

THE PERSE SCHOOL
CAMBRIDGE

Rouse Research Project

How effective is an automatic follow spot tracking system made using consumer parts?

Word Count: 1749

Acknowledgements

I would like to thank my Rouse research supervisor Dr North for his support throughout the project, as well as the PAC staff for their continued support and for kindly providing access to such state of the art equipment.

Contents

Acknowledgements **i**

1 Introduction **1**
1.1 Context . . . 1
1.2 Objectives . . . 3
1.3 Auditorium Infrastructure . . . 3

2 Product Design Research **4**
2.1 Positioning Technology . . . 4
2.1.1 RTLS Criteria . . . 4
2.1.2 Distance Estimation Techniques . . . 4
2.1.3 Position Estimation Techniques . . . 7
2.1.4 Local Positioning Technologies . . . 8
2.1.5 Conclusion . . . 8
2.2 Cartesian-Spherical coordinate conversion . . . 9
2.3 ETC Eos Lighting Desk Interface Protocols . . . 10
2.3.1 DMX-512 (Digital Multiplex) . . . 10
2.3.2 sACN (Streaming ACN) / Art-Net . . . 10
2.3.3 OSC (Open Sound Control) . . . 10
2.3.4 Conclusion . . . 10

3 Hardware Design **11**
3.1 Tracker and Anchors . . . 11
3.1.1 Radio Transceiver . . . 11
3.1.2 Micro-controller . . . 12
3.2 Server . . . 13

4 Software Design **14**
4.1 Firmware . . . 14
4.1.1 Anchor . . . 15
4.2 Backend . . . 15
4.3 Frontend . . . 16

5 Testing and Analysis **18**
5.1 Reliable, stable system . . . 18
5.2 Spacial deviation of ±20cm . . . 18
5.3 Low cost compared to other systems available . . . 18
5.4 Made with consumer parts only . . . 20
5.5 Integrate smoothly into a common system . . . 20
5.6 Work on any moving light . . . 20

5.7 No operator required 20
5.8 Simple User Interface (UI) 20

6 Conclusion **21**

Chapter 1

Introduction

1.1 Context

Technical lighting in all live event contexts is seeing a quick progression towards increasingly automated systems. The last 10 years have seen the widespread adoption of LED colour changing fixtures[1] [1], as well as more recently intelligent fixtures[2].

Intelligent fixtures are really versatile, allowing for its various parameters (e.g. focus[3], zoom, iris, gobo[4], prism) to be controlled and recalled remotely. They are incredibly valuable during the production process, saving a lot of time and rig[5] space not having to refocus[6] or add multiple conventional fixtures[7] - one fixture can be used to light various areas in lots of different ways. They also allow for much more complicated effects, enhancing the quality of live events. [2]

Followspots are a special fixture type, designed to allow an operator to manually track actors smoothly around the stage. They are mainly used to provide targeted front light, guiding the audience exactly where to look. [3]

Unfortunately, each followspot requires an individual operator to be physically controlling it at all times which limits the number locations they can be rigged[8], often resulting in unideal directions of light which create distracting and unflattering shadows.

Recently, larger venues have adopted automated systems to replace the traditional fol-

[1]Fixture: lighting instrument
[2]Intelligent fixture: light with the ability to change parameters other than its colour and intensity
[3]Focus (noun): the direction a light is pointing
[4]Gobo: metal cutout placed in front of the light to project a pattern on the stage
[5]Rig (noun): a structure above the stage and audience used to hold lights
[6]To focus (verb): the act of changing where a light is pointing
[7]Conventional fixtures: lighting fixture with only intensity controllable
[8]To rig (verb): the process of hanging, positioning and setting up a light

lowspot, with the first step being a remotely controlled operation system. Systems such as the Robe® Robospot™ allow for real-time tracking of performers, using a camera feed and joystick-based input mapped to a moving light[9]. Not only do these provide a much safer operation, they also allow lights to be rigged in inaccessible locations as well as control multiple different lighting fixtures at once, creating a more pleasing look whilst only highlighting the performer. [4]



Figure 1.1: Robe® Robospot™ remote follow spot system.

While such remote control systems are incredibly versatile, they still require an operator for each performer.

Completely automated systems remove the need for an operator, tracking performers without any external input required. Currently, the three main systems in use by venues are the FOLLOW-ME Track-iT [5], Zactrack® [6], and ClayPaky Buddylight [7] systems.

All consist of three key components (with slight variation in naming):

- **Server** - Rack-mountable computer, responsible for data computation, routing, and distribution
- **Anchors** - Located at known points around the venue
- **Tags** - Compact and lightweight, worn by the target

To perform the tracking, Ultra-Wideband (UWB) technology is used, which will be the method this project uses.

[9]Moving light: intelligent fixture with pan and tilt (moving) capabilities

The main barrier to entry with such products is their high price, so this project aims to create similar tracking system, at a competitive price.



Figure 1.2: Zactrack system, featuring (left to right): anchor, set tracker, actor trackers [6]

1.2 Objectives

To determine the success of the project and aid in the design, the following are the key criteria the system should meet.

1. Reliable, stable system
2. Spacial deviation of ±20cm
3. Low cost compared to other systems on the market
4. Made with consumer parts only
5. Integrate smoothly into a common theatrical system (see Section 1.3)
6. Work on any moving light
7. No operator required
8. Simple User Interface (UI)

1.3 Auditorium Infrastructure

- ETC Eos Family Lighting Console [8]
- Simple production network with a layer 3 switch[10]
- Patch panels[11] at various points around the building (include power, RJ45)

[10]Layer 3 switch: network switch which allows network segmentation into VLANs (Virtual Local Area Networks), QoS (Quality of Service) and intelligent packet forwarding

[11]Patch panels: connectors terminating in the server room allowing for routing between any two

Chapter 2

Product Design Research

2.1 Positioning Technology

This project requires a specialised positioning system[1] called a RTLS (real-time locating system), the core of the product being built. If the RTLS is unreliable or inaccurate, the errors will affect the whole project. [9]

2.1.1 RTLS Criteria

Factor	**Reason**
Area covered	Needs to be able to cover the entire venue
Accuracy	Target always needs to be lit
Environmental noise	Needs to cope with reflections off set elements and turbulence within the venue
Line of sight	Needs to be able to be hidden under clothing
Signal Attenuation	Needs to be reliable from behind clothing/set
Latency + update speed	Needs to keep up with the target

Table 2.1: Important RTLS factors

2.1.2 Distance Estimation Techniques

Distance estimation between the trackers and anchors is a key part of the system, being the main input to the position estimation algorithm. [11]

[1]Positioning system: determines the location of an object / person within a space

Time on Arrival (ToA)

ToA measures the time taken for a signal to travel from transmitter the receiver. Given that the signal travels at a known speed, the distance can be calculated as:

$$d = v \cdot t \tag{2.1}$$

where:

- d is the distance,
- v is the speed of the signal through air,
- t is the measured time of arrival.

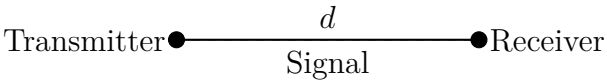


Figure 2.1: Time on Arrival (ToA) principle

ToA requires the clock on all devices to be synchronized - any errors in this will cause imprecise readings.

Time Difference on Arrival (TDoA)

TDoA relies on the difference in arrival times of a signal at multiple receivers. The equation for TDoA can be expressed as:

$$\Delta t = \frac{d_2 - d_1}{v} \tag{2.2}$$

where:

- Δt is the time difference,
- d_1 and d_2 are the distances to the two different receivers,
- t is the speed of the signal through air.

TDoA eliminates the need for synchronised clocks between anchors and trackers, however still requires for the anchors' clocks to be kept in time.

Received Signal Strength Indicator (RSSI)

RSSI estimates distance based on the strength of the received signal. The signal strength decreases with distance according to the path loss model [10]:

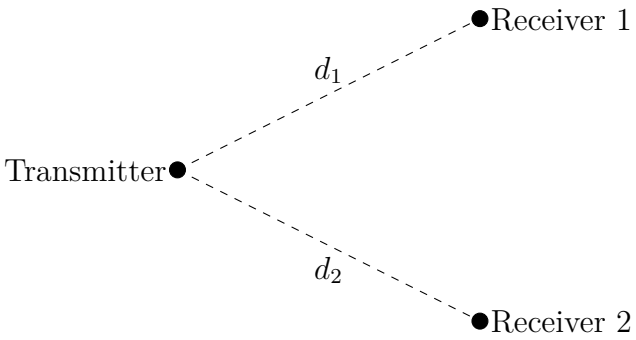


Figure 2.2: Time Difference on Arrival (TDoA) principle

$$P_r = P_t - 10n \log_{10}(d) \tag{2.3}$$

where:

- P_r is the received power,
- P_t is the transmitted power,
- n is the path loss exponent,
- d is the distance.

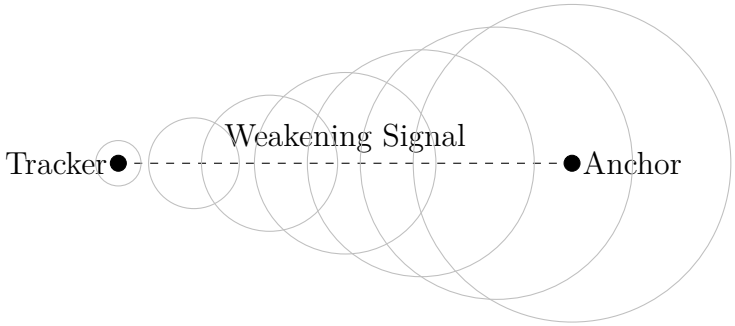


Figure 2.3: Received Signal Strength Indicator (RSSI) principle

Time of Flight (ToF)

ToF measures how long it takes for a signal to be sent from a single anchor to a tracker and returned. Given that the signal travels at a known speed, the distance can be calculated as:

$$d = \frac{v \cdot t}{2} \tag{2.4}$$

where:

- d is the distance,

- v is the speed of the signal,
- t is the measured time of the total round trip.

This method only relies on one device to perform the timing, which eliminates the errors introduced from having to synchronise clocks between multiple devices.

2.1.3 Position Estimation Techniques

Inertial Sensing / Dead Reckoning

Inertial sensing / dead reckoning uses data from accelerometers and gyroscopes to track changes in position over time.

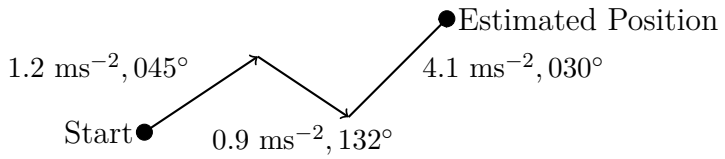


Figure 2.4: Dead Reckoning trajectory with accelerometer/gyroscope readings

In this method, errors accumulate over time, requiring periodic correction using external references.

Multilateration

Multilateration determines the position of the tracker based on at least three distance readings between the tracker and anchors around the tracking area. The possible locus[2] for the tracker based off a singular anchor can be given by the following formula:

$$d_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2} \tag{2.5}$$

where:

- d is the distance between the anchor and tracker
- (x_1, y_1) is the known centre of the anchor
- (x, y) are the possible locations for the tracker

You can then find the intersection between the three circles by setting all the equations equal to each other. This calculates the coordinates of the tracker.

[2]Locus: a curve or other shape made by all the points satisfying a particular equation of the relation between the coordinates, or by a point, line, or moving surface

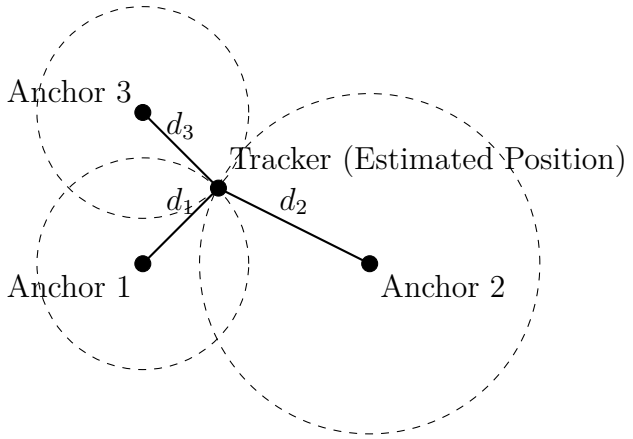


Figure 2.5: Multilateration with three transmitters

2.1.4 Local Positioning Technologies

This section outlines the various mediums available to which we can apply different position estimation technologies. [11]

Pros and cons

Technology	**Method**	**Pros**	**Cons**
Wi-Fi and Bluetooth	RSSI, time-of-flight, or angle-on-arrival from multiple access points	Simple to implement with pre-existing infrastructure	Low accuracy, negatively impacted by interference
Ultra-Wideband (UWB)	Short, high-frequency pulses and precise time-of-flight measurements	Very precise, low latency	Requires dedicated infrastructure, higher cost compared to others
Infrared (IR)	Camera measures position of IR signal	Cheap	Requires line-of-sight, sensitive to ambient lighting conditions
Acoustic (ultrasound)	Time-of-flight of sound waves between anchors and trackers		Sensitive to noise, temperature and air currents

Table 2.2: LPS Pros and Cons

2.1.5 Conclusion

Following the research into the various RTLS methods, it was decided to use multilateration via time of flight measurements from ultra-wideband.

Technology	**Accuracy**	**Reliability**	**Latency**	**Range**	**Cost**
Wi-Fi	2 - 10 m	Medium	Medium	20 - 50 m	Low
BLE	1 - 5m	Medium	Low	10 - 30 m	Low
UWB	10 - 30 cm	High	Very Low	10 - 30 m	Medium
Infrared	< 1 m	Low (LOS only)	Low	1 - 5 m	Very Low
Ultrasound	5 - 20 cm	Medium	Medium	3 - 10 m	Low

Table 2.3: Comparison of LPS Specifications

2.2 Cartesian-Spherical coordinate conversion

Moving lights are controlled using pan (horizontal rotation - azimuth) and tilt (vertical rotation - elevation) angles, which define a direction in 3D space. To calculate what the light is pointing at, these angles must be converted into a Cartesian direction vector.

Assuming a unit vector, the conversion from spherical coordinates to Cartesian coordinates is:

$$x = \cos(\mathrm{T}) \cdot \sin(\mathrm{P})$$
$$y = \sin(\mathrm{T})$$
$$z = \cos(\mathrm{T}) \cdot \cos(\mathrm{P})$$

where:

- P is the pan (elevation), measured from the z-axis to the xy-plane.
- T is the tilt (azimuth), measured from the horizontal xy-plane.

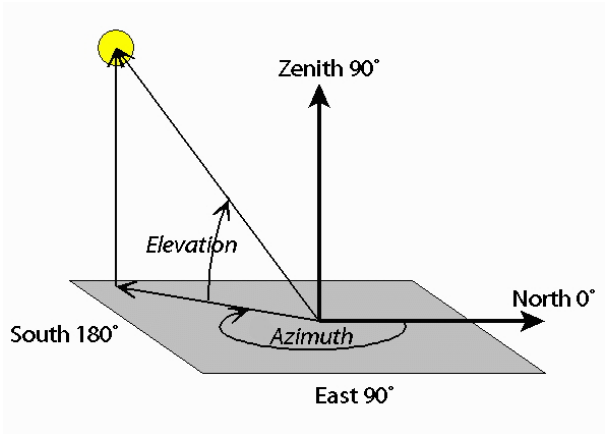


Figure 2.6: Pan/Tilt diagram

These vectors can then be intersected with the estimated position of the tracker resulting in the light pointing at the performer.

These values will however be calculated by the lighting desk, with the tracking system feeding in just the cartesian coordinates.

2.3 ETC Eos Lighting Desk Interface Protocols

The ETC Eos family consoles support a variety of control and communication protocols. [8] For the purpose of feeding real-time pan/tilt position data into the console—such as from a tracking system for automated followspot control—some methods are more suitable than others.

2.3.1 DMX-512 (Digital Multiplex)

Unidirectional serial communication protocol used for direct control of lights (outputted by the lighting console). A DMX merge could be used between the tracking system and lighting console, however would be very complicated to setup and difficult to troubleshoot. [12]

2.3.2 sACN (Streaming ACN) / Art-Net

sACN/Art-Net are protocols for transmitting DMX over a local area network (LAN). Has the same drawbacks as DMX above. [13]

2.3.3 OSC (Open Sound Control)

OSC allows control of any feature of the lighting console over a network. For example making a request to `/eos/chan/1/param/xfocus=5` sets the x value of channel 1 to 5 meters. [14]

OSC is very easy to interface with and commands are in plain English so easier to write and debug. [15]

2.3.4 Conclusion

It was decided to use OSC due to its ease to implement and the ability to directly control console features.

Chapter 3

Hardware Design

3.1 Tracker and Anchors

The trackers and anchors have two main hardware components, the micro-controller and the radio transceiver module.

3.1.1 Radio Transceiver

The radio transceiver is responsible for performing the ToF/TDoA measurements. The DW series of chips are well-regarded for their high accuracy UWB ranging capabilities and low power consumption. [16] The DWM1000 encapsulates the DW1000 chip with all the necessary RF components, handling the RF layout, power regulation and antenna design. [17] As such, the DWM1000 UWB module was chosen as the radio transceiver.

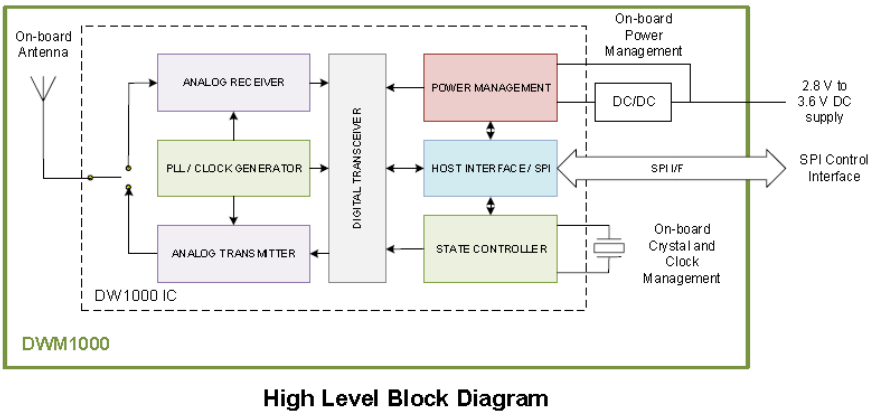


Figure 3.1: DWM1000 High Level Block Diagram [17]

The following pins are required to be connected to communicate with the DWM1000. [18][19]

Pin	Function
VCC	3V3 external supply
GND	Common ground
MISO	SPI[1] data output
MOSI	SPI data input
SCK	SPI clock
CS	SPI chip select
RST	Reset
IRQ	Interrupt request

Table 3.1: DWM1000 Pinout Requirements [17][18]

3.1.2 Micro-controller

Purpose

The micro-controllers are responsible for controlling the DWM1000 module as well as sending the distances to the server.

Tracker

For the trackers, the XIAO SAMD21 from Seeed Technology was chosen. [20]

- Very small - ideal for being worn
- Built in USB-C port for power and to flash firmware
- SPI port for interfacing with the DWM1000
- Works at 3.3V, the nominal voltage for the DWM1000
- Supports the Arduino devleopment platform, allowing for the use of a pre-written library to interface with the module



Figure 3.2: XIAO SAMD21 micro-controller from Seeed Technology [20]

Anchor

A different micro-controller was chosen for the anchors, the ESP32-POE from Olimex. This board shares very similar feature as the XIAO SAMD21, with the key addition of the ethernet port. This allows for fast communication with the server, minimising any latency. The ethernet port also has the added advantage of being able to supply POE[2]. [21]

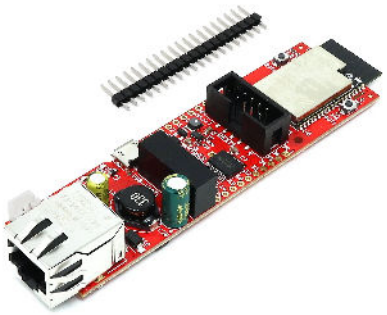


Figure 3.3: ESP32-POE from Olimex [21]

3.2 Server

For the server, a Raspberry Pi SBC[3] was chosen - a small, powerful, cost effective computer. It has ethernet connectivity for low latency communication, and allows for future expandability with additional input/outputs available. [22]

[2]POE: power over ethernet, with a compatible network switch ethernet and power is achieved over the same connection

[3]SBC: single-board computer



Figure 3.4: Raspberry Pi 5 [22]

Chapter 4

Software Design

4.1 Firmware

Firmware for both the trackers and anchors were written using the Arduino development platform. The `Arduino-DW1000` library by Thotro was adapted to interface with the DWM1000 module. [23]

An asymmetric ranging method was utilised to overcome the difference in processing time occuring as a result of heterogeneous micro-controllers being used for the trackers and anchors. [24]

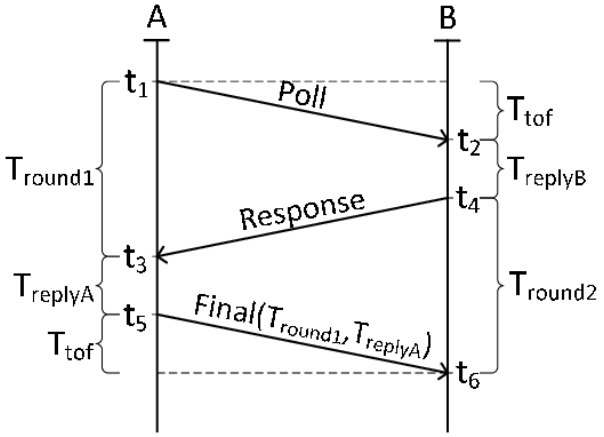


Figure 4.1: Asymmetric Double-Side Two-Way Ranging [25]

1. Tag (A) sends a poll message to the anchor (B) and records the transmit timestamp t_1
2. The anchor receives the poll and replies with a response message after a fixed delay time. The tag records the receive timestamp t_4 when it gets the response.

3. The tag sends a final message back to the anchor also sending t_{round1} and t_{replyA}. This then allows the anchor to calculate the T_{tof}

To allow for a single tag to range with multiple anchors, the `Arduino-DW1000` library was adapted to support MAC address filtering, with a round robin scheduling algorithm allotting each anchor a time slice.

4.1.1 Anchor

The anchors also had the added requirement of transmitting the distance readings back to the server. This was done via POST requests to the server, with the anchorId, tagId and distance being passed through as JSON[1]. [26]

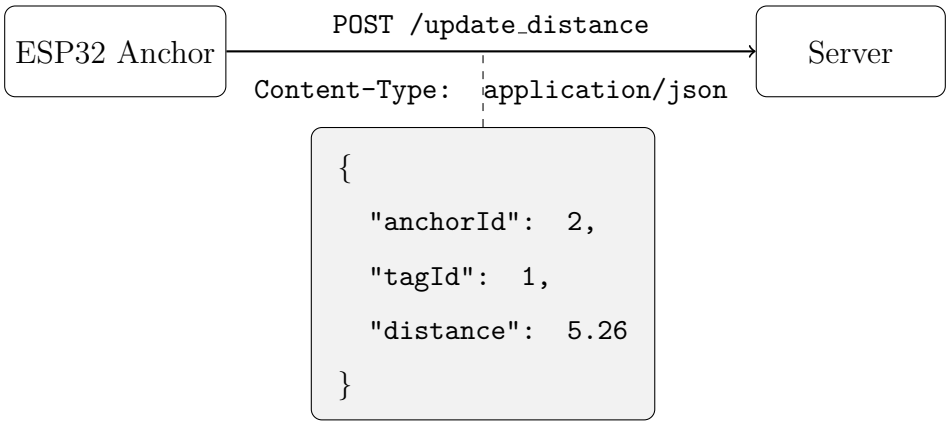


Figure 4.2: POST request to `/update_distance` endpoint with JSON data

Unfortunately, the ethernet module on the ESP32 used up the primary SPI bus, which required the `Arduino-DW1000` library (which only natively supports the default SPI pins) to be adapted to allow for custom SPI pins. [21]

4.2 Backend

The API[2] is being hosted via a simple Flask[3] server with only one route `/update_distance`. [27]

From there, it then calculates the estimated Cartesian coordinates of the trackers using trilaterion. Due to the slight noise in the distance readings, the least squares regression algorithm is used to find the best-fitting solution to the set of trilateration equations. It

[1] JSON: JavaScript Object Notation

[2] API: application programming interface, set of rules and specifications that enables different software applications to communicate and exchange data

[3] Flask: a small and lightweight Python web framework that provides useful tools and features that make creating web applications

does this by minimizing the sum of squared differences (residuals) between the loci of the equations and the predicted values. [28]

The DWM1000 module produces noisy distance readings, so to combat this, a Kalman filter is used to smooth out the readings. [29][30]

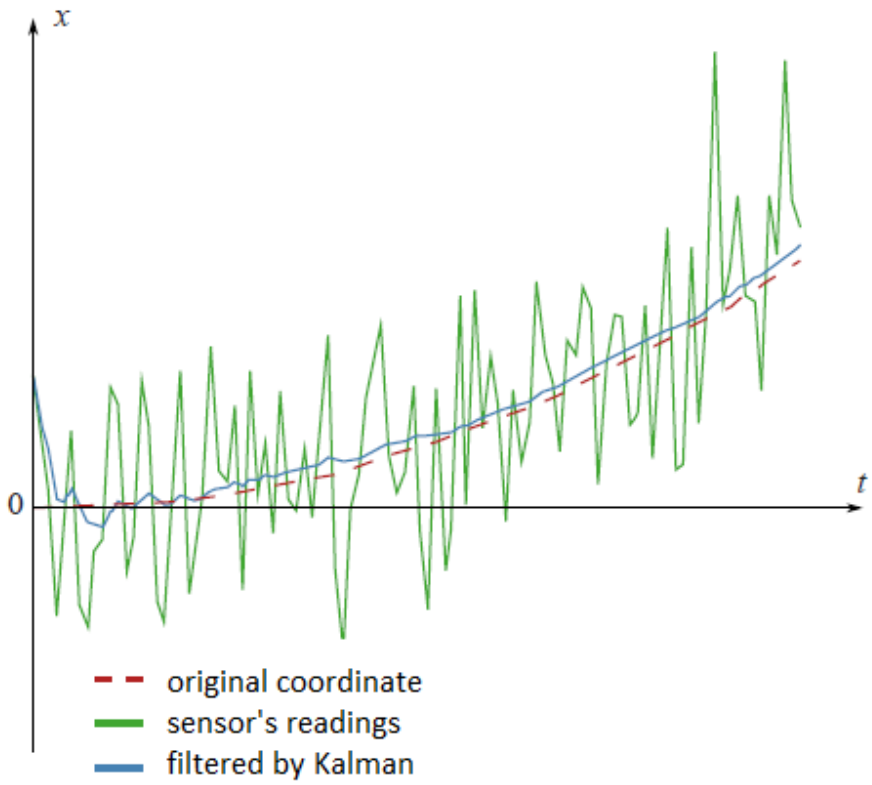


Figure 4.3: Kalman filter [31]

For communication with the lighting console, the package `python-osc` handles the network requests. [32]

Whilst the position data could just be directly applied to the desired moving light(s), a more versatile method is used by applying the position data to a SEM[4], then setting a FP[5] to follow the SEM. This way, the FP can be dynamically applied to any number of moving lights, and can be programmed into the existing cue playback system within Eos. [33]

4.3 Frontend

The frontend is an electron app which allows the user to monitor system status, view a real time visualiser for the trackers' locations and configure various parameters about the

[4]SEM: scenic element moveable, virtual channel within ETC Eos
[5]Focus palette: preset which holds moving light position data

system.

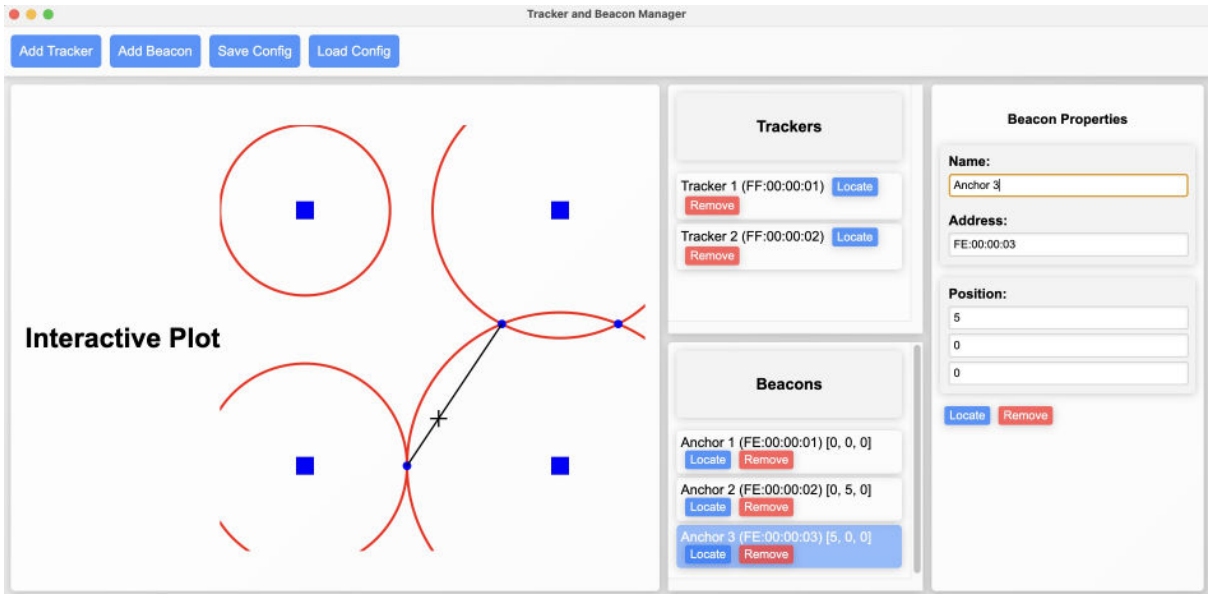


Figure 4.4: User Interface



Figure 4.5: ETC Eos Visualiser

Chapter 5

Testing and Analysis

To test and analyse the efficacy of the project, we will reference the original design criteria.

5.1 Reliable, stable system

Throughout testing, on the whole, the system was relatively reliable. Unfortunately the CPU in the anchor's ESP32 occasionally stalled, causing the device to restart. The restart process takes under a second, so only a slight pause is experienced. The round robin algorithm waits for the anchor to respond before polling the next anchor, so stalling in one causes all the others to pause distance readings. The algorithm could have been amended to move onto the next anchor after a time out.

5.2 Spacial deviation of ±20cm

To evaluate whether the system is always within ±20cm of the actual position, a series of distance measurements were taken at known locations. A graph has been plotted to show the results. The measured distances have had a constant offset applied to match the actual distance. As you can see, all the readings are within 20cm of the actual distance.

5.3 Low cost compared to other systems available

The following are the the BOMs[1] per tracker and anchor.

At a minimum, you require three anchors and one tracker. This comes out to a total of **£151.82**. This price is incredibly low compared to a Zactrack Pro [6] or similar systems,

[1]BOM: bill of material

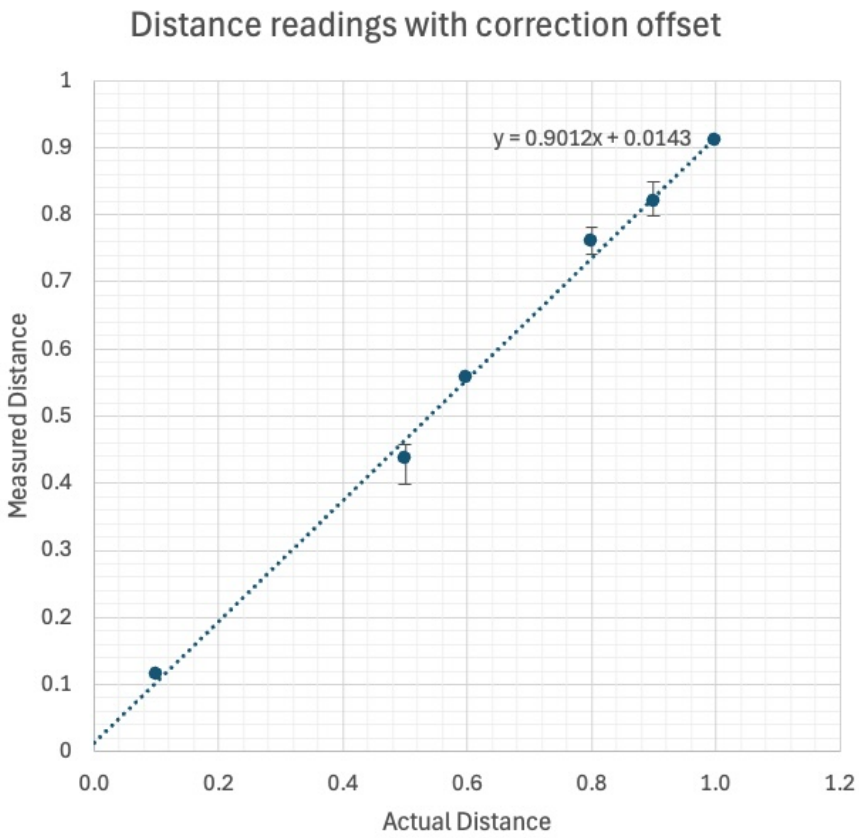


Figure 5.1: Distance vs Measure Distance

DWM1000	£21.38
Olimex ESP32-POE	£19.90
Total	£41.28

Table 5.1: BOM per anchor

DWM1000	£21.38
Seeed XIAO SAMD 21	£6.60
Total	£27.98

Table 5.2: BOM per tracker

which cost around £700 per tracker and £5000 per anchor, with large additional costs for the server.

5.4 Made with consumer parts only

The parts used in the project are widely available, such as from DigiKey[2].

5.5 Integrate smoothly into a common system

The project smoothly integrates with the common system described in Section 1.3, harnessing the pre-existing cue system.

5.6 Work on any moving light

Provided the moving light contains FPE[3] data within ETC Eos, the tracker can be applied to any moving light.

5.7 No operator required

Following the initial setup, no operator is required. The tracker mappings follow the existing cue system.

5.8 Simple User Interface (UI)

The user interface is very visual and self-explanatory, being simple to use. This makes it appropriate for users with any level of expertise.

[2]DigiKey: one of the largest electronics components distributors
[3]FPE: fixture position estimation

Chapter 6

Conclusion

In conclusion, the effectiveness of an automatic follow spot tracking system can be described as quite effective. Whilst the issues in reliability can cause the beam on fast moving targets to slightly deviate, the majority of the time the target is fully in the light.

In the future, the project would benefit from improving its reliability (e.g. prevent the CPU from stalling), as well as the potential for adding more than 3 anchors for redundancy and to improve accuracy. Whilst the system was only tested with one tracker, multiple trackers could be added to track multiple targets simultaneously.

Bibliography

[1] 'Lighting with LEDs – Theatrecrafts.Com'. Accessed 13 April 2025. https://www.theatrecrafts.com/pages/home/topics/lighting/lighting-with-leds/.

[2] Lav Studio — Intelligent light management: Key features of automated stage lighting fixtures. 'Intelligent Lighting Fixtures — LAV Studio — Moving Heads', 25 August 2023. https://www.lav-studio.hr/en/blog/intelligent-light-management-key-features-of-automated-stage-lighting-fixtures/.

[3] 'Followspotting Tips and Tricks – Theatrecrafts.Com'. Accessed 13 April 2025. https://www.theatrecrafts.com/pages/home/topics/lighting/followspotting-tips-tricks/.

[4] ROBE Lighting. 'RoboSpot™ — Follow Spot Systems'. Accessed 12 April 2025. //www.robe.cz/robospot.

[5] Follow-Me. 'Manual and Automated Performer Tracking in One Scalable System'. Accessed 10 April 2025. https://follow-me.com/.

[6] zactrack. 'Automated Follow System — Zactrack Lighting Technologies GmbH — Wien'. Accessed 10 April 2025. https://www.zactrack.com.

[7] Claypaky. 'Buddylight', 25 January 2024. https://www.claypaky.it/products/buddylight/.

[8] 'Eos Family Consoles'. Accessed 20 Jan 2025. https://www.etcconnect.com/Products/Consoles/Eos-Consoles/.

[9] Zebra Technologies. 'What Is RTLS? — An Overview of Real-Time Locating Systems — Zebra'. Accessed 20 Jan 2025. https://www.zebra.com/content/zebra1/gb/en/resource-library/faq/what-is-rtls.html.

[10] Tun, Phyo Thu Zar. 'PATH LOSS PREDICTION BY USING RSSI VALUES'. Accessed 9 May 2025. http://www.ijciras.com/PublishedPaper/IJCIRAS1056.pdf

[11] Aerts, Simon, and Joost Poelmans. 'Person tracking with a DMX-control system on FPGA for stage lighting and entertainment industry', 2020. https://documentserver.uhasselt.be//handle/1942/32035.

[12] 'What Is DMX? DMX 512 for Lighting Explained'. Accessed 8 May 2025. https://www.learnstagelighting.com/blog/what-is-dmx-512.

[13] Betopper. 'Art-Net and sACN [Full Guide]'. Accessed 8 May 2025. https://betopperdj.com/blogs/news/art-net-and-sacn.

[14] 'Eos Family Show Control User Manual'. Accessed 14 Oct 2024. https://media.musson.com/mti/docs/e/o/eosfamily$_s howcontrol_u serguide_r evc.pdf$.

[15] 'Open Sound Control', 13 August 2021. https://ccrma.stanford.edu/groups/osc/index.html.

[16] 'DW1000 Datasheet'. Accessed 15 Oct 2024. https://www.qorvo.com/products/d/da007946.

[17] 'DWM1000 Datasheet'. Accessed 15 Oct 2024. https://www.qorvo.com/products/d/da007948.

[18] 'Interfacing DWM1000 Positioning Module with Arduino'. Accessed 16 Oct 2024. https://electropeak.com/learn/interfacing-dwm1000-positioning-module-with-arduino/.

[19] 'Serial Peripheral Interface (SPI) - SparkFun Learn'. Accessed 8 May 2025. https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all.

[20] 'Getting Started with Seeed Studio XIAO SAMD21 — Seeed Studio Wiki', 11 January 2023. https://wiki.seeedstudio.com/Seeeduino-XIAO/.

[21] 'ESP32-POE - Open Source Hardware Board'. Olimex. Accessed 8 May 2025. https://www.olimex.com/Products/IoT/ESP32/ESP32-POE/open-source-hardware.

[22] Ltd, Raspberry Pi. 'Raspberry Pi 5'. Raspberry Pi. Accessed 8 May 2025. https://www.raspberrypi.com/products/raspberry-pi-5/.

[23] Trojer, Thomas. 'Thotro/Arduino-Dw1000'. C++, 9 May 2025. https://github.com/thotro/arduino-dw1000.

[24] IEEE. 'Text for relative positioning and location'. 14 May 2015. https://mentor.ieee.org/802.15/dcn/15/15-15-0429-00-0008-text-for-relative-positioning-and-location.docx

[25] ResearchGate. 'Fig. 1. Asymmetric Double Sided Two-Way Ranging Method'. Accessed 8 May 2025. https://www.researchgate.net/figure/Asymmetric-double-sided-two-way-ranging-method$_f ig1_3$39270612.

[26] ReqBin. 'How Do I Post JSON to the Server?', 16 January 2023. https://reqbin.com/req/4rwevrqh/post-json-example.

[27] Auth0 - Blog. 'Developing RESTful APIs with Python and Flask — Auth0'. Accessed 9 May 2025. https://auth0.com/blog/developing-restful-apis-with-python-and-flask/.

[28] 'Least Squares Regression'. Accessed 9 May 2025. https://www.mathsisfun.com/data/least-squares-regression.html.

[29] Lacey, Tony. 'Tutorial: The Kalman Filter'. Accessed 9 May 2025. https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf

[30] GeeksforGeeks. 'Kalman Filter in Python', Accessed 9 May 2025. https://www.geeksforgeeks.org/kalman-filter-in-python/.

[31] 'Kalman Filter Diagram'. Accessed 9 May 2025. https://i.sstatic.net/tjCcN.png.

[32] 'Python-Osc: Open Sound Control Server and Client Implementations in Pure Python'. Python. Accessed 9 May 2025. https://pypi.org/project/python-osc/

[33] Electronic Theatre Controls Inc. 'Controlling a Scenic Object in Eos', 22 July 2020. https://support.etcconnect.com/ETC/Consoles/Augment3d/Controlling_a_Scenic_Object_in_Eos.