Evaluating different localization methods for robotic swarming

Bachelor End Thesis

Sven Dukker Melle Minten

Delft University of Technology

TU Delft

Evaluating different localization methods for robotic swarming

by

Sven Dukker Melle Minten

Student Name	Student Number
Sven Dukker	4953517
Melle Minten	4909526

EEMCS Supervisor:	Dr. R.R. Venkatesha Prasad,
Daily Supervisor:	Dr. A. Simha
Project Duration:	19/04/2022 - 13/06/2022
Faculty:	Faculty of Electrical Engineering, Mathematics & Computer Science
Department:	BSc Electrical Engineering

TU Delft

Abstract

Localization is an important element for a moving swarm of robots. A swarm contains many individuals, and it is essential that the swarm members do not collide. Expensive and complex implementations to localize others is undesirable. Hence, a localization system designed using the available resources is desired. The available techniques that have been analysed are Received Signal Strength Indicator (RSSI), Time difference of arrival (TDOA) using the ESP-NOW protocol and a spinning time of flight (TOF) sensor. The spinning TOF sensor showed to be the most promising, with close range distance detection only containing a maximum errors of 5cm. The sensor is implemented as a cheap LiDAR system by mounting it to the front of the robot, which spins around its own axis. An IMU is responsible for keeping track of the orientation of the TOF sensor. Furthermore, two small algorithms were designed and compared in order to process the TOF data.

Preface

The subject of this thesis is creating a localization technique with a poor man solutions which is suitable for flocking behavior in a robot swarm. The thesis is written by Sven Dukker and Melle Minten, two electrical engineering students studying at the Delft University of Technology. We want to thank our supervisors dr.ir. Ashutosh Simha and ir. Suryansh Sharma as well as our head supervisor prof. Venkatesha Prasad for their support and council during the project. We also want to thank MSc David Zwart for delivering the needed hardware at the start of the project. furthermore, we wantto thank dr.ir. Chris Verhoeven for the extra guidance during the green light of the project. In the end, we would like to thank our colleagues: Ivar Hendrikson, Thijmen Hoenderboom, Jeroen van Uffelen and Lars de Kroon whom we had a pleasant and productive time during the project.

Sven Dukker, Melle Minten
Delft, 2022

Contents

1 Introduction

In the current day and age, humans have the desire to achieve equal productivity whilst doing less manual work. Hence the rise of automation and robots emerged. One of the new inventions from this current century to fulfill this desire is the swarming of robotics: The collective movement of a group of individual task-performing robots, whilst avoiding collisions but still remaining together. The application of swarms or flocking behaviour in robotics becomes more and more present. Using drones is the most famous form of swarming. Swarming can moreover be found in agricultural and aquatic applications [1]. Some of the higher-end solutions still have limitations, such as flocking robots only being able to function in predetermined and bounded environments. This thesis is part of a Bachelor Graduation Project that aims to research multiple swarming methods and implement these as well.

1.1. Project Division

The project was divided into three sub-groups of two persons per group. When integrated together, the three sub-parts form the total flocking system.

The first part of the project, and the focus of this thesis, is localization. Localization aims to locate the members of the swarm, with an additional goal to also locate obstacles in the vicinity of the swarm. Using the location of other members, each individual member should be able to decide in which way it can and cannot head. Localization does not only aim to just locate other members of the swarm, but doing so without the usage of external measuring instruments such as beacons or a central radar system. Such an localization system is called decentralised.

The second part of communication. Communication is responsible for the exchange of information between swarm members. In larger swarms it is unnecessary as well as disadvantageous for each member to communicate with every other member in the swarm. Often it is only necessary to keep track, and therefor communicate with the relative neighbouring members. Consequently, communication also aims to identify which members in the swarm are neighbouring the current member and use this information to only communicate with these neighbouring swarm members. It lastly aims to build a platform in order to have wireless logging of the swarm network in order to observe the behaviour of the swarm which will be useful for observational purposes and debugging.

The third part of the project is control. The control group designs and controls the individual swarm members, based on the information that is provided by the communication- and localization sub-parts. It aims to design the cars and their controls in order for each swarm member to manoeuvre through a two-dimensional space. It furthermore aims to follow a simplified version of boids flocking algorithm [2]. Where each members tries to stay a certain distance away from their respective neighbours, while trying to reach a globally similar heading.

1.2. Problem definition

As stated in the previous section, this thesis specifically focuses on the localization of flock members as well as the detection of obstacles which the flock should avoid. This is essential in order for the sub-group that does the the control to steer each flock-member in the correct direction, without colliding. The main objective of this thesis is to investigate in different localization techniques to find the best to implement for this project.

However not all localization techniques can be investigated; The selection of possible methods is limited to the available hardware. Hence this thesis searches for the best method of localization within these limitations of the project.

1.3. Situation Assessment

The enormous research field of localization is in continuous motion. As chips grow smaller and cheaper, the availability of higher end localization technology keeps improving over time. One of more dominantly used method for localization is the global positioning system (GPS). This method can achieve high accuracy in most applications. However, GPS is relatively expensive and the efficiency of the GPS trackers is heavily influenced by their surroundings: For instances large buildings or an indoor environment [3] [4].

Another method of localization is the usage of Received Signal Strength Indication, or RSSI for short. This method uses either a WiFi or Bluetooth signal strength relative to an access point (AP) in order to determine the physical distance between the AP and the receiver[5] [6] [7]. It was further elaborated that the RSSI method could be improved when there were multiple access points to connect with and therefore more accurately determine the location [8]. However, RSSI still shows large uncertainties in its distance approximation due to relative orientation of the receiver and noise reflections [9] [8].

The "Time Difference of Arrivals" method (TDOA) is the simple concept of measuring the time between sending and receiving a signal in order to estimate distance. It is often used with Ultra Wide-Band (UWB) frequency signals [10] [11] [12]. This technique often uses multiple beacons and trackers in order to locate individual objects, which limits the work area of potential swarm implementations. [13] [14] [15].

A more complex localization method is the usage of LiDAR. With LiDAR, it is not possible to detect other members of the swarm, but it can be used to map its surroundings. It has also been proven that LiDAR implementations can be used to navigate an area rather than to map it [16] [17]. A similar approach can be found using simpler hardware, such as time of flight sensors [18].

2 Programme of Requirements

The assignment provided by the supervisor stated that the subgroups should work together to build a swarm that can drive in an formation. However during the project some plans where altered. Instead of a platooning robot swarm, the project became more about flocking robots then platooning robots. The assignment for the localization subgroup is to investigate in different localization techniques that can be used for a flocking swarm. From these different techniques one should be chosen and implemented in a swarming group of robots.

Mandatory Functional Requirements

This elaborates on the actions that the robots must be able to perform in order to successfully design the localization method.

- The robots must behave like a swarm, in the sense that they remain together while avoiding collisions.
- The robots must be able to determine the angle and distance to other robots in their surroundings.
- The amount of robots in the swarm must be scalable without any software or hardware changes to the robot itself.
- The robots must measure obstacles and other robots with an maximum accuracy error of 20 cm.
- The robots must have a detection range of at least one meter, relative to the robot itself.

Mandatory Non-Functional Requirements

These are the requirements that sketch the essential contours in which the project must be confined.

- The robots must be designed for an indoor environment.
- The robots must be designed for two-dimensional applications.
- The robots must be designed using only the following hardware:
  - Robot chasis: Joy-IT ROBOT CAR KIT 4WD
  - Micro-controller: ESP32-DevKit-V1 with an ESP32-WROOM-32 CPU
  - Inertial measurement unit: MPU9250
  - Time of flight sensor: VL53L0X

Trade-off Requirements

The trade-off requirements indicate what design aspects are aimed to reach, but with the knowledge that all aspects can realistily never exist.

- The measurement rate should be fast enough for the respective implementation, with an lower bound limit of 10 Hz.
- Minimize the error in localization method.

Assumptions

It is furthermore assumed that the conditions that these robots are implemented in have a certain standard in order to simplify this first design.

- The robots are non-holomonic
- The robot is programmed on an ESP32-board written in C programming language, using the Espressif framework.
- The robots operate on a flat, horizontal surface.
- The potential objects within the operating area are always clearly detectable within the 2D-plane, and such the robots do not have to account for ledges or obstacles that are hard to detect like wired fencing.

3

Localization Techniques

In this chapter multiple localization techniques will be discussed that were considered during the bachelor graduation project for the localization of the robots. There are many different techniques for localization. It was kept in mind that the solution had to be efficient while being low in costs. The methods that were tested and their respective results can be found in this chapter. During the whole project the robot that is used got upgraded with some techniques. In Appendix A the robot that is used during this project is shown.

3.1. Received Signal Strength Indication

This localization method is based on a beacon device which sends out a WiFi or Bluetooth signal. Another device searches for this signal and measures the signal strength in dBm. Based on this received signal strength a relative distance between the two devices can be established. This method was researched for this paper as the costs of a transmitter and/or receiver are low in price (sub 10 euros) and the modules are already integrated into the ESP32-WROOM development boards that are used for this project. Hence it is easy to test and will be easy to implement for other projects in the future. In one of the papers the estimated localization had an average error of 0.427 meter [8]. This error margin is for this project not sufficient. However in this specific article the researchers used beacons. In this project the robots that are tracked can be a receiver and a transmitter simultaneously. The hypothesis is that this could improve the accuracy of the RSSI distance estimation method.

In order to calculate the distance between the receiver and the transmitter, a linear relationship between the observed intensity and a reference intensity was assumed [19]. The equation for this calculation is depicted in Equation 3.1. Where D = distance [m], C = reference power [dBm], RSSI = the measured RSSI value [dBm] and N = an environmental constant. The reference power and the environmental constant have to be determined beforehand, and can differ per location and/or situation. Furthermore, it is possible to use the Law of Cosines in order to estimate the relative angle of another car, based on the measured distances[20]. In order to correctly estimate the positions of all robots in the swarm, one requires at least 4 nodes to remove all uncertainties in this estimation.

$$D = 10^{\frac{C-RSSI}{10*N}} \tag{3.1}$$

Determining the constants

The first unknown to establish is the one-meter-constant (C). This is necessary to calibrate Equation 3.1 to the signal strength of the transmitter and the receiver's detection strength. To retrieve this value the following setup was established: Two ESP32 boards were placed on the same elevation, one meter apart and facing in the same heading. The first ESP32 board acted as a transmitter for the WiFi signal. The second ESP32 board measures the strength of the WiFi signal. A hundred measurements were done by the second ESP32 board, of which the data can be seen in Figure 3.1. A mode of -52 dBm is observed from this data, although there are frequent deviations. The standard deviation of this data set can be calculated to be 1.085 dBm. This margin is really promising because when converted into meters it equals σ = 0.074 m. This observation is further emphasised by the fact that the measured values are rounded off to full integers by the ESP32 board, hence small deviations in the RSSI measurements will result in larger sways in the standard deviation. After the one-meter constant was determined, the environmental constant (N) could be established. According to the theory [19] the value of N should be between 2 and 4. This value was based on multiple factors such as the amount of WiFi signals visible to the ESP32 chip, and the amount of metallic object in the test area that disturb the WiFi signals. Because the RSSI measurements were perform in a large, free space, the N-value was assumed to be 2. During the measurements and optimisation of the RSSI technique it was discovered that, in this study, the environmental value acted as a scaling factor for the converted meter value. It did not impact the RSSI method's performance in any way.

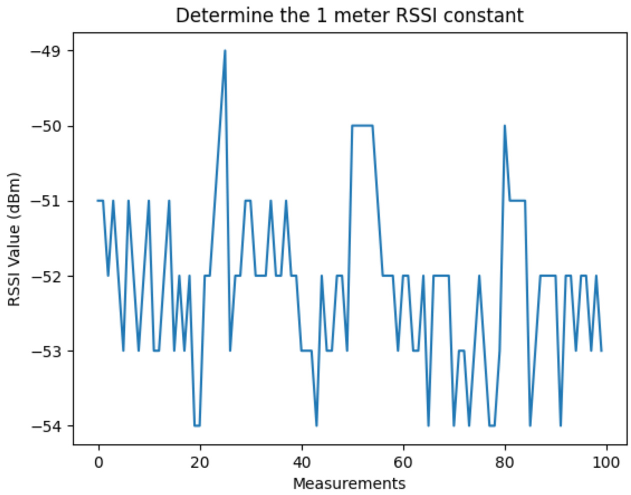

Figure 3.1: Plot which shows 100 measurements.

Results

The measured RSSI values from the performed tests were not behaving like the designed model. In Figure 3.2 it can be observed that the measured RSSI values are not in line with the model. The standard deviation (σ) of these measurements is smaller than 1.5 dB., hence the error is consistent. It is believed that this behaviour is caused by reflections. The dip observed between 0 and 0.5 meters can be explained by the orientation of the ESP-boards. The antenna is angled in such a way that when the cars are within the range of 0.5 meters, the antennas have difficulty in direct line-of-sight-communication. Hence the data in this range is dominated by unreliable reflected signals. Unfortunately when there is a lot of multipath-interference, the relation with measured RSSI and distance vanishes. Another downside that arose while testing this implementation, is that a single measurement of RSSI takes around 200ms. This is a very slow measuring rate, especially when the amount of robots in the swarm is increased. This is something that needs to be taken into consideration.

Determining a new model

The provided model, Equation(3.1), is not sufficient for this setup. Therefore a simple solution is introduced. Due to the small variance (σ) in the measurements there is some consistency. Three zones are introduced. The first zone is 0 to 0.5 meter, the second zone is 0.5 to 2 meter and the third zone is from 2 to 3 meter. In these zones separate models can be formulated. In Figure(3.2) the simple linear model (green line) can be seen, Equation(3.2). Where D is the distance predicted by the model and RSSI is the measured RSSI. It can be seen that the model is rough and only is valid within the 0.5 and 2 meters. Similar models could be made for the other zones, then the cars are restricted to stay in these zones. With smart predictions the models can be integrated with one another. However there are more localization systems that look more promising. Therefore only the second zone model is created. The model is tested on the robots. The robots drove behind one another while keeping 1.25 meter distance.

$$D = \frac{38 - RSSI}{11.63} \tag{3.2}$$

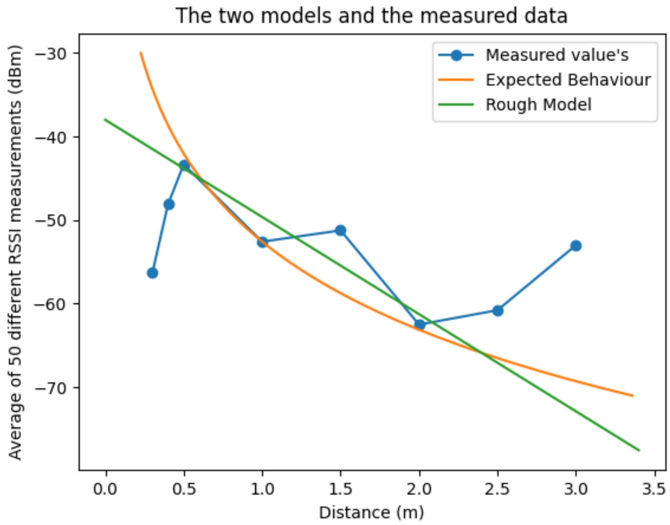

Figure 3.2: RSSI different models in comparison with the measured values.

3.2. Time Difference of Arrival

Time difference of arrival (TDOA) is a distance determination method that uses the difference between the transmission and receival of a signal to estimate the distance between the two antenna's. This method is not bound to a certain technique or hardware; it can have multiple implementations. In this thesis the ESP-NOW communication protocol was used, as it was designed by Espressif and hence readily available on the ESP32 board used in this project. ESP-NOW uses a 2.4 GHz low power wireless connectivity, but remains a separate protocol and does not use WiFi or Bluetooth. The implementation that is investigated is a "ping pong" protocol using two ESP32 boards. When the angles at which the other car is then located, can be calculated using the Law of Cosine [20].

Designing the protocol

The protocol works as follows: Two ESP32 boards are activated, one acts as a transmitter, the other as receiver. The transmitter sends an one byte message via ESP-NOW to the receiver. This receiver immediately returns an one byte message back to the transmitter to indicate that a message has been received. Once the transmitter receives this return signal, it uses the time difference between the originally transmitted- and return signal to determine the distance between the two chips. Equation(3.3) shows the formula for a vacuum. In the equation Δt is the time it takes from sending the ping signal and receiving the pong signal, C is the constant of light speed and d is the distance between the two chips. When the chips are 1 meter apart then Δt is equal to 7 ns. Where there are no reflections as it acts on the first signal, received from direct line of sight.

$$\Delta t = \frac{2d}{C} \tag{3.3}$$

Results

When the protocol was implemented it's performance was tested. The the ping-pong protocol was executed with a steady distance of one meter between the two ESP32 boards, while the response time was measured and logged. The ESP32 board was only able to determine the TDOA with an accuracy of micro seconds. The The results of this test can be seen in Figure(3.3). It can be seen that there is a nominal delay of around a thousand micro seconds, but there are also spikes presents of several thousands micro seconds. In order to get a better understanding of the basic behaviour of the measurements the spikes are filtered out. These filtered measurements can be seen in Figure 3.4. From this it is observed that the measurements from the tests at one meter still range from $718\mu s$ to $1054\mu s$. Using Equation 3.3, this consequently results a difference ranging from 215km to 315km. These results appear extremely far off the desired distance range of around 0.5 meters. In order to investigate whether a model-based approach, likewise to the RSSI, was possible, it was moreover performed multiple times at different differences. These results are shown in Appendix B. These results show that there is no correlation between the data and the distance over which the measurements were performed.

The CPU clock speed of the ESP32 board is 240 MHz at maximum. It was expected that there would be a constant delay of several nanoseconds due to processing, where one cycle takes 4.2ns. However this is not what can be seen in the results from the performed measurements. The results show a large variance between measurements. It is believed that this is due to the ESP32-WROOM-32 chip's internal processor, which performs non directly visible tasks within these clock cycles and hence produces unpredictable and largely varying delays in each individual measurement. Examples of such tasks could include f.i. the handling of internal flags and interrupts or the performing of queued tasks in the pipeline. It was tried to prioritise the TDOA task within the ESP32 chip itself, but due to disappointing results and the availability of alternatives methods it was not looked into further.

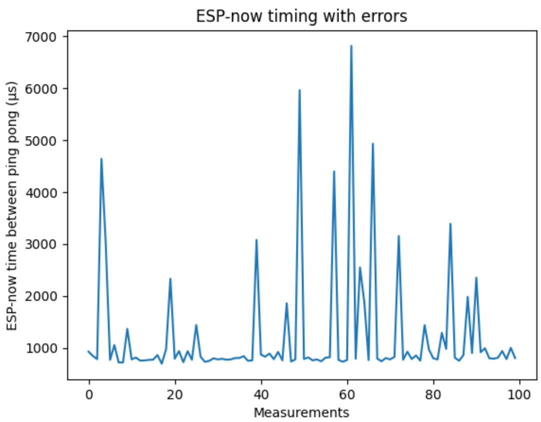

Figure 3.3: Δt when the chips are 1 meter from one another without the peaks filtered out.

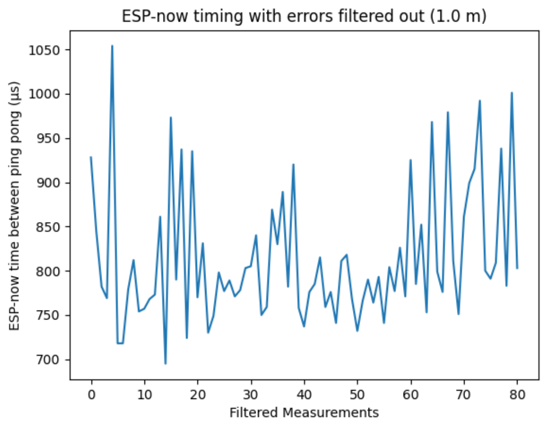

Figure 3.4: Δt when the chips are 1 meter from one another. Equal data as in Figure 3.2, but high spikes were filtered out

Additional Considerations

The TDOA method maintains promising for localization purposes. For instance using Ultra Wide Band (UWB) to broadcast the signal and having either a separate or faster processor available for the implementation of TDOA would greatly improve it's potential [21]. Moreover, the ESP-NOW protocol communicates using a peer-network. This network is limited to 6 ESP32 chips that are able to send and receive signals, where this ability is critical for the ping-pong protocol. This means that for the current implementation the swarm size would be limited to 6 robots. This may be able to be extended with other implementation, but that was not tested for this thesis.

3.3. Time of Flight

The last method that was researched and implemented for this thesis is the time of flight (TOF) sensor. This sensor uses the time difference between transmitting a infrared light signal and receiving its reflection. This is an external sensor and, as such, can only be connected to the ESP32 board using a I^2C bus. For this thesis the time of flight sensor that was used for the implementation and measurements was the 'VL53L0X', designed and produced by STMicroelectronics. This sensor is capable of measuring a distance from 20mm up to 1200mm, which includes the required range for this thesis. The accuracy in the data sheet is very promising, as the ranging error offset is, according to the data sheet, smaller than 3%. This means that at the maximum measuring distance of 1200mm would theoretically result in a maximum error of 36mm, which is better than the maximum acceptable error of 20cm (200mm). The sensor is capable of performing these measurements within roughly 33ms, which roughly equal 30 measurements per second. A downside of the sensor is that it is only capable to determine an one dimensional distance. However, the proposed solution to this issue in this thesis is to place the sensor on a vertically extended, rotating pillar. So that it will be able to scan a 360°, two dimensional surroundings using the TOF sensor.

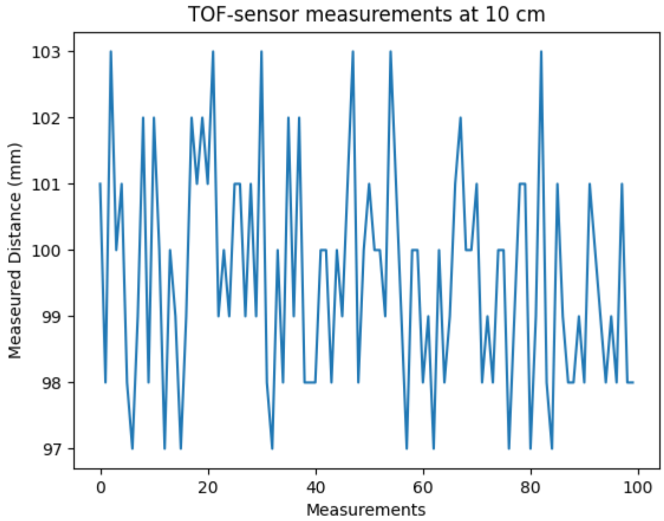

Figure 3.5: 100 measurements with a TOF with obstacle at 10 cm.

One Dimensional Performance

Before implementing the sensor for the 360° scan, the sensor was first tested stationary. The TOF was connected to the ESP32 board sensor via I^2C. It was then placed horizontally on the floor facing a wall, which was 10cm away from the sensor. The results showed inconsistencies due to the practical functioning of the sensor itself: In practise the sensor sends out a light signal in a cone shape with a 35° angle, opposed to a single one dimensional array. This meant the TOF sensor would sometime sense reflections from the floor instead of the wall it was facing, which is undesired behaviour. The sensor was then elevated from the floor, as it will be when mounted on the robot. Once elevated, the distance was measured again. The results of these measurements are depicted in Figure 3.5. In this figure it can be seen that the measured error margins are really small: The standard deviation (σ) is 1.6mm and the maximum deviation was 3mm. These deviation results are within the requirements for this project and hence showed real promise for implementation. This simple setup was then repeated at multiple distances, of which the results can be found in B.2. These results also show respectively similar results.

However, when the sensor was pointed toward the robot that were used in this project, see Figure(A.1), it sometimes failed to measure the slim profile of the robot's chassis. When the TOF sensor fails to detect a object within a 1.2m range, it returns a miss-value of 8191mm. The data from the TOF sensor trying to detect a robot at 90cm, is depicted in Figure 3.6. This data shows a frequent failure by the TOF to sense any object within a 1.2m distance. The proposed solution to this problem is to improve the detectability of the robot by constructing a larger shell around or on top of the robot. In this thesis, the shell was realised by mounting a paper sheet on the side to improve the reflection surface, as shown in Figure A.3. The TOF sensor was again faced towards to car, that now had an improved detectability. The results, as depicted in Figure 3.7, show that the TOF sensor detects an object at approximately 90cm without any failed measurements. Hence it can be concluded that the sensor is very accurate and is able to be implemented for localization.

Two Dimensional Scan

In order to detect obstacles and neighbouring robots inside a swarm, the robot needs to be able to detect it's 2D surroundings. The TOF sensor is only capable to detect in one direction. By implementing multiple sensors facing in different directions, the desired 2D result could be obtained. However this is not a low cost solution, as the TOF sensors are €15,- per unit. The proposed solution was to place the sensor on a rotating pillar, but this was in practise not obtainable for this thesis. Hence, a variation of this solution was implemented by making the robot spin with the TOF sensor horizontally mounted at the front of the robot, facing forward. Making the robot spin creates an additional challenge: The robot needs to know the heading in which a TOF measurement was taken, in order to accurately depict its surroundings. Therefor, an inertia measurement unit (IMU) was implemented on the robot. The IMU used for this project was the 'mpu9250', designed by InvenSense. This IMU contained a magnetometer, which provides the angle of the IMU by mea-

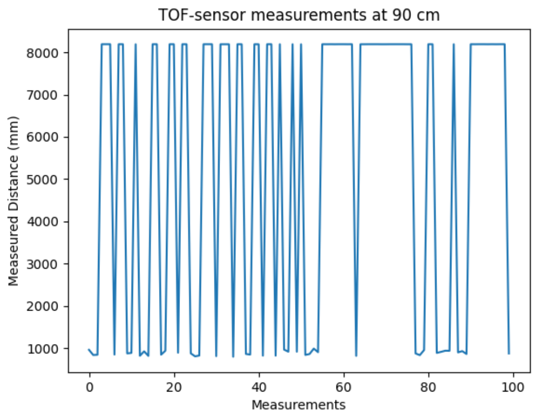

Figure 3.6: 100 TOF measurements of a robot at 90 cm. The 8191mm readings are misses of the TOF sensor

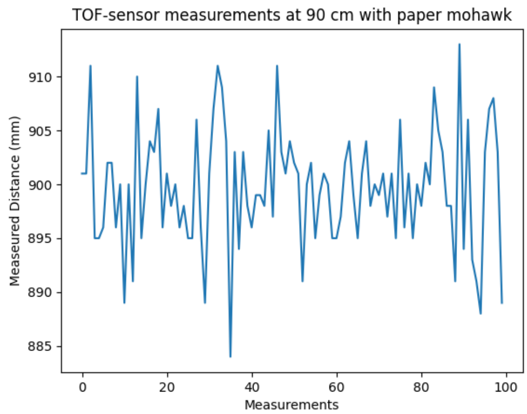

Figure 3.7: 100 TOF measurements of a robot at 90 cm, where the robot's profile is improved for detectability.

suring the magnetic field of the earth, likewise to a compass. This IMU was also connected to the ESP32 board via the I^2C bus.

Now the robot is able to scan it's surroundings by measuring the distance and corresponding heading whilst rotating. In order to test the robot's ability to scan it's surroundings, an object was placed at a 225° heading and a 450mm distance from the car. The results of the spinning robot's scan can be seen in Figure 3.8. In the figure it can be observed that there is an object between 160 and 270°at a distance of about 440 mm. This shows that the robot is able to correctly measure the distance and associate this with the correct heading. The last step for the robot in detecting the surroundings is to understand where a possible object might be located. This is done by creating an algorithm that is able to filter out false data and locates possible swarm members.

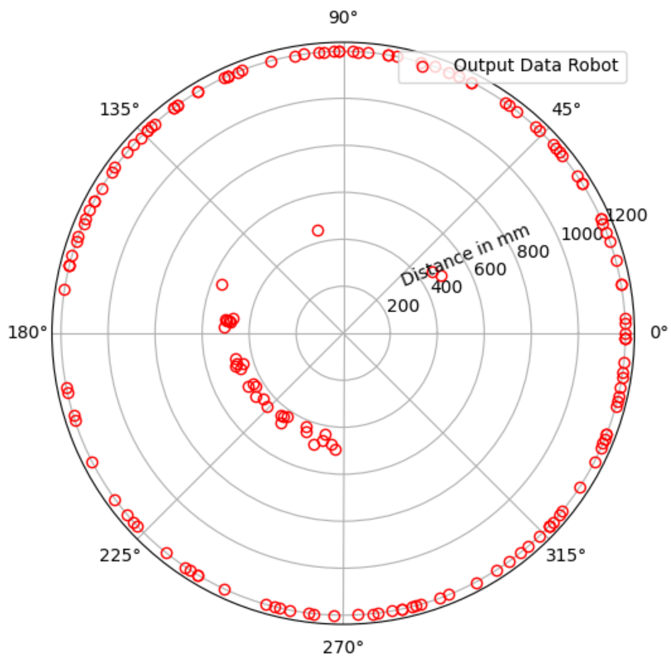

Figure 3.8: Spinning the TOF sensor with the IMU to produce a 360°surroundings of the robot. The red dots represent measurements. Measurements at 1200mm were in fact failed measurements that returned 8191mm, but capped for readability purposes.

2D Obstacle Detection Algorithm

Obstacle detection implementations such as the one required for this project, can be regarded as their own thesis subjects on their own. Nonetheless, it is still necessary for this localization method to implement. Two different, basic algorithms were made and tested for the purpose of this thesis. They will be evaluated in this thesis by using measurement data used was gathered using the real sensors on the car. The first algorithm that was tested was a simple differential filter. This algorithm filtered data based on the difference in distance with previous data. The steps of the algorithm can be found in Algorithm D.1. The output of this filter was then visualised so the results could be compared. These visualisations are shown in Figure 3.9 and 3.10. Figure 3.9 clearly shows that the wrong data points get filtered out so that a clear representation of the obstacle in the scan can be formed. The exact same algorithm was used on a different data set in Figure 3.10. However, here the algorithm shows quite a few errors in its approximation, due to concurrent faulty measurements.

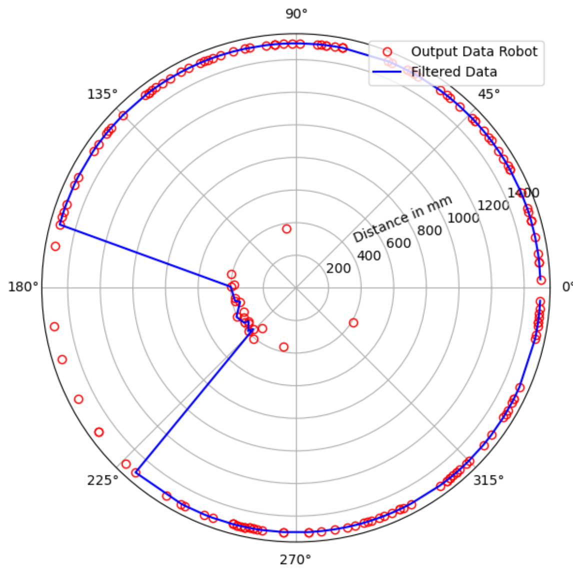

Figure 3.9: Differential filter algorithm on real data with good results.

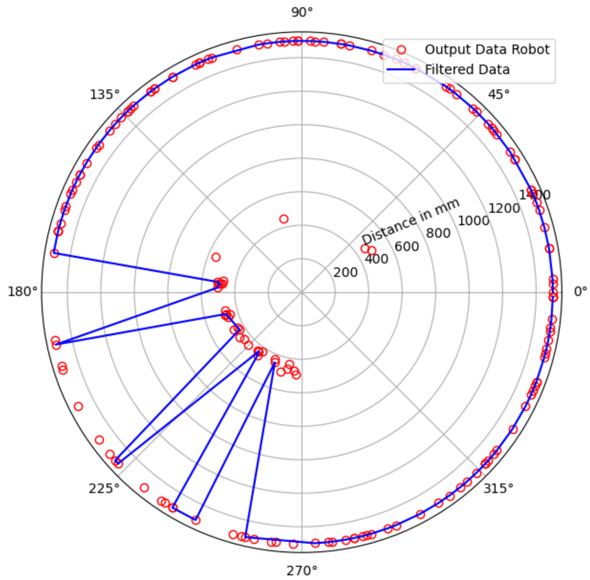

Figure 3.10: Differential filter algorithm on real data with bad results.

The second algorithm is based on a window that stay preserved when the average off all the values in that window stay above a certain threshold. The algorithm starts with checking whether the robot sees something or looks at nothing. When the robots see an object it adds an 1 to an array, when it sees nothing it adds an zero to this array. Then the average of the whole array is taken, see Equation 3.4.

$$\frac{\sum_{n=1}^{\text{Hits}} 1}{\text{Hits} + \text{Misses}} \tag{3.4}$$

Every time the average stays above the threshold, which is 0.5. The algorithm adds a 1 to the prediction array. This prediction array is then normalized to the average of the measured point. In this way the robot recognizes the whole obstacle, instead as lose pieces of an obstacle. The results can be seen in Figure 3.11 and Figure 3.12. In Algorithm D.2 The pseudo code of this algorithm can be found.

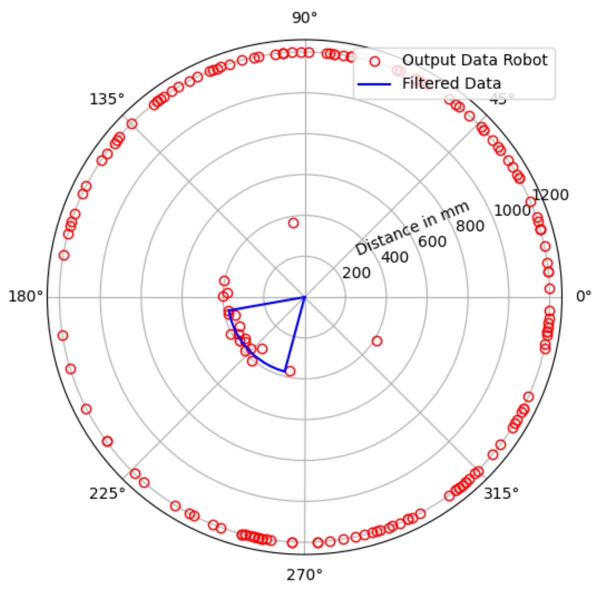

Figure 3.11: Result of the windowed version, with no error

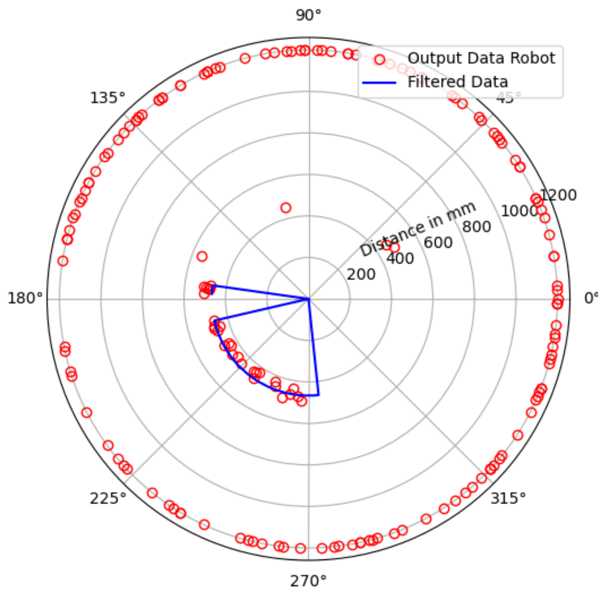

Figure 3.12: Result of the windowed version, which is improved relative to the derivative solution.

3.4. Result Comparison

This chapter contains a small representation of the measured values that resulted from tests performed in this thesis. The explanation of these values can be found in their respective chapters. This table only serves as a small recap of the obtained results.

Method	**Total Measurement Duration**	**Effective Range**	**Error in Effective Range**	**Starting Costs**
WiFi RSSI	200ms	0.5m-2.0m	0.02m-0.5m	€5,- to €10,-
TDOA (ESP-NOW)	1ms	0.01m to >10m	0.02m-0.5m	€9,-
Time of Flight	33ms	0.02m-1.2m	0.02m	€15,-

Table 3.1: Comparison table for the different methods and their accuracy- and costs considerations. The effective range was determined either by limitations of the sensor, or the data contained too much uncertainty to be considered usable.

4
Conclusion and Discussion

4.1. Conclusion

This thesis discusses multiple localization methods with the aim to find the best method within the contours of the project. From the tests and measurements performed in this thesis, the time of flight sensor is the only one out of the total of three methods that is able to perform high accuracy measurements, within the required range distance and with an sufficient measurement frequency. Therefor this method seems best for localization purposes.

4.2. Discussion

The results of the research for different localization techniques gave great insights. The problem definition of this thesis was to find a poor mans solution. Several solutions where investigated. The RSSI and the TDOA via ESPNOW didn't gave the results that where expected. The reason why the results weren't what was expected is not sufficient tested. Therefor it cannot be concluded that these techniques will never work.

The technique that is used for determining the localization of the robot is spinning with a TOF sensor on top of the robot. However it can be stated that the robot does not need a full 360 °image. It can be hypothesised that looking in a forward cone is enough to let the control algorithm work.

There are many other solutions to localize. The solutions tested in this report is a small sample of the many solutions. While the results of TOF are sufficient, there could be a solution out there that will perform better in comparison to TOF.

4.3. Future work

While the presented results comply with the requirements, there are a number of improvements that can be made in future projects. The first improvement is the algorithm used in the spinning TOF sensor to determine where obstacles are. The algorithm that is written for the this thesis can be more optimized. However, it can be concluded that the algorithm to determine obstacles could be a whole theses in itself. Another improvement is the robot vehicles. The wheels and chassis where already upgraded during this project, however to have better performance an upgrade in the mechanical hardware would be preferred.

In future works it could also benefit the driving speed of the robots when the they do not have to turn a full spin to localize and decide where to go. As told before it can be hypothesised that a full spin is not mandatory to let the robots drive in a flocking pattern.

There are many more solution to localization. F.i. another version of the ESP32 chip; the ESP32-S2, has different sensors which could be useful for localization. The ESP32-S2 chip has a Fine Timing Measurement (FTM) unit, measures the round trip time over WiFi. This works with the principles of TDOA, but it is purely made for approximating the distance, where ESP-NOW is designed to transfer data. In future works more of these systems could be tested and therefor considered as a poor mans solution.

A

Additional Images

Robot models

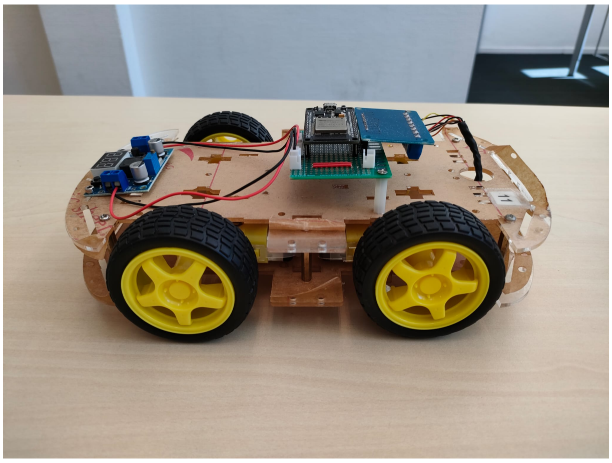

Figure A.1: Robot delivered original to us, hard to rotate.

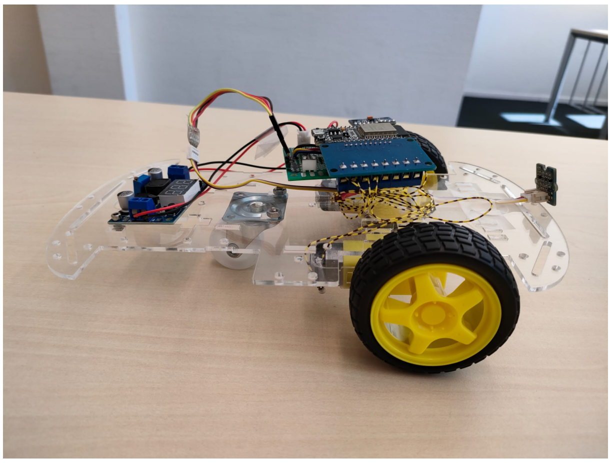

Figure A.2: Robot designed for spinning around its own axis.

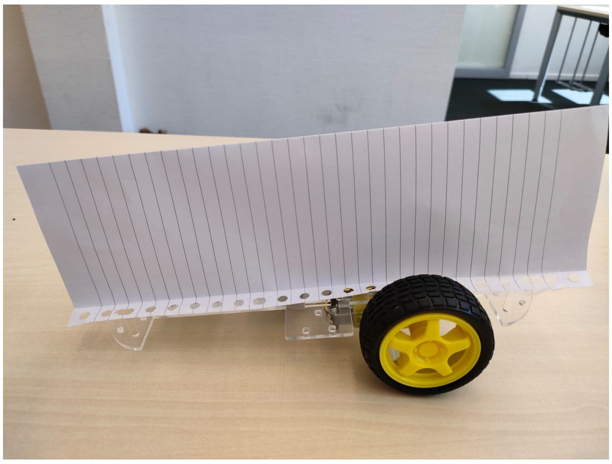

Figure A.3: The robot, mounted with a paper sheet in order to optimize detectability by the TOF sensor.

B

Additional Simulation Results

B.1. ESP-NOW

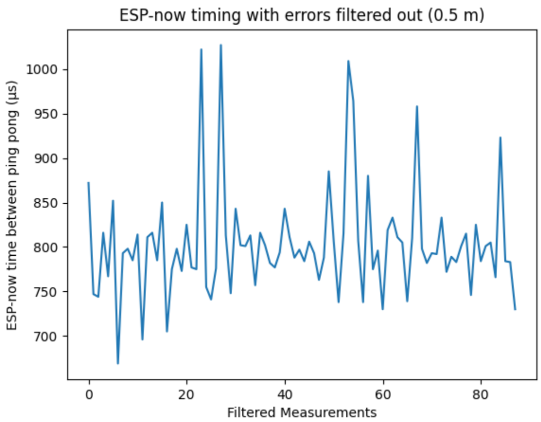

Figure B.1: Δt when the chips are 0.5 meter from one another.

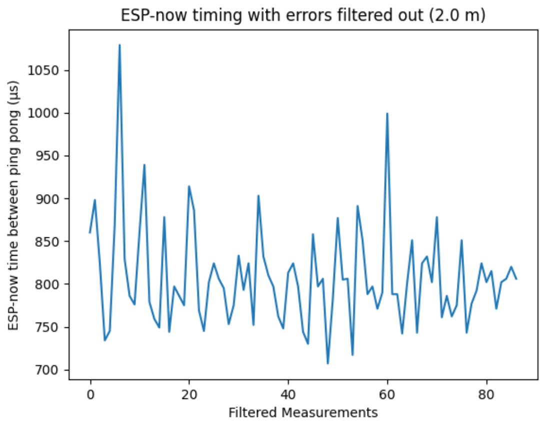

Figure B.2: Δt when the chips are 2 meter from one another.

B.2. TOF

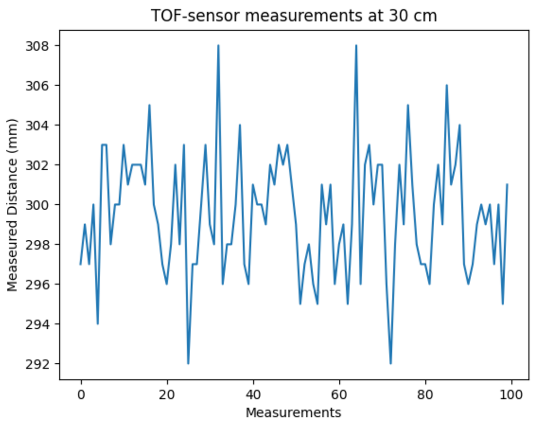

Figure B.3: 100 measurements with a TOF at 30 cm.

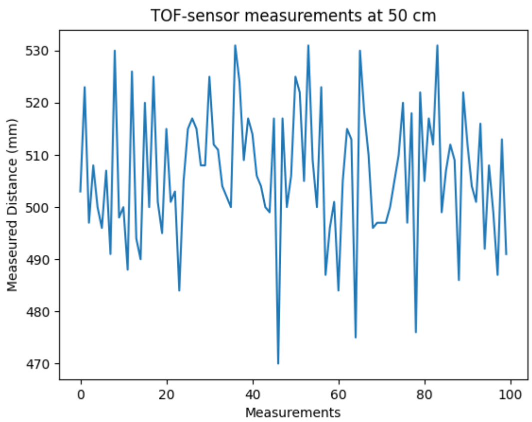

Figure B.4: 100 measurements with a TOF at 50 cm.

C
Setup

This section will discuss the some predetermined hardware and software aspects of the project as an whole.

Vehicles

The "Autonomous robots" that are used, are non-holonomic, 3 wheeled vehicles. They are outfitted with an ESP-WROOM 32 development board, as well as a motor control chip. 2 of the wheels are driven by a PWM signal, generated by the ESP. Whilst the third wheel is an unpowered castor wheel.

ESP-32 WROOM

The ESP-32 WROOM development board functions as the main decision making unit of the car. It is a low-cost, low-power system on a chip micro-controller. The chip is capable of WiFi and ESP-NOW communications, which allows the car on which is outfitted to communicate with other nearby cars. Furthermore, the board is outfitted with an Inter-IC-bus (I^2C). This feature allows other integrated circuits to communicate with the micro-controller. This is necessary for the required magnetometer in the form of an IMU. The I^2C bus is also used by the TOF sensor.

Inertial Measurement Unit

The inertial measurement unit consists of a integrated gyroscope, accelerometer and magnetometer. The magnetometer is most of interest, as the magnetometer determines the heading of the car using the earth's magnetic field. This is a feature necessary for swarming. The car is equipped with the MPU-9250. This IMU has a full scale range of $\pm 4800 \mu T$. Magnetometer measurement values can be retrieved at a frequency of 200 Hz using this chip. In Figure C.1 the accuracy can be seen when the motors operate at 40% power.

Time of Flight sensor

The time of flight (TOF) sensor uses the time difference between transmitting a infrared light signal and receiving its reflection. For this thesis the time of flight sensor that was used for the implementation and measurements was the 'VL53L0X', designed and produced by STMicroelectronics. This sensor is capable of measuring a distance from 20mm up to 1200mm, which includes the required range for this thesis. It determines the distances by using a laser in the form of a infrared light signal and measuring the time to detect the reflection of said signal.

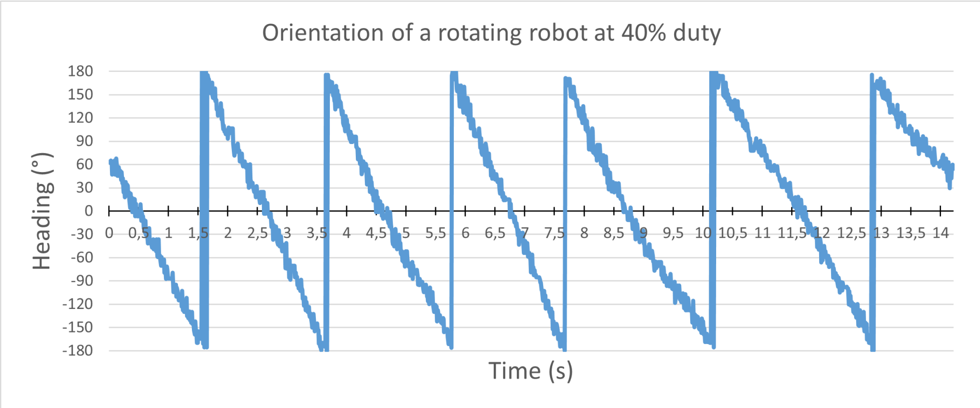

Figure C.1: The results of the IMU when the robot is spinning.

Espressif

The Espressif framework is used building, flashing and monitoring the code on an ESP chip. ESP-IDF is a tool written in python that comes with Espressif. It allows the user to interact with and flash instructions to build its program on an ESP chip.

Clion, C

ESP-IDF flashes C code onto the ESP chip. The C code is written in the integrated development environment Clion. Using a Cmake framework, independent modules are created.

PyCharm, Python

Python is used for modeling, Libraries such as numpy and pyplot are used to plot data and do calculations.

D

ToF Filter Algorithms Psuedo-code

D.1. Derivative filter algorithm

Used to filter the gathered data for the TOF sensor in order to detect potential obstacles.

Algorithm 1 Differential filter algorithm

```
sorted_data = empty_list[360]
for data in scan_data do
    sorted_data[data.heading] = data.distance
end for
return_list = empty_list[360]
for distance in sorted_data do
    previous_distance = previous_valid_data in sorted_data
    if distance − previous_distance ≤ threshold then
        return_list.add(distance)
    end if
end for
```

D.2. Windowed filter algorithm

Used to filter the gathered data for the TOF sensor in order to detect potential obstacles.

Algorithm 2 Windowed filter algorithm

```
for data in scan_data do
    if robot sees something then
        hit_miss_array.append(1)
    else
        hit_miss_array.append(0)
    end if
end for
for i of hit_miss_array do
    if hit_miss_Array[i] == 1 or AVG > 0.5: then
        AVG = calculate Average of hit_miss_array, AVG_array.append(AVG)
    else
        AVG_array.append(0)
    end if
end for
for i of AVG_array do
    if AVG_array[i] == 0 and AVG_array[i - 1] != 0  then
        windowArray = [AVG_array[i - 5], AVG_array[i - 4], AVG_array[i - 3], AVG_array[i - 2], AVG_array[i - 1]]
        if windowArray has more than 2 zeros then
            last 5 items in AVG_array are set to 0.
        end if
    end if
end for
Average the seen points and plot.
```

Bibliography

[1] Melanie Schranz et al. "Swarm Robotic Behaviors and Current Applications". Dutch. In: Frontiers in Robotics and AI 7 (2020). DOI: 10.3389/frobt.2020.00036.

[2] Craig W. Reynolds. "Flocks, herds and schools: A distributed behavioral model". Dutch. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87 (1987). DOI: 10.1145/37401.37406.

[3] Rathin Chandra Shit et al. "Location of Things (LoT): A Review and Taxonomy of Sensors Localization in IoT Infrastructure". Dutch. In: IEEE Communications Surveys Tutorials 20.3 (2018), pp. 2028–2061. DOI: 10.1109/comst.2018.2798591.

[4] Daniel Maier and Alexander Kleiner. "Improved GPS sensor model for mobile robots in urban terrain". Dutch. In: 2010 IEEE International Conference on Robotics and Automation (2010). DOI: 10.1109/robot.2010.5509895.

[5] Joydeep Biswas and Manuela Veloso. "WiFi localization and navigation for autonomous indoor mobile robots". Dutch. In: 2010 IEEE International Conference on Robotics and Automation (2010). DOI: 10.1109/robot.2010.5509842.

[6] Na Fan et al. "Decentralized Multi-robot Collision Avoidance Algorithm Based on RSSI". Dutch. In: 2021 13th International Conference on Wireless Communications and Signal Processing (WCSP) (2021). DOI: 10.1109/wcsp52459.2021.9613586.

[7] Minh Tu Hoang et al. "Recurrent Neural Networks for Accurate RSSI Indoor Localization". Dutch. In: IEEE Internet of Things Journal 6.6 (2019), pp. 10639–10651. DOI: 10.1109/jiot.2019.2940368.

[8] Aswin N Raghavan et al. "Accurate mobile robot localization in indoor environments using bluetooth". Dutch. In: 2010 IEEE International Conference on Robotics and Automation (2010). DOI: 10.1109/robot.2010.5509232.

[9] Xiuyan Zhu and Yuan Feng. "RSSI-based Algorithm for Indoor Localization". Dutch. In: Communications and Network 05.02 (2013), pp. 37–42. DOI: 10.4236/cn.2013.52b007.

[10] Yuan Zhou, Choi Look Law, and Jingjing Xia. "Ultra low-power UWB-RFID system for precise location-aware applications". Dutch. In: 2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW) (2012). DOI: 10.1109/wcncw.2012.6215480.

[11] Y. Tamai et al. "Three ring microphone array for 3D sound localization and separation for mobile robot audition". Dutch. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (2005). DOI: 10.1109/iros.2005.1545095.

[12] S. Gezici and H.V. Poor. "Position Estimation via Ultra-Wide-Band Signals". Dutch. In: Proceedings of the IEEE 97.2 (2009), pp. 386–403. DOI: 10.1109/jproc.2008.2008840.

[13] Yuan Zhou, Choi Look Law, and Francois Chin. "Construction of Local Anchor Map for Indoor Position Measurement System". Dutch. In: IEEE Transactions on Instrumentation and Measurement 59.7 (2010), pp. 1986–1988. DOI: 10.1109/tim.2010.2047987.

[14] Houria Chabbar and Mouhcine Chami. "Indoor localization using Wi-Fi method based on Fingerprinting Technique". In: 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS) (2017). DOI: 10.1109/wits.2017.7934613.

[15] Daisuke Sekimori and Fumio Miyazaki. "Self-Localization for Indoor Mobile Robots Using Multiple Optical Mouse Sensor Values and Simple Global Camera Information". In: Transactions of the Japan Society of Mechanical Engineers. C 71 (2005), pp. 3478–3485.

[16] Chao-Chung Peng, Yun-Ting Wang, and Chieh-Li Chen. "LIDAR based scan matching for indoor localization". Dutch. In: 2017 IEEE/SICE International Symposium on System Integration (SII) (2017). DOI: 10.1109/sii.2017.8279202.

[17] Yuan Xu et al. “Indoor INS/LiDAR-Based Robot Localization With Improved Robustness Using Cascaded FIR Filter”. Dutch. In: IEEE Access 7 (2019), pp. 34189–34197. DOI: 10.1109/access.2019.2903435.

[18] Jinseok Woo and Naoyuki Kubota. “Simultaneous localization and mapping using a robot partner in dynamic environment”. In: SICE Annual Conference 2011 (2011), pp. 524–529.

[19] Rakshit Shah. Formula to convert the RSSI value of the BLE (Bluetooth Low Energy) Beacons to Meters | BeingCoders. Feb. 8, 2022. URL: https://medium.com/beingcoders/convert-rssi-value-of-the-ble-bluetooth-low-energy-beacons-to-meters-63259f307283.

[20] Athanasios E. Lagias, Thomas D. Lagkas, and Jie Zhang. “New RSSI-Based Tracking for Following Mobile Targets Using the Law of Cosines”. Dutch. In: IEEE Wireless Communications Letters 7.3 (2018), pp. 392–395. DOI: 10.1109/lwc.2017.2779507.

[21] Jihao Yin et al. “A Simple and Accurate TDOA-AOA Localization Method Using Two Stations”. Dutch. In: IEEE Signal Processing Letters 23.1 (2016), pp. 144–148. DOI: 10.1109/lsp.2015.2505138.