Indoor localization using Local Positioning Systems.

MSc.

By

SHIVAN RAMDHANIE

Department of Engineering Mathematics
UNIVERSITY OF BRISTOL & UNIVERSITY OF THE WEST OF ENGLAND

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of MASTER OF SCIENCE in the Faculty of Engineering.

SEPTEMBER 2020

Word count: TBD

Abstract

Here goes the abstract

Dedication and acknowledgements

Here goes the dedication.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .... DATE: ....

Nomenclature

- UAV - Unmanned Aerial Vehicle.
- GPS - Global Positioning System.
- LPS - Local Positioning System.
- FCU - Flight Controller Unit.
- UWB - Ultra-WideBand.
- TOF - Time of Flight.
- PSoC - Programmable System on Chip.
- NLOS - No Line of Sight.
- TWR - Two Way Ranging.
- I2C- Inter-Integrated Circuit Bus.
- SPI - Serial Peripheral Interface.
- OSH - Open-Source Hardware.

Table of Contents

Page

List of Tables

Table **Page**

List of Figures

CHAPTER 1

INTRODUCTION

In recent years Unmanned aerial vehicle(UAV) usages has grown exponentially becoming common in industry and households Custers (2016). A major part of UAV applications is their ability to localise themselves in the given environment with acceptable precision and accuracy. This is a common requirement in any robotic system but UAV's are often limited by strict payload requirements and therefore have to rely on sensors that are lightweight and robust. (Mendoza-Mendoza et al. 2020) gives a good summary of physical components that are used in various vehicles but a UAV system, specifically, a quadrotor system cab be summarised as follows:

- Rotor build - This section contains parts that should be researched based on the size and physical requirements of the drone. These include: brushless motors, electronic speed controllers, frame size.

- Flight controller unit(FCU) - This acts as the mother board and brain of the quadrotor system. It collates data from various sensors, sends commands to the motors and if there is a companion computer is attached collects and sends data to it. Commercial FCU's contain the various control systems and laws required for stable flight and movement. Most have an array of sensors built in.

- Sensors - These vary from from inertial, positioning, barometric and camera. Aside from inertial and barometric sensors that are present in most FCU's, sensors are chosen based on the environment and use case of the system.

- Companion computer - In some cases higher level processing is required by the system to execute autonomy and a secondary computer is used to do this.

- Transmitter and Receiver - This is used to implement manual control over the drone by a user.

Further delving into the sensors, we can classify UAV's based on their operating environment, indoors or outdoors. These give rise to two forms of localisation and navigation systems:

- Global Positioning Systems (GPS) - As the name suggests this setup uses GPS as well as other sensors.
- GPS-denied - These systems do not have access to GPS due to their operating environment.

In outdoor applications GPS provides a reliable and fairly accurate way to localise with use of several other sensors. However, indoor applications are denied the benefits of GPS and often must use other sensors for the task of localisation. Utilizing a similar concept of triangulation used by GPS a local positioning system(LPS) can be used for indoor environments. (Pozyx Labs 2018) has developed a commercial system that utilizes Ultra-WideBand technology(UWB) with a bandwidth of $\approx 500MHz$.

With indoor environments users have more control of the environment so a LPS can create a feasible solution for indoor localisation for UAV's/robots operating there. The core idea of this research would be to integrate a commercial LPS directly into an existing FCU to produce accurate position estimates that can be used for autonomy. The measurements from the LPS would then be transformed into observations of the state of the UAV and fused with other observations from other sensors. This fused pose estimate would then be fed into the companion computer for off-board processing.

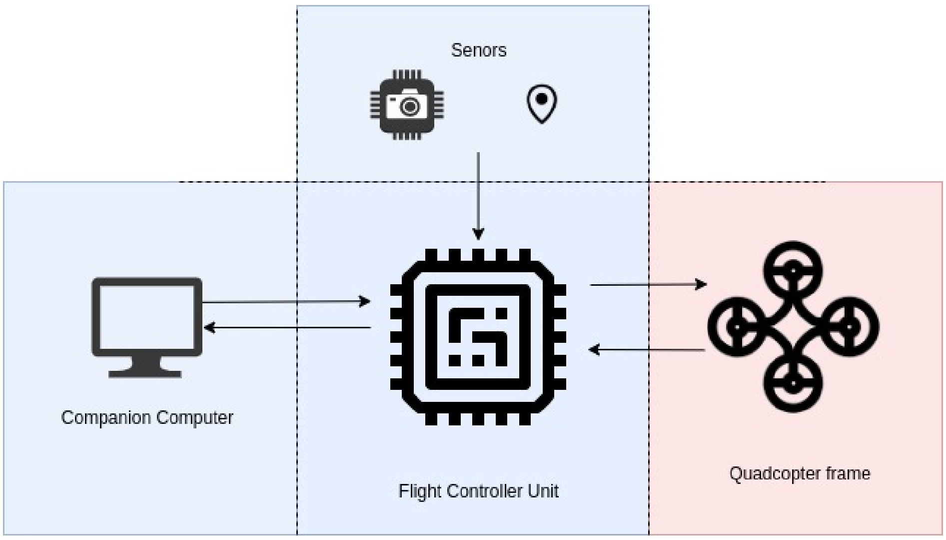


Figure 1.1: The typical setup for an autonomous UAV.

Figure: 1.1 shows a typical setup for UAV. Parts of the system highlighted in blue represent systems that would be worked on during the course of this research. The idea is that the system being designed should provide localisation data which should be independent of the rotor build. These will be further scoped in the upcoming sections but it will involve doing a quality exercise of the LPS tp determine measurement uncertainty and limitations, writing additions or modifying the firmware of the FCU to integrate the LPS and setting up the piplines for a companion computer to receive the pose estimates and use them.

1.1 Aims & Objectives

In 1 we briefly touched on what would be addressed over the course of this research. Expanding on that, the research would entail the use of the commercial version of an UWB sensor for positioning from Pozyx Labs (2018). At a high level the project and be split into three modules that must be researched, unit tested and finally integrated. Figure 1.1 highlights the major systems within the project and are as follows:

- The Pozyx LPS providing measurements that will be used in localisation.
- A flight controller collating fusing various observations from sensors to provide a pose estimate.
- A companion computer to visualise and utilise the pose information in a meaningful manner.

From these systems and the overall aim of indoor localisation the following objectives were created:

- Evaluation and qualitative analysis of the LPS, documentation limitations from previous done work and current physical setups as well as compare with other ranging standards.
- Based on the qualitative analysis and experiments determine the best configuration in a household to place the anchors for the system.
- Use the incoming data from the sensors to produce a suitable measurement/observation model for the pose of the system.
- Relay the data to a flight controller unit via a suitable hardware interface.
- Delve into the firmware of the flight controller and apply sensor fusion algorithms on the flight controller to provide pose estimates.
- Pipe the pose estimates to a companion computer for visualisation and higher level control of a UAV.

All of these objectives can be completed without flying the UAV autonomously. Given the current situation and time-frame it was determined that setting up the pipelines to visualise the localisation in realtime from the companion computer is adequate for the last objective. Furthermore, with the autonomous flight being out of scope of this project much of the work fell into software engineering to achieve the overall aim. Broadly, this means delving into the software libraries and interfaces for the Pozyx sensor, modifying and making additions to the Ardupilot flight stack to integrate the Pozyx sensor with the (pixhawk?) FCU, and finally digging into the MAVLINK protocol and libraries to use the pose estimates on a Raspberry Pi 3 Model B+ PSoC. To achieve these objectives a solid software engineering aproach would need to be applied with familiarity of Python and C++ programming languages.

1.2 Background Research

Indoor localisation has become a core part of many system in recent years. These range from robotics, multimedia, logistics and sporting systems. Modern localisation systems can be split in active or passive systems, active systems require the system being localised to have electronics to either process or send information that will be used to determine location. In passive systems the position is determine based on a variance of a measured signal or image data. As noted by Deak et al. (2012), some of these techniques include, Received Signal Strength Indicator (RSSI), Time of Arrival (TOA), Time Difference of Arrival (TDOA) and Angle of Arrival (AOA). The Pozyx commercial system uses UWB signals with a TOF technique in order to determine the position of a receiver (tag) in a network of transmitters (anchors). Since processing is done on-board the tag, it falls under the active localisation category. Active systems are ideal for indoor localisation systems for drones since the positional data can be fed directly to FCU's or companion computers in order to correct pose estimates calculated by the system directly.

As noted by the producers of Pozyx the core of the system uses a communication bandwidth of $\approx 500MHz$, this results in pulses of $0.16ns$ wide. Assuming that speed of light is $299792458ms^-1$ we get pulses of length $0.04797m$ which is very small and hence robust to noise from reflections. The major factors affecting the performance of the system would be materials that would slow down the signals before they reach the tag. So No Line of Sight (NLOS), conductors and changing mediums of travel are noted to affect the performance the most.

With increasing complexity of FCU's it is possible to do relatively dense calculations in a real time scenario without delegating them to a separate processing system. This is beneficial to indoor drone systems since they need to be small and maneuverable. A standard FCU comes equipped with several standard communication interfaces (I2C, Serial, SPI) so integrating external sensors is possible. Furthermore, multiple autopilot firmware provides a Hardware Abstraction Layer (HAL) making any sensor integration developed on one unit easily ported to another system. Additionally, on board libraries contain sensor fusion implementations (Extended Kalman Filter (EKF)) that can combine the Pozyx data and on-board sensor data to provide fairly accurate positional data while in motion.

CHAPTER 2

LITERATURE REVIEW

2.1 Indoor Localisation Systems

2.1.1 Passive Systems

In summary, passive systems do not require the object being tracked to have some of electronics on them to do positioning. Some examples of passive systems are (Deak et al. 2012):

- Computer Vision and Imaging systems.
- Tactile and contact sensors.
- Attenuation of signals.
- Differential air pressure.

A common example of computer vision based localisation is to use a setup consisting of multiple cameras in a space trying to detecting a single object. Using the intrinsic and extrinsic properties of each camera it is possible to determine the transform to the object in a given frame with relatively high accuracy. A prime example of this is the commercial VICON motion capture systems. Aerial Robotics IITK (n.d.) shows a drone application in which the UAV is positioned via using a VICON motion capture system. Figure 2.1 shows the simplified setup, it should be noted that the UAV must be equipped with specialised marker that is used to identify it. Furthermore, the positions are fed to a companion computer connected to the autopilot system. The VICON setup provides highly accurate positions and is often used to gather ground truth positional data to compare to other positioning systems. The additional requirements of the VICON systems, however, do not make it feasible for indoor applications.

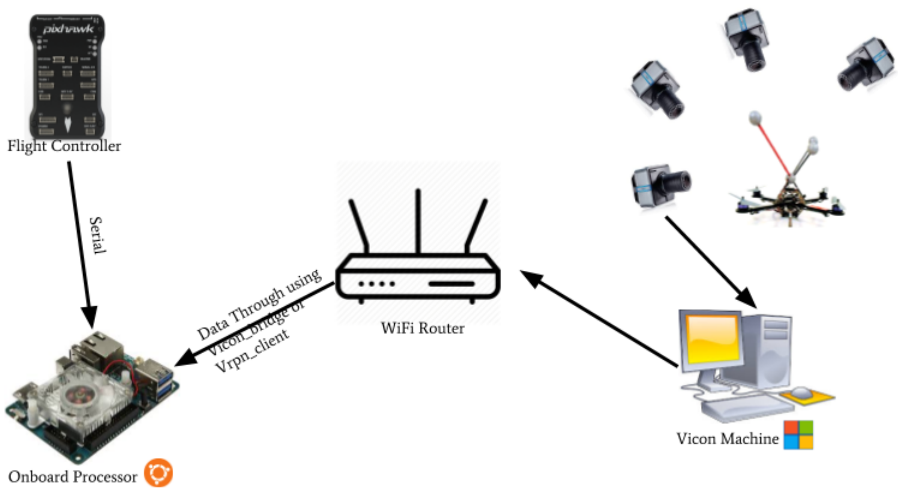


Figure 2.1: VISON setup for position of a UAV. (`https://aerial-robotics-iitk.gitbook.io/wiki/estimation/setup-with-vicon`)

2.1.2 Active Systems

In contrast to passive systems, active systems have the object being positioned equipped with electronics. Many indoor localisation techniques use this and some examples are (Deak et al. 2012)

- Radio-frequency identification
- UWB
- Wireless Local Area Network
- Bluetooth Low energy (BLE)

Many of these setups use an anchor and tag configuration. The tag receives signals from multiple anchors and triangulates the tag.

An approach using and comparing UWB and BLE is developed by Jiménez & Seco (2017) to do localisation in a museum. Both methods are combined with a dead reckoning system to improve accuracy. Six paintings are equipped with both a BLE and UWB tag. The test setup first did a calibration where both sensors were placed at fixed points in the museum with a clear line of sight to each tag. From initial ranging performance, the UWB setup was shown to perform better with a distance variance of $\pm 0.4m$ while the BLE setup had errors of over $10m$. It is noted by the authors that a ranging approach with BLE is challenging since it uses an RSSI method but can match the accuracy of the UWB setup when combined a dead reckoning system after some initial steps by the user.

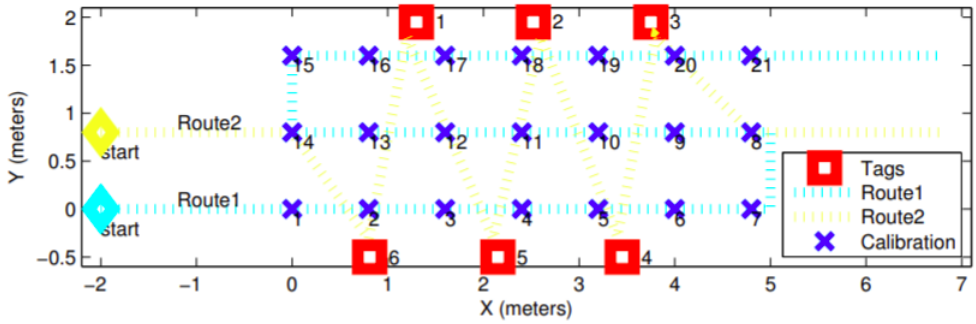


Figure 2.2: Setup used to compare UWB and BLE performance in a museum. Jiménez & Seco (2017) Page: 3

2.2 Sensor Fusion

From the previous section we have described the basic principle of operation of the Pozyx system. With a tag we are able to at least determine a rough estimate of the position of a tag in a given reference frame. Additionally, the tag has an Inertial Measurement Unit (IMU), consisting of an Accelerometer, Gyroscope and Magnetometer, and an Altimeter. These are used in one of the operational modes of the Pozyx system in order to improve accuracy. The idea of sensor fusion is to combine multiple sources of data in order to get a fairly accurate estimate of the pose of the system, The measurements from the tag can be combined with the sensors onward a FCU in order to achieve this. The de-facto sensor fusion technique is called the Extended Kalman Filter (EKF). Chadaporn et al. (2014) present a useful description and example of how the EKF works. Algorithmically, the EKF is a recursive process using predictions based on the dynamics of the vehicles and updating the estimate based on these predictions and measurements from various sources. The major requirement for the EKF is that the process model and the measurement model are differentiable. The steps for the EKF are as follows:

1. Provide and initial estimate for the state, $\hat{x}_k^+$, and the prediction error, P_k^+.
2. Compute the Kalman gain, $K_k = P_k^+ H_k^T (H_k P_k^+ H_k^T + R)^{-1}$
3. Update the estimate with measurement z_k, $\hat{x}_k = \hat{x}_k^+ + K_k(z_k - h(\hat{x}_k^+))$
4. Update the prediction error, $P_k = (I - K_k H_k) P_k^+$
5. Project the state ahead, $\hat{x}_{k+1}^+ = f(\hat{x}_k, u_k, w)$
6. Project the Prediction error ahead, $P_{k+1}^+ = A_k P_k A_k^T + Q_k$

7. Repeat from step 2.

Where K is the Kalman gain, R is the Measurement Noise Covariance Matrix, Q is the Process Noise Covariance matrix. Tsai (1998) document work for localising a mobile robot using ultrasonic measurements. Although, the system was a wheeled mobile robot instead of a UAV, the calculations using the TOF measurements and dead reckoning data as well as the use of an EKF to improve the pose estimates are similar to what is trying to be achieved in this current project. A notable contribution of the work is the use of a voting scheme to obtain the best 'observation' for the the robot's orientation before being fed into the EKF algorithm.

2.3 Pozyx - Behind the Scenes

For this research project a commercially available, active UWB sensor system is proposed to aid in localisation of an indoor UAV system.

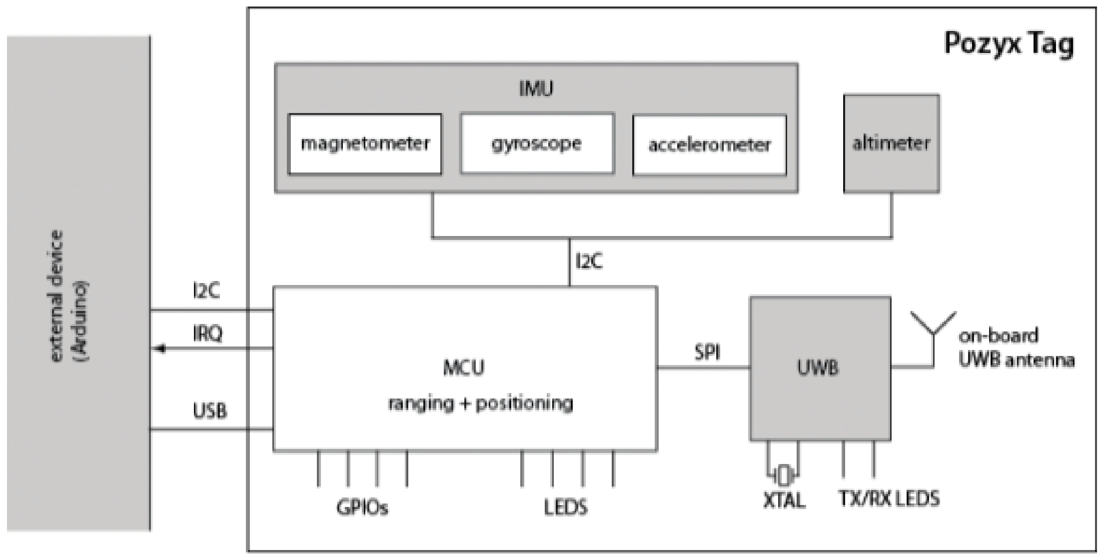


Figure 2.3: Simplified Block Diagram of the Pozyx tag.

2.3.1 Ultra-WideBand (UWB)

The core of the Pozyx system operates using an UWB approach. UWB is a short range, low energy, high bandwidth communication radio technology. Radio waves travel at the speed of light ($c = 299792458ms^{-1}$) so using a TOF approach the range between a tag and an anchor can be obtained simply by:

$$d = c * TOF$$

Knowing the position of each anchor in a given reference frame, Mimoune et al. (2019) discuss a method to use raw range readings in order to determine the position of the tag. The positions can be described by the following system of equations:

$$\begin{bmatrix} (x-x_1)^2+(y-y_1)^2+(z-z_1)^2=d_1^2 \\ \vdots \\ (x-x_n)^2+(y-y_n)^2+(z-z_n)^2=d_n^2 \end{bmatrix} \tag{2.1}$$

Where n represents an index of an anchor and (x,y,z) represents the position of the tag. This can be converted into matrix form of **A.x = B**:

$$\begin{bmatrix} 1-2x_1-2y_1-2z_1 \\ \vdots \\ 1-2x_n-2y_n-2z_n \end{bmatrix} * \begin{bmatrix} x^2+y^2+z^2 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_1^2-x_1^2-y_1^2-z_1^2 \\ \vdots \\ d_n^2-x_n^2-y_n^2-z_n^2 \end{bmatrix} \tag{2.2}$$

The position of the tag can be calculated as:

$$\hat{x}=(A^TA)^{-1}A^TB$$

From the algorithms and work presented we can confirm that the Pozyx system can achieve an accuracy of $\pm 10cm$ in standard environments with LOS. The work confirms that LOS of the anchors to tag is a major factor in accuracy and this will be taken into account when anchors are being placed in the research. Furthermore, the work mentioned in this section proposes an algorithm with raw range readings in order to do localisation, my research will focus on the integration of the Pozyx tag with a standard FCU, so the pose data coming directly from the tag can be used. The pose can be obtained via two modes: 1). A pure Two Way Ranging (TWR) Approach or 2.) A tracking approach using a Kalman prediction filter in addition to the TWR pose.

Additional work done by Di Pietra et al. (2019) also perform an evaluation of UWB systems for indoor applications. The work compares several commercial UWB systems available to consumers and their viability in ranging and pose estimation. The sensors are treated as black box systems and the proprietary algorithms for each system was used. The results documented in this paper shows useful steps to do a primary evaluation of multiple anchor configurations in a given space.

2.3.2 Pozyx Localisation

In contrast to the raw readings obtained in the previous work, the Pozyx system has been used in several localisation systems, both indoors and outdoors. Experiments done by DeStefano et al. (2019) use the Pozyx system in multiple scenarios for the purpose of education. The setup is simple and the two dimensional positions, x and y, form the basis of graphs that can be used to determine average velocities from linear interpolation of the a position versus time graph. Figures

2.4 and 2.5 show both how the physical setup looked and the some of the data collected. Although the UWB positioning system has advantages over several other systems the researchers note that using the Pozyx system in this scenario is not ideal for several reasons. Due to the accuracy of the system ($\pm 10cm$) leads to large variance in instantaneous velocity when calculations are done on a point to point basis.

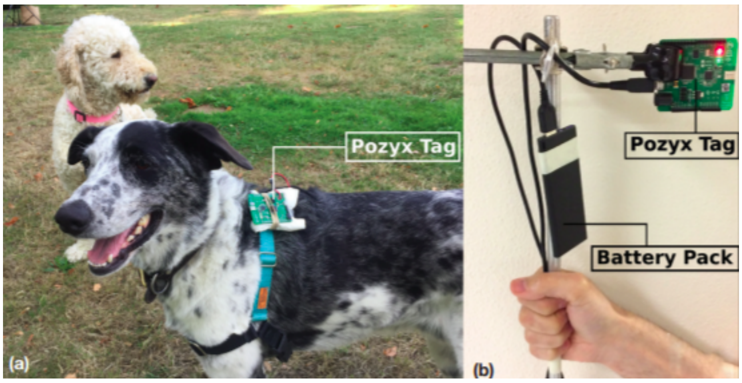


Figure 2.4: The experimental setups for both a dog and a student. DeStefano et al. (2019) Page: 1.

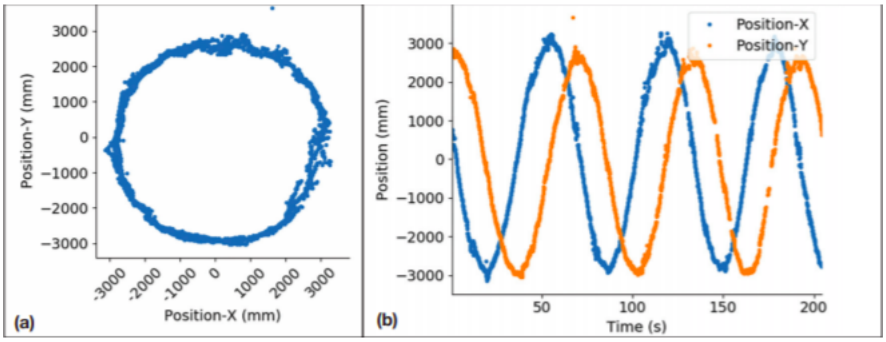


Figure 2.5: Data collected from a student walking in a circle. DeStefano et al. (2019) Page: 2.

Conceição et al. (2017) present a method for using the Pozyx in an outdoor environment. The systems is proposed to operate in two ways: 1.) Using raw range values from the Pozyx sensor and using an EKF externally for pose estimates. 2.) The pose estimates coming from the Pozyx sensor itself. A modified version of the EKF is described in the first scenario. An additional check is done for outliers. Outliers can be described as measurements that do not fit the pattern of the previous measurements. If an outlier is detected it is simply not considered in that time step and the previous calculated values are used instead. The test environment was an outdoor scenario with range error characterisation test which allowed for optimal anchor placement and a suitable testbed for implementing multiple positioning algorithms. A similar methodology would be adapted for this work with focus on indoor limitations and the presence of dynamic obstacles (humans).

2.3.3 Pozyx - Arduino Implementation

Before continuing it should be noted that Ardupilot (2019) has addressed the idea of combining the Pozyx system with a Flight controller using the Ardupliot firmware. The implementation uses the Pozyx tag's compatibility with the Arduino UNO R3 or R2 pin layout. The Pozyx Arduino library is used to gather the relevant information from the Pozyx tag and then send it via serial to the FCU. This research aims to bridge this gap in the hardware and remove the need for the Arduino UNO for the indoor navigation. The major driving force of this is to minimize the amount of extra hardware that should be mounted on and indoor UAV.

2.3.4 Summary

From a overview of work done with UWB technology and the Pozyx system we can see it is a well researched area in terms of evaluating the performance in static scenarios with not much obstacles. However, there seems to be a gap of these systems being tested and localised in households where dynamic obstacles can be present in the area. Furthermore, a major objective of this research would be to directly integrate the Pozyx system directly with a FCU and utilize the sensor fusion systems on board and evaluate the accuracy of the pose estimates. To that end, a similar approach taken by Di Pietra et al. (2019) would be used to first find an optimal anchor configuration and then evaluate it similar to how experiments were carried out by Conceição et al. (2017).

CHAPTER 3

RESEARCH METHODOLOGY

From the literature there is a clear lack of results for localisation systems using the commercial Pozyx sensor network with drones in an household environment. To address the aims and objectives stated in 1.1, it is proposed that a lightweight on-board localisation be developed to check the feasibility of minimal hardware solution for an indoor drone. This would entail connecting a Pozyx tag directly to a FCU and integrate it with the pose localisation system existing already. To close the loop the pose information should be available in some format that can be used for autonomous control. Furthermore, the system should be tested in a practical environment, to this end, the Pozyx anchor network was setup in a kitchen. A kitchen represents one of the highest traffic areas in a household and it contains various materials that will make raw readings from a UWB system noisy and inaccurate. As a kitchen will contain both dynamic and static obstacles multiple anchor configurations must be tested in order to find optimal anchor positions in this given environment.

3.1 Anchor Configurations

Noting work from Di Pietra et al. (2019) and the setup procedures from Pozyx Labs (2018) it is important to have at least 4 anchors setup in non-planar orientations for the best positioning results. To determine a suitable anchor configuration in the given space a tag was placed at fixed known point in a given reference frame and the mounted locations of the 4 anchors were varied on each wall of the kitchen in order to prevent planar configurations and ambiguity when the tag calculates its position. Figure 3.1 shows a simplified layout and toybox configuration of the tag and anchors. Grey areas represent areas in the space that is impossible to traverse, red represents known obstacles that interfere and attenuate the UWB signal, blue represent anchors, green the tag, cream is partially obstructed and the rest of the area is fully traversable. Before each run and data collection the tag was manually configured with the location of each of the anchors in the reference frame of the kitchen and recorded for at least 5 seconds. The tag was triggered to calculate the position repeatedly within the timeframe and the function call is blocking and takes $\approx 70ms$ so data was recorded at a frequency of $\approx 14.28Hz$. With a tag position of (1480,2330,980)mm, the following statistics were calculated using Matlab. In addition to the average error and standard deviation, kurtosis and skew metrics were added to show the number of outliers and symmetry of the recorded data. The error metrics are important to give an idea of how noisy the overall data is while kurtosis and skewness gives an indication if the noise follows a uniform distribution which would be beneficial ideal for state estimators. From table 3.2 we can see that configuration 12 gives acceptable error metrics and reasonable skew and kurtosis making it the best out of the configurations tested. To further support this Figure 3.2 shows several results obtained with several configurations and it can be seen that configuration 12 produced the best results with minimal noise and measurment outliers as well as it was centered around the ground truth position of the tag.

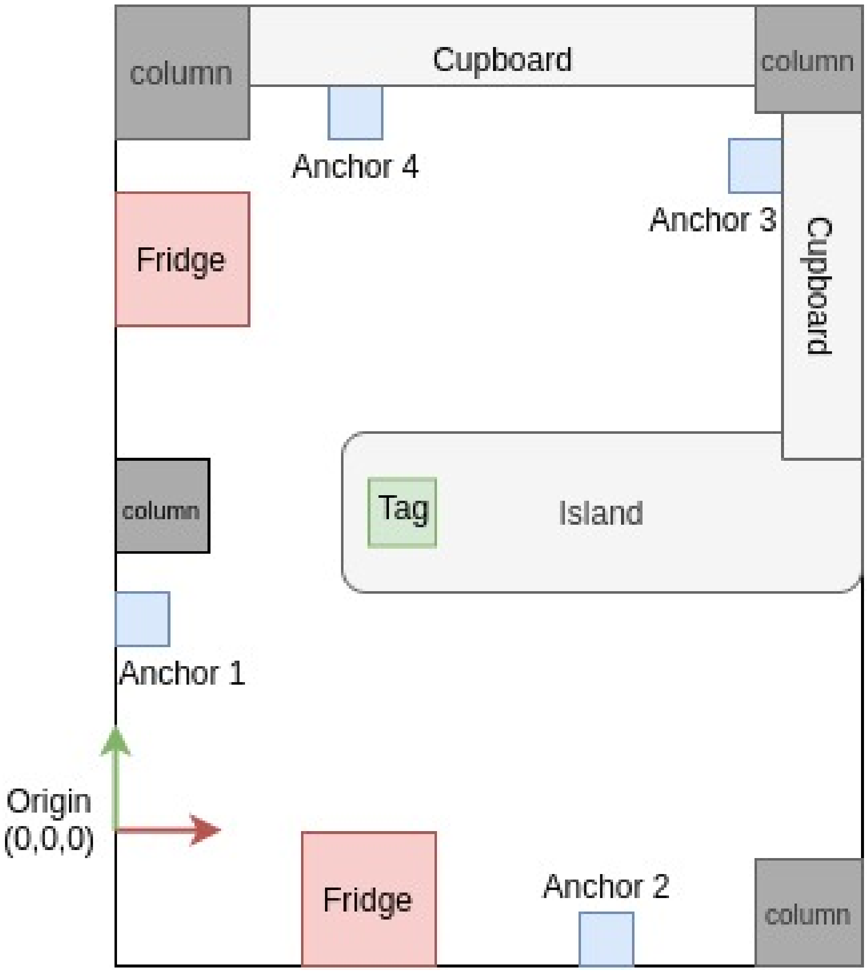


Figure 3.1: Birds eye view of the test environment.

Config	Anchor positions (mm) $\begin{pmatrix} Anchor1(x,y,z), \\ Anchor2(x,y,z), \\ Anchor3(x,y,z), \\ Anchor4(x,y,z) \end{pmatrix}$
1	$\begin{pmatrix} (0,0,1115), \\ (3680,-405,1550), \\ (3655,4080,1906), \\ (270,4465,2090) \end{pmatrix}$
2	$\begin{pmatrix} (0,665,1115), \\ (2995,-405,1889), \\ (3655,4080,1906), \\ (270,4465,2090) \end{pmatrix}$

3		$\begin{pmatrix} (0,665,1115), \\ (2995,-405,1889), \\ (3655,4080,1906), \\ (1526,4559,837) \end{pmatrix}$
4		$\begin{pmatrix} (0,665,1115), \\ (2995,-405,1889), \\ (3655,4080,1906), \\ (1070,5170,491) \end{pmatrix}$
5		$\begin{pmatrix} (0,665,1115), \\ (2995,-405,1889), \\ (3960,3368,2304), \\ (1070,5170,491) \end{pmatrix}$
6		$\begin{pmatrix} (0,665,1115), \\ (2737,-410,1913), \\ (3655,3598,1668), \\ (1187,4760,590) \end{pmatrix}$
7		$\begin{pmatrix} (0,665,1115), \\ (2737,-410,1913), \\ (3655,4529,1777), \\ (1187,4760,590) \end{pmatrix}$
8		$\begin{pmatrix} (0,645,1214), \\ (2737,-410,1913), \\ (3651,4120,1853), \\ (1066,4760,491) \end{pmatrix}$
9		$\begin{pmatrix} (0,645,1214), \\ (2737,-410,1913), \\ (3970,3063,2320), \\ (1066,4760,491) \end{pmatrix}$
10		$\begin{pmatrix} (0,645,1214), \\ (2737,-410,1913), \\ (3651,3550,1810), \\ (1066,4760,491) \end{pmatrix}$
11		$\begin{pmatrix} (0,645,1214), \\ (2737,-410,1913), \\ (3651,3550,1810), \\ (1591,4450,1775) \end{pmatrix}$

12		$\begin{pmatrix}(0,645,1214),\\(2737,-410,1913),\\(3651,4120,1853),\\(1591,4450,1775)\end{pmatrix}$

Table 3.1: Anchor locations for each Configuration

Config	Avg. Error	Std. Deviation $\begin{pmatrix}x,\\y,\\z\end{pmatrix}$	Kurtosis	Skewness
1	342.5906	$\begin{pmatrix}183.1707,\\205.3426,\\972.6239\end{pmatrix}$	$\begin{pmatrix}2.349,\\1.5967,\\1.415\end{pmatrix}$	$\begin{pmatrix}-0.4083,\\-0.3522,\\0.4551\end{pmatrix}$
2	173.5938	$\begin{pmatrix}43.2345,\\45.6897,\\133.8616\end{pmatrix}$	$\begin{pmatrix}21.8923,\\10.1765,\\119.1344\end{pmatrix}$	$\begin{pmatrix}-2.8672,\\-0.3453,\\8.9025\end{pmatrix}$
3	203.8502	$\begin{pmatrix}128.4693,\\118.3216,\\209.1637\end{pmatrix}$	$\begin{pmatrix}14.2955,\\11.2045,\\2.9124\end{pmatrix}$	$\begin{pmatrix}-2.579,\\-0.3920,\\0.4055\end{pmatrix}$
4	85.8562	$\begin{pmatrix}40.2197,\\40.6081,\\66.2120\end{pmatrix}$	$\begin{pmatrix}6.7558,\\7.5180,\\3.4722\end{pmatrix}$	$\begin{pmatrix}-0.4260,\\-0.5344,\\-0.1242\end{pmatrix}$
5	64.8616	$\begin{pmatrix}65.3883,\\53.8458,\\78.5751\end{pmatrix}$	$\begin{pmatrix}13.3707,\\13.2178,\\11.0111\end{pmatrix}$	$\begin{pmatrix}-0.6016,\\0.3745,\\0.0461\end{pmatrix}$
6	189.933	$\begin{pmatrix}40.8435,\\32.6482,\\58.5618\end{pmatrix}$	$\begin{pmatrix}11.8957,\\11.8924,\\11.092\end{pmatrix}$	$\begin{pmatrix}0.7516,\\-0.0100,\\-0.1322\end{pmatrix}$
7	100.6929	$\begin{pmatrix}75.8269,\\72.0940,\\41.9398\end{pmatrix}$	$\begin{pmatrix}26.0285,\\17.4295,\\7.2328\end{pmatrix}$	$\begin{pmatrix}-3.1681,\\-0.8384,\\-0.7151\end{pmatrix}$
8	138.1276	$\begin{pmatrix}26.2243,\\21.8750,\\70.6189\end{pmatrix}$	$\begin{pmatrix}4.6048,\\52.2374,\\50.9467\end{pmatrix}$	$\begin{pmatrix}-0.4858,\\5.0752,\\5.1757\end{pmatrix}$
9	258.296	$\begin{pmatrix}165.7105,\\95.081,\\475.0623\end{pmatrix}$	$\begin{pmatrix}2.2833,\\1.7939,\\2.3406\end{pmatrix}$	$\begin{pmatrix}-0.6982,\\0.2746,\\0.7517\end{pmatrix}$

10	235.348	$\begin{pmatrix} 116.1306, \\ 89.9868, \\ 420.8322 \end{pmatrix}$	$\begin{pmatrix} 3.6670, \\ 4.7058, \\ 3.8066 \end{pmatrix}$	$\begin{pmatrix} 1.5079, \\ -1.7613, \\ -1.5615 \end{pmatrix}$
11	223.4468	$\begin{pmatrix} 120.9674, \\ 79.8602, \\ 409.8192 \end{pmatrix}$	$\begin{pmatrix} 3.389, \\ 4.6663, \\ 3.1988 \end{pmatrix}$	$\begin{pmatrix} 1.1099, \\ -1.5397, \\ -1.2038 \end{pmatrix}$
12	111.8493	$\begin{pmatrix} 27.568, \\ 17.7124, \\ 64.4278 \end{pmatrix}$	$\begin{pmatrix} 4.8522, \\ 7.4335, \\ 6.2489 \end{pmatrix}$	$\begin{pmatrix} -0.0239, \\ -0.8535, \\ 3.7132 \end{pmatrix}$

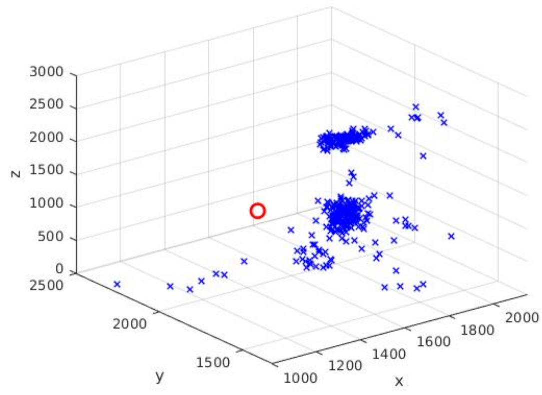
Table 3.2: Statistics of the data recorded for each configuration.

Figure: 3.3 shows the physcial locations of the pozyx anchors with respect to obstacles in the environment. Some things to note are:

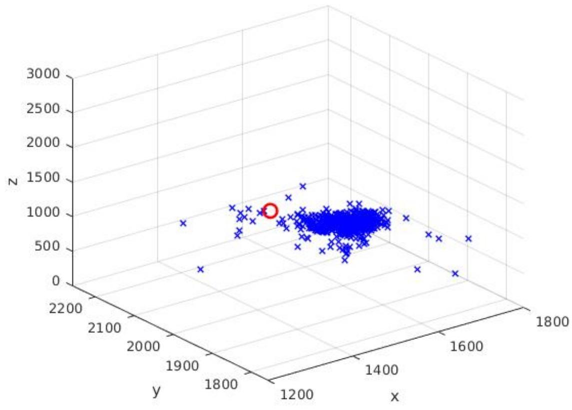
- The walls are made from concrete and dry wall and there are no underlying materials that affected the performance of the anchors.
- The fridges have dimensions of $(600 * 600 * 1750)$mm.
- The corner of the island is located at $(1525, 2106, 918)$ and it has a dimension of $(2450 * 900 * 918)$mm.
- Above the island is considered traversible while beneath it will be considered to be impossible to traverse to simply future experiments.

Summary

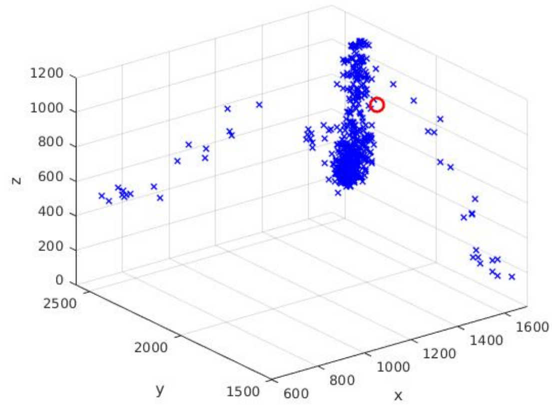
The initial results allowed for a suitable configuration to be obtained but results were recorded in an ideal scenario. Although these provide a good baseline for what to expect under good operational conditions, the parameters and limitations of the system would need to be tested before any form of optimisation and improvement to position estimates can be done. In the next Section: 3.2 we investigate this.



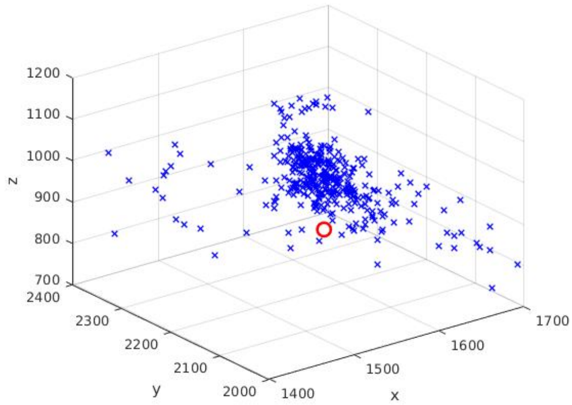
(a) Plot of Config 1



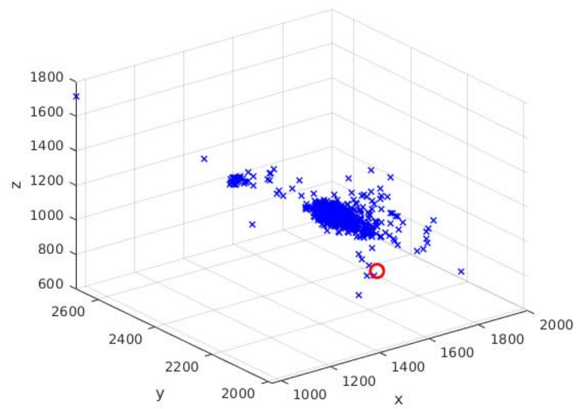
(b) Plot of Config 2



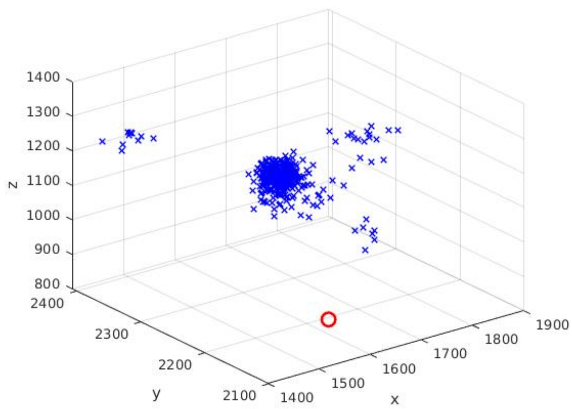
(c) Plot of Config 3



(d) Plot of Config 4

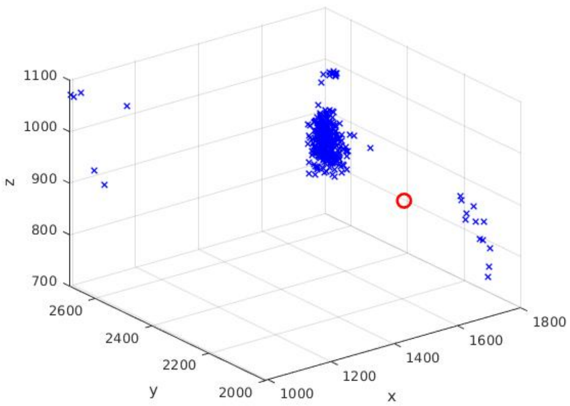


(e) Plot of Config 5

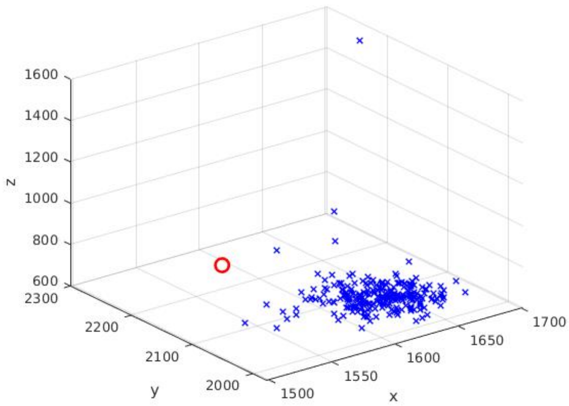


(f) Plot of Config 6

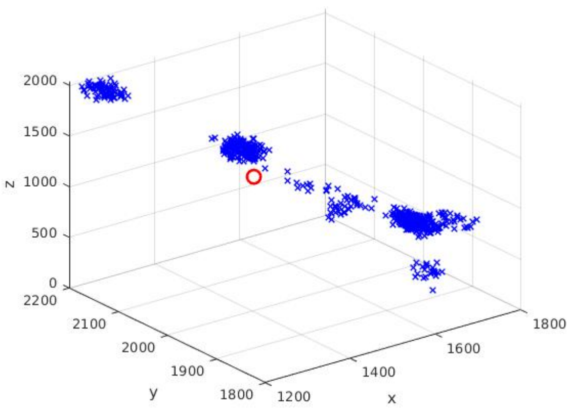
Figure 3.2: Sample plots of several anchor configurations.



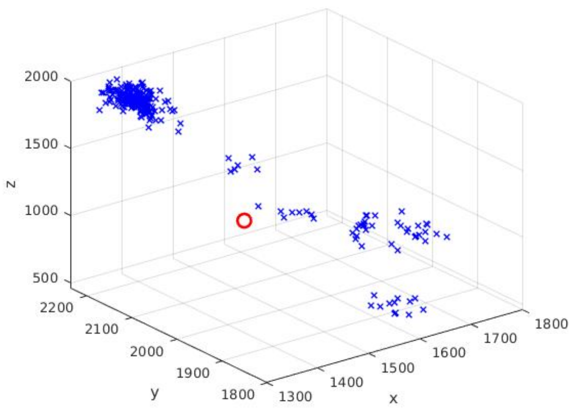
(g) Plot of Config 7



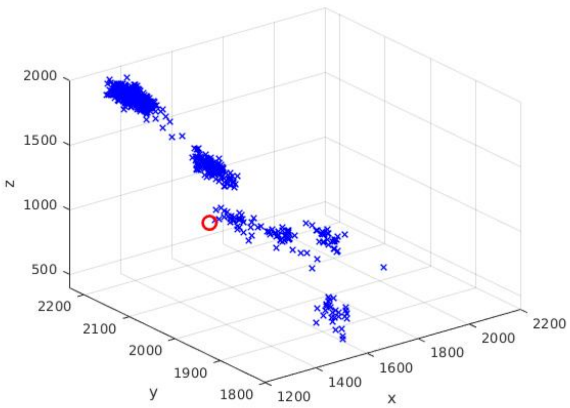
(h) Plot of Config 8



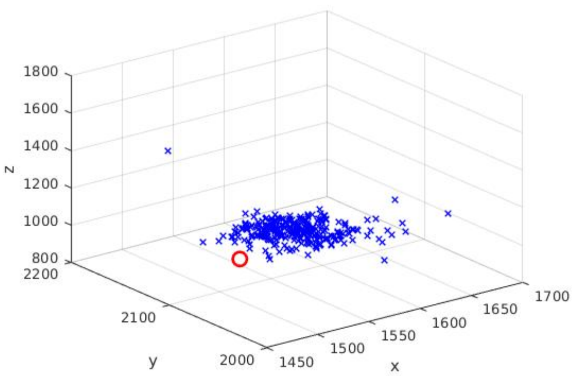
(i) Plot of Config 9



(j) Plot of Config 10



(k) Plot of Config 11



(l) Plot of Config 12

Figure 3.2: Sample plots of several anchor configurations. (cont'd)

(a) Anchor 1

(b) Anchor 2

(c) Anchor 3

(d) Anchor 4

(e) Actual Kitchen Layout

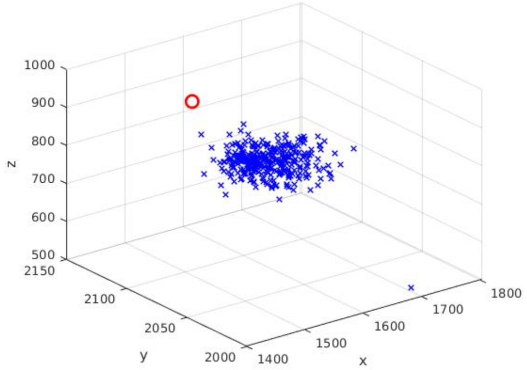
Figure 3.3: Physical layout of the kitchen/Test Area

3.2 Operation Parameters

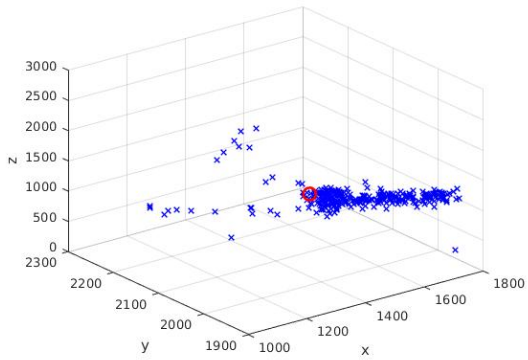
As mentioned in the previous section: 3.1 we determined the best anchor configurations for the given environment but the results were collected in an ideal scenario. From work done by Mimoune et al. (2019) we can see that the major factor affecting the positional accuracy seems to be No Line Of Sight (NLOS) between the anchors and a tag at any given time. Furthermore the work discusses the use of using an optimal triplet of anchors in order to improve accuracy this is an indication that a loss of a single anchor in the optimal configuration determined in the previous section should still yield reasonable results. To confirm there operational parameters the following experiments were carried out.

3.2.1 Loss of Anchor

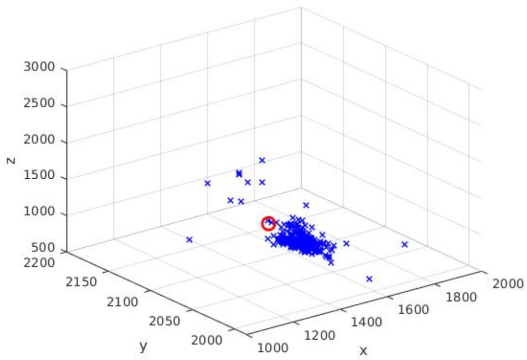
In theory a minimum of three anchors are all that is needed for determining the two dimensional position, (x, y), of the tag via trilateration. Losing a single anchor would therefore allow for the (x, y) position to be determined with a fair amount of accuracy with discrepancies showing up in the z position readings. To test this data was recorded starting with all four anchors and then systematically one achor was turned of in each sample. Figure: 3.4 shows the plots obtained under this test. It can be see that the (x, y) position remains relatively unchanged in all with only a notable shift the z position with the loss of Anchor 1 and more noise introduced with the loss of Anchor 4.



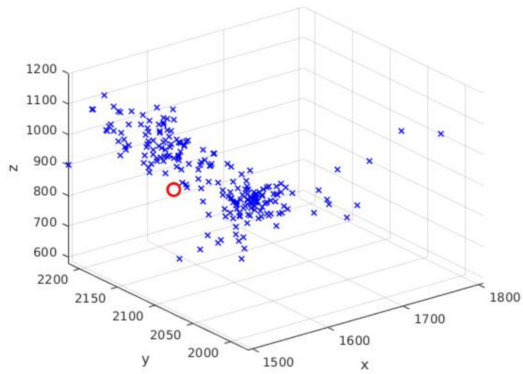
(a) Loss of Anchor 1



(b) Loss of Anchor 2



(c) Loss of Anchor 3



(d) Loss of Anchor 4

Figure 3.4: Results obtained when an Anchor was lost.

3.2.2 No Line of Sight

The positioning calculations are based on a Two way ranging and Time of flight Calculation scheme see Appendix: A. The major crux of the calculation is that the wave is always moving at the speed of light this means that introducing an obstacle between the any anchor and the tag that attenuates the signal would introduce discrepancies in the results obtained. Figure: 3.5 shows the setup used to test this scenario. The system was allowed to record data with no obstacles then a person walked along the trajectory indicated by the dotted arrows in the Figure and came to a rest as seen. It was ensured that the path taken by the person did not obstruct any of the anchors during motion. This was carried out 3 times:

1. Position 1: Provides NLOS between Anchor 2 and the tag.

2. Position 2: Does not provide NLOS between any anchors and the tag.

3. Position 3: Provides NLOS between Anchro 4 and the tag.

Position 2 was done to confirm that no reflections or attenuation occur with a person just being in the proximity of the tag. From Figure: 3.6 we can see that a person being introduced at position 1 and which obstructs line of sight between an anchor and the tag has adverse effects. It is clear that the 2 distinct blobs seen in the plots for position 1 and 3 show that the tag's 'position' has changed although it is stationary.

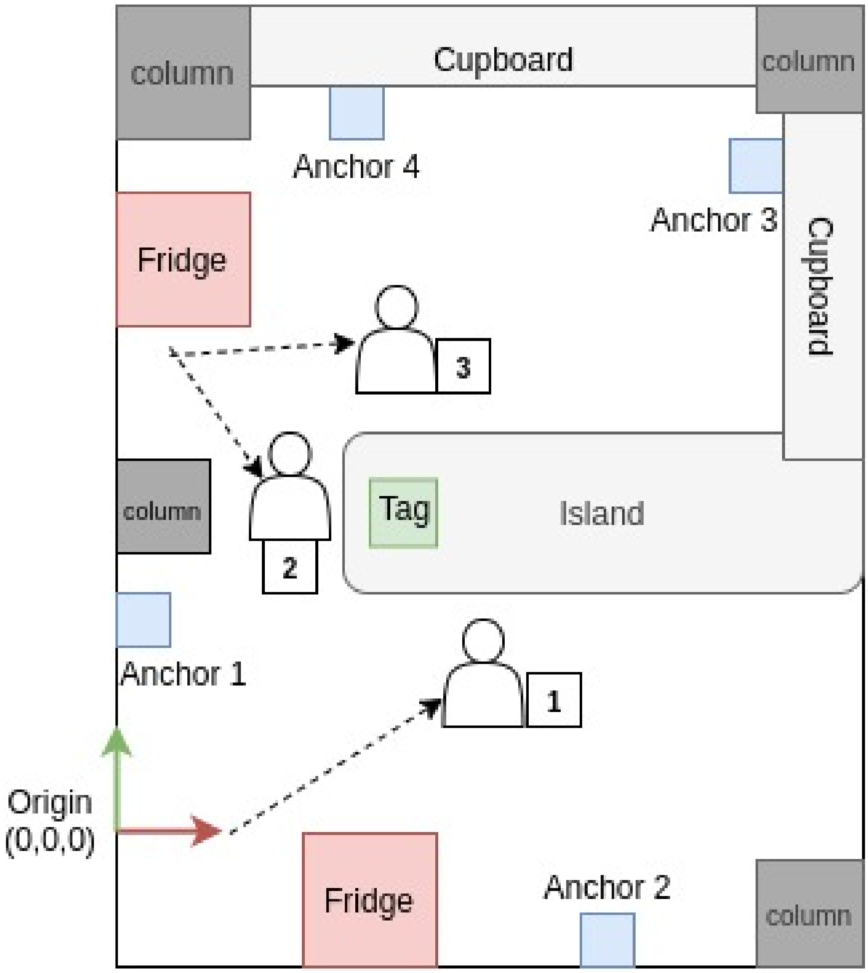
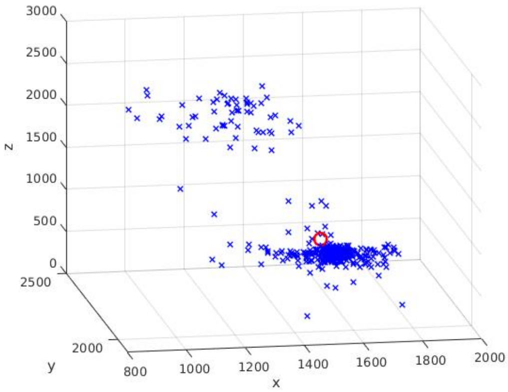
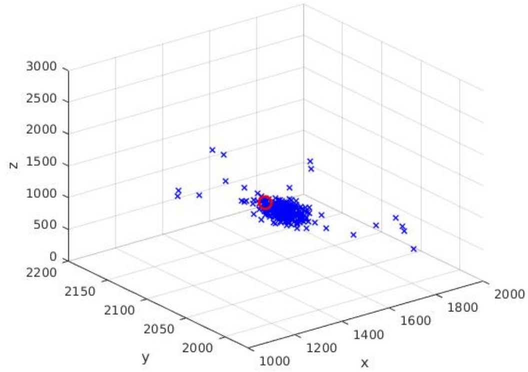


Figure 3.5: The test scenario for NLOS experiment

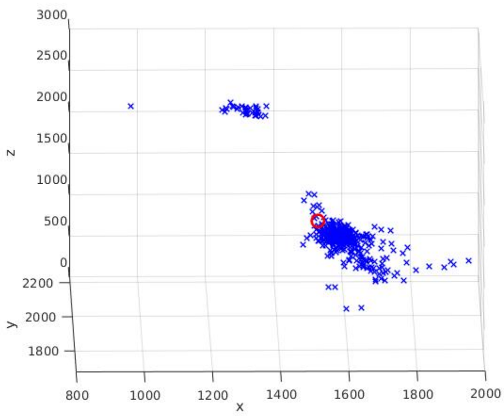
As it stands, NLOS is the major factor affecting the operation of the system and the accuracy of the raw readings. As such it is proposed that additional layers be added to procees this position data and try to improve the positional accuracy of the system and make it robust.



(a) Person at Position 1



(b) Person at Position 2



(c) Person at Position 3

Figure 3.6: Results obtained when an Anchor was lost.

3.3 Technical Design

At the end of section: 3.2 it was shown that using the raw position readings coming from the sensor is not feasbile in cases where NLOS is present. Furthermore, the overall goal is for the position of an indoor UAV to be recovered with a fair amount of accuracy since this forms the basis of any autonomous behaviour and navigation. As such it is desirable to use readings from multiple sensors but at the same time of reducing the overall physical footprint of the final system to keep it lightweight. With that in mind it is proposed to directly integrate the POZYX sensor system with a motherboard unit that is capable of doing sensor fusion, UAV control and communicating with external systems. In recent years FCU's have become versatile and robust and these requirements are easily met so in this section the technical parameters of the research will be discussed.

3.3.1 Flight Controller Unit

In order to prevent consuming too much time on dertermining a suitable flight system the Open-Source Hardware (OSH) community was consulted in order to find a suitable FCU for the research. Survey work done by Ebeid et al. (2018) present a qualitative analysis of several commercial hardware solutions. Table: 3.3 shows some of the major hardware considerations. All of the units have the standard UART, PWM and I2C interfaces in addition to other onboard sensors and interfaces. At the time of writing the Pixhawk series of FCU's are the most common and oldest systems. They have all the standard interfaces as well as several sensors allowing making them a keen choice for developers and researchers in a plug and play platform. Many in the series share the same intefaces with some of the smaller boards excluding some of the less popular interfaces. As such, to allow for flexibility in the technical design one of the Pixhawk family was desirable for this research. After deliberation and consulting the objectives and scope of this research it was determined that a basic Pixhawk would be satiable. Figure: 3.7 shows the board chosen, it was shipped quickly and it contains everything necessary to meet the objectives of the project allowing for quick prototyping and proof of concepts.

Platform	MCU	Sensors	Licenses	Interfaces
Pixhawk	STM32F427	b, m	BSD	c, s, a, pp, sb, ds
Pixhawk2	STM32F427	b, m	CC-BYSA-3.0	c, s, a, pp, sb, ds
PixRacer	STM32F427	b' m	CC-BY 4.0	c, pp, sb, ds
Pixhawk 3 Pro	STM32F427	b' m	CC-BY 4.0	c, s, pp, sb, ds
PX4 FMUv5 and v6	STM32F427	b' m	CC-BY 4.0	c, s, a, pp, sb, ds
CC3D	STM32F103	None	GPLv3	pp, ds, sb
APM 2.8	ATmega2560	b	GPLv3	pp, a
Erle-Brain 3	Raspberry Pi	b, m	CC BY-NC-SA	a
PXFmini	Raspberry Pi	b, m	CC BY-NC-SA	a

Table 3.3: Comparisons of various FCU's that are commercially available. Source: Ebeid et al. (2018) Page: 2.
b: barometer, m: magnetometer, p: pitot tube sensor, c: CAN, s: SPI, a: ADC, pp: PPM, sb: S.BUS, ds: DSM, da: DAC, x: XBEE, au: AUX, [d]: discontinued.

Furthermore, since the chosen board is OSH it has several options of firmware that can be used which makes code and software developed on this system extendable to other boards given they are able to run the firmware.

Figure 3.7: Radiolink Pixhawk used for the project

3.3.2 Flight Firmware

Now that a suitable FCU was chosen the next major step was determining a flight stack to run on the board. The major firmware options for the Pixhawk are either Ardupilot or PX4 stack. Both are well documented and have their advantages. Both support a large array of vehicles but the major differences come from the licenses they are under. Without delving into the technicalities of the licenses it is often summarised that PX4 is attractive to business owners who want to protect their property whilst Ardupilot pushes for changes to be put back into the main codebase. Additionally, from a quick brief and use of each of the firmwares, Ardupilot is slightly more documented due to its general public license and a bit more user friendly in terms of compilation and flashing firmware thanks to its pythonic based wrapper for compilation and uploading. Since both firmwares cover all the generic UAV types and there was no need for any niche systems Ardupilot was chosen as the primary codebase. It is worth noting that there is a subsection of the Ardupilot codebase is dedicated to Beacon based positioning systems which a previous Pozyx implementation is coded Ardupilot (2019).

3.3.3 Communication Interface (I2C vs Serial)

As mentioned in Chapter: 2 there is an implementation of using a Pozyx system in a GPS denied scenario Ardupilot (2019). This shows that it is possible to get the positioning data into the Ardupilot codebase and have it interacting with the onboard sensor fusion systems. A critique of this approach however is that it uses an additional Arduino Uno to pipe information from the Pozyx tag to the Pixhawk. Given the availability of the standard interfaces onboard the Pixhawk and the scope and objectives of this research project it is feasible that the Uno can be stubbed out of this experiment and data can be integrated directly onto the Pixhawk. This means the physical integration of the system can be summarised into the following steps:

- Choose a suitable interface.

- Place the code within a suitable section of the codebase.
- Utilise the parent class of the section of the Ardupilot to integrate the sensor.
- After testing the base functionality, integrate the new library within the background scheduler so it can run and feed measurements in the background to the copter main code.

An Arduino pozyxLabs (2020) and python library was originally designed by the Pozyx developers so it provided an algorithmic starting point for the functionality of the library being designed. The Arduino library is based around the I2C protocol whilst the python module uses the serial interface. As noted by the developers the python module is a port of the Arduino library and there is no plans to maintain the code and address bug fixes, as such the Arduino library was studied fo to build the new AP_Bacon extension. Furthermore, the I2C interface was designed onboard an embedded system so much of the flow can be ported effectively to the Pixhawk. Lastly, the previous implementation uses the UART/serial interface so that avenue has already been explored so in the intrest of quick prototyping the I2C interface was chosen. A positive note is that the existence of the prebuilt system allows data and experiments to be collected in lieu of development of this library since the information being provided is the same data generated by the Pozyx tag. The design of this library proves that the sensor data can be integrated directly whilst the pre-built systems, which are stable, can be used for data collection and experiments.

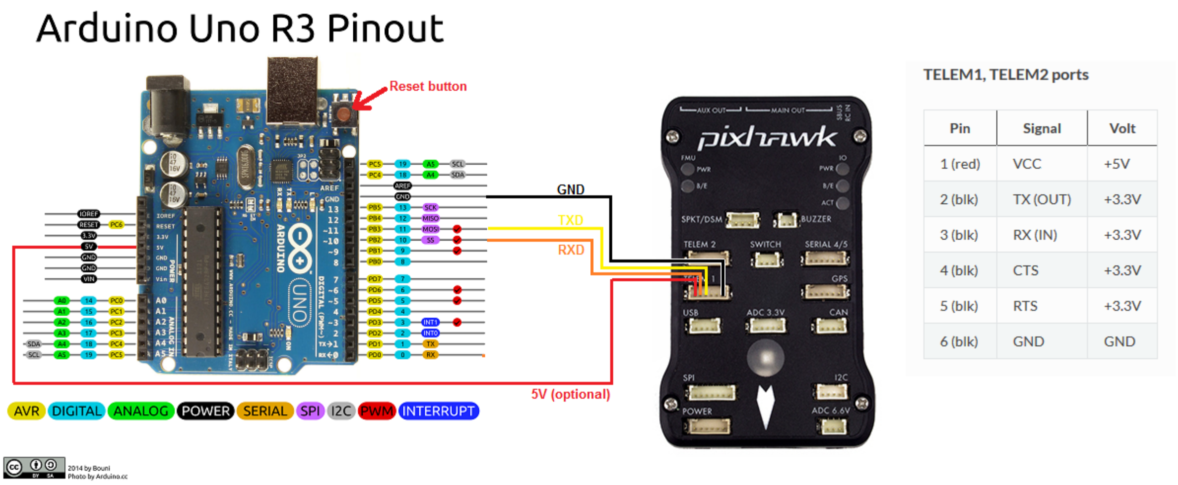


Figure 3.8: Non-GPS loiter solution provided by Ardupilot. Soruce:`https://ardupilot.org/copter/docs/common-pozyx.html`

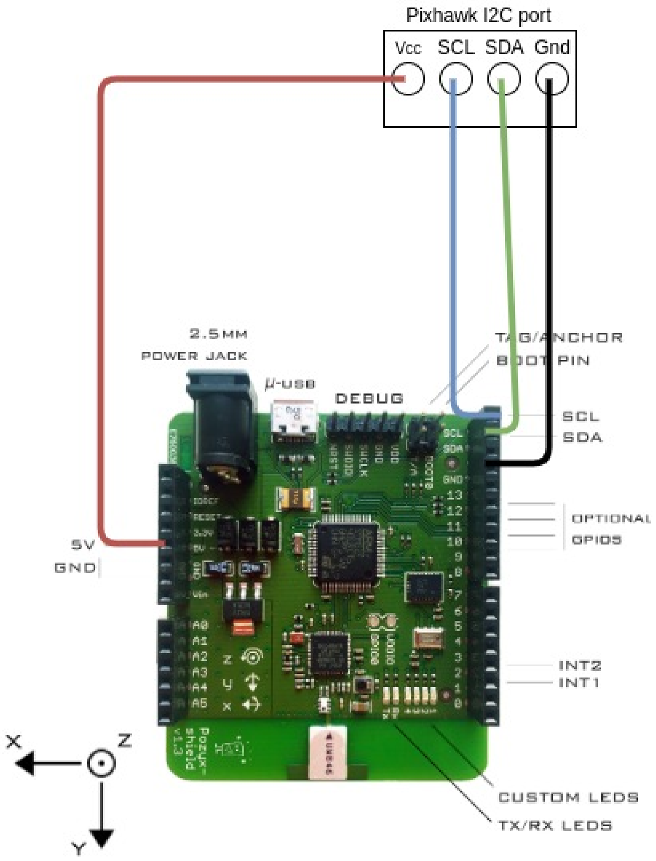


Figure 3.9: Proposed wiring of the I2C interface being developed

3.3.4 Sensor Fusion: EKF on Ardupilot

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Figure 3.10: System being unit tested in the environment.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu.

Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

APPENDIX A

Appendix A

Two Way Ranging (TWR)

TWR is a ranging method that utilises TOF and delays during transmission of a packet in order to determine the range between a tag and anchor. Figure A.1 provides a simple illustration of how this works. The distance for an individual tag and an anchor can be obtained by:

$$d = c.\frac{(TT2 - TT1) - (TA2 - TA2)}{2}$$

This is repeated for each anchor and then the position of the tag can be determined via trilateration. Geometrically, the position of the tag can be described as the point intersection of all the circles with distance,d , from the tag. This can be seen in Figure A.2.

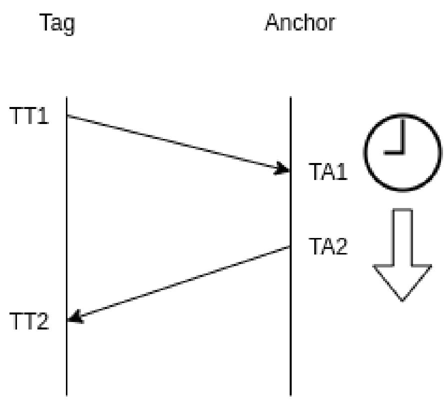


Figure A.1: Packet transfer in TWR.

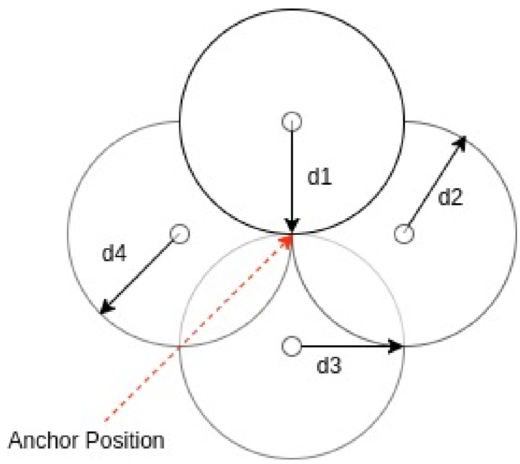


Figure A.2: 4 Anchor and 1 Tag trilateration in 2D.

Bibliography

Aerial Robotics IITK (n.d.), 'Visual inertial odometery', `https://aerial-robotics-iitk.gitbook.io/wiki/estimation/setup-with-vicon`.
Accessed : 2020-07-09.

Ardupilot (2019), 'Pozyx for non-gps navigation', `https://ardupilot.org/copter/docs/common-pozyx.html`.
Accessed: 2020-07-12.

Chadaporn, K., Baber, J. & Bakhtyar, M. (2014), Simple example of applying extended kalman filter.

Conceição, T., dos Santos, F. N., Costa, P. & Moreira, A. P. (2017), Robot localization system in a hard outdoor environment, *in* 'Iberian Robotics conference', Springer, pp. 215–227.

Custers, B. (2016), Drones here, there and everywhere introduction and overview, *in* 'The Future of Drone Use', Springer, pp. 3–20.

Deak, G., Curran, K. & Condell, J. (2012), 'A survey of active and passive indoor localisation systems', *Computer Communications* **35**(16), 1939–1954.

DeStefano, P. R., Siebert, C. & Widenhorn, R. (2019), 'Using a local positioning system to track 2d motion', *The Physics Teacher* **57**(7), 508–509.

Di Pietra, V., Dabove, P., Piras, M. & Lingua, A. (2019), Evaluation of positioning and ranging errors for uwb indoor applications., *in* 'IPIN (Short Papers/Work-in-Progress Papers)', pp. 227–234.

Ebeid, E., Skriver, M., Terkildsen, K. H., Jensen, K. & Schultz, U. P. (2018), 'A survey of open-source uav flight controllers and flight simulators', *Microprocessors and Microsystems* **61**, 11–20.

Jiménez, A. R. & Seco, F. (2017), Finding objects using uwb or ble localization technology: A museum-like use case, *in* '2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)', pp. 1–8.

Mendoza-Mendoza, J. A., Gonzalez-Villela, V., Sepulveda-Cervantes, G., Mendez-Martinez, M. & Sossa-Azuela, H. (2020), *Advanced Robotic Vehicles Programming*, Springer.

Mimoune, K., Ahriz, I. & Guillory, J. (2019), Evaluation and improvement of localization algorithms based on uwb pozyx system, *in* '2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)', pp. 1–5.

Pozyx Labs (2018), 'Pozyx accurate positioning'.

pozyxLabs (2020), 'Pozyx-arduino-library', `https://github.com/pozyxLabs/Pozyx-Arduino-library`, note = Acessed: 2020-07-12.

Tsai, C.-C. (1998), 'A localization system of a mobile robot by fusing dead-reckoning and ultrasonic measurements', *IEEE Transactions on Instrumentation and Measurement* **47**(5), 1399–1404.