5,5) = PredictiveUncertainty(zm_5)^2;
% R(6,6) = PredictiveUncertainty(zm_6)^2;

% Predictive uncertainty calculation:
% R(1,1) = PredictiveUncertainty(zm_1/1000)^2;
% R(2,2) = PredictiveUncertainty(zm_2/1000)^2;
% R(3,3) = PredictiveUncertainty(zm_3/1000)^2;
% R(4,4) = PredictiveUncertainty(zm_4/1000)^2;
% R(5,5) = PredictiveUncertainty(zm_5/1000)^2;
% R(6,6) = PredictiveUncertainty(zm_6/1000)^2;
% R(7,7) = PredictiveUncertainty(zm_7/1000)^2;
% R(8,8) = PredictiveUncertainty(zm_8/1000)^2;
% R(9,9) = PredictiveUncertainty(zm_9/1000)^2;

R = (eye(9) * 0.1)^2;

% Location calculations
ze_1 = sqrt((x - xb1)^2 + (y - yb1)^2 + (z - zb1)^2);
ze_2 = sqrt((x - xb2)^2 + (y - yb2)^2 + (z - zb2)^2);
ze_3 = sqrt((x - xb3)^2 + (y - yb3)^2 + (z - zb3)^2);
ze_4 = sqrt((x - xb4)^2 + (y - yb4)^2 + (z - zb4)^2);
ze_5 = sqrt((x - xb5)^2 + (y - yb5)^2 + (z - zb5)^2);
ze_6 = sqrt((x - xb6)^2 + (y - yb6)^2 + (z - zb6)^2);
ze_7 = sqrt((x - xb7)^2 + (y - yb7)^2 + (z - zb7)^2);
ze_8 = sqrt((x - xb8)^2 + (y - yb8)^2 + (z - zb8)^2);
ze_9 = sqrt((x - xb9)^2 + (y - yb9)^2 + (z - zb9)^2);

% Return measurement residuals
Y = [(zm_1 - ze_1); (zm_2 - ze_2);
     (zm_3 - ze_3); (zm_4 - ze_4);
     (zm_5 - ze_5); (zm_6 - ze_6);
     (zm_7 - ze_7); (zm_8 - ze_8);
     (zm_9 - ze_9)];
```

```
end
```

```
Not enough input arguments.

Error in MeasurementResidual (line 14)
xb1 = A(1,1);
```

Published with MATLAB® R2018a

```
function UWB = MeasurementJacobian(X, anchors)
%MEASUREMENTJACOBIANGEN
%    UWB =
 MEASUREMENTJACOBIANGEN(X,XB1,XB2,XB3,XB4,XB5,XB6,XB7,XB8,XB9,Y,YB1,YB2,YB3,YB4,YB

%    This function was generated by the Symbolic Math Toolbox version
 8.1.
%    16-Aug-2019 09:44:57

x = X(8);
y = X(9);
z = X(10);

xb1 = anchors(1,1);
xb2 = anchors(2,1);
xb3 = anchors(3,1);
xb4 = anchors(4,1);
xb5 = anchors(5,1);
xb6 = anchors(6,1);
xb7 = anchors(7,1);
xb8 = anchors(8,1);
xb9 = anchors(9,1);

yb1 = anchors(1,2);
yb2 = anchors(2,2);
yb3 = anchors(3,2);
yb4 = anchors(4,2);
yb5 = anchors(5,2);
yb6 = anchors(6,2);
yb7 = anchors(7,2);
yb8 = anchors(8,2);
yb9 = anchors(9,2);

zb1 = anchors(1,3);
zb2 = anchors(2,3);
zb3 = anchors(3,3);
zb4 = anchors(4,3);
zb5 = anchors(5,3);
zb6 = anchors(6,3);
zb7 = anchors(7,3);
zb8 = anchors(8,3);
zb9 = anchors(9,3);

t2 = x-xb1;
t3 = y-yb1;
t4 = z-zb1;
t5 = t2.^2;
t6 = t3.^2;
t7 = t4.^2;
t8 = t5+t6+t7;
t9 = 1.0./sqrt(t8);
t10 = x.*2.0;
```

```
t11 = x-xb2;
t12 = y-yb2;
t13 = z-zb2;
t14 = y.*2.0;
t15 = t11.^2;
t16 = t12.^2;
t17 = t13.^2;
t18 = t15+t16+t17;
t19 = 1.0./sqrt(t18);
t20 = z.*2.0;
t21 = x-xb3;
t22 = y-yb3;
t23 = z-zb3;
t24 = t21.^2;
t25 = t22.^2;
t26 = t23.^2;
t27 = t24+t25+t26;
t28 = 1.0./sqrt(t27);
t29 = x-xb4;
t30 = y-yb4;
t31 = z-zb4;
t32 = t29.^2;
t33 = t30.^2;
t34 = t31.^2;
t35 = t32+t33+t34;
t36 = 1.0./sqrt(t35);
t37 = x-xb5;
t38 = y-yb5;
t39 = z-zb5;
t40 = t37.^2;
t41 = t38.^2;
t42 = t39.^2;
t43 = t40+t41+t42;
t44 = 1.0./sqrt(t43);
t45 = x-xb6;
t46 = y-yb6;
t47 = z-zb6;
t48 = t45.^2;
t49 = t46.^2;
t50 = t47.^2;
t51 = t48+t49+t50;
t52 = 1.0./sqrt(t51);
t53 = x-xb7;
t54 = y-yb7;
t55 = z-zb7;
t56 = t53.^2;
t57 = t54.^2;
t58 = t55.^2;
t59 = t56+t57+t58;
t60 = 1.0./sqrt(t59);
t61 = x-xb8;
t62 = y-yb8;
t63 = z-zb8;
t64 = t61.^2;
```

```
t65 = t62.^2;
t66 = t63.^2;
t67 = t64+t65+t66;
t68 = 1.0./sqrt(t67);
t69 = x-xb9;
t70 = y-yb9;
t71 = z-zb9;
t72 = t69.^2;
t73 = t70.^2;
t74 = t71.^2;
t75 = t72+t73+t74;
t76 = 1.0./sqrt(t75);
UWB =
 reshape([0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
xb1.*2.0).*(1.0./2.0),t19.*(t10-xb2.*2.0).*(1.0./2.0),t28.*(t10-
xb3.*2.0).*(1.0./2.0),t36.*(t10-xb4.*2.0).*(1.0./2.0),t44.*(t10-
xb5.*2.0).*(1.0./2.0),t52.*(t10-xb6.*2.0).*(1.0./2.0),t60.*(t10-
xb7.*2.0).*(1.0./2.0),t68.*(t10-xb8.*2.0).*(1.0./2.0),t76.*(t10-
xb9.*2.0).*(1.0./2.0),t9.*(t14-yb1.*2.0).*(1.0./2.0),t19.*(t14-
yb2.*2.0).*(1.0./2.0),t28.*(t14-yb3.*2.0).*(1.0./2.0),t36.*(t14-
yb4.*2.0).*(1.0./2.0),t44.*(t14-yb5.*2.0).*(1.0./2.0),t52.*(t14-
yb6.*2.0).*(1.0./2.0),t60.*(t14-yb7.*2.0).*(1.0./2.0),t68.*(t14-
yb8.*2.0).*(1.0./2.0),t76.*(t14-yb9.*2.0).*(1.0./2.0),t9.*(t20-
zb1.*2.0).*(1.0./2.0),t19.*(t20-zb2.*2.0).*(1.0./2.0),t28.*(t20-
zb3.*2.0).*(1.0./2.0),t36.*(t20-zb4.*2.0).*(1.0./2.0),t44.*(t20-
zb5.*2.0).*(1.0./2.0),t52.*(t20-zb6.*2.0).*(1.0./2.0),t60.*(t20-
zb7.*2.0).*(1.0./2.0),t68.*(t20-zb8.*2.0).*(1.0./2.0),t76.*(t20-
zb9.*2.0).*(1.0./2.0)],[9,10]);
```

```
Not enough input arguments.

Error in MeasurementJacobian (line 8)
x = X(8);
```

Published with MATLAB® R2018a