Robot platform positioning in a theater setting

N. (Nick) van Lange

BSc Report

Committee:
Dr.ir. E.C. Dertien
Prof.dr.ir. G.J.M. Krijnen
Prof.dr.ir. J. Schmitz

July 2018

018RAM2018
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Summary

Mobile robot positioning is used in many applications. Many different system have been devised for this purpose. The usual setting for such systems are in factories, homes or outside. In this paper a positioning system is proposed for a more unique kind of setting, a positional system that can be setup in a theater.

The system is designed to work for a specific juggling performance. In this juggling performance the juggler throws balls, which bounce multiple times against ramps. After bouncing multiple times they come back to him. In this show the mobile robot platforms, with ramps on them, need to move to exact positions during the show. The ramps need to be in very exact places to ensure that the balls bounce back correctly.

The proposed solution to this problem is to use a combination of odometry and tape markers on the ground. Both of these are not affected by the distortion and interruption caused by the theater setting and can be setup with ease. With odometry the position and the orientation of a vehicle are calculated using measurements of the rotation of the wheels. An issue with odometry is that it does not measure the position directly. Because of this errors in orientation and position will build up over time and distance. The usual solution to minimizing this effect, is to use an absolute measurement system. In the proposed system the mobile robots will gain absolute measurements using the known position of marked patterns on the ground. These marked patterns are detected using reflectance sensors on the bottom of the robot. With the use of these sensors the robot is placed exactly on top of the markers. With the position and orientation known of the markers, the errors in position and orientation can be fixed. The resulting system has promising results with room for improvement.

Contents

1 Introduction

These days autonomous moving robot platforms are used in many applications. Including but not limited to: logistics, maintenance, security and cleaning. These robots rely on positioning systems to know where they are and where to go. Many of these positioning systems exist. With all of them having trade-offs. Some are very precise, but only work in a limited space. Other systems can function almost everywhere, but as a result the system is less accurate.

The usual setting for these robot platforms are in factories, outside or in homes. The setting presented in this report is unique, a theater setting. A robotic platform has to position itself on stage for a juggling show. In this juggling performance a ball is thrown against multiple ramps. The ball bounces against multiple ramps and comes back to the juggler. Two stills from the show are depicted in figure 1.1 and figure 1.2. Currently in this performance the platforms are static, but this causes some issues. If he wants to do a different routine the platforms have to be moved by hand. This problem is already partially solved, with remote control platforms. But using remote control platforms also has some issues. It is difficult to drive the platforms to a precise enough spot. Small errors in orientation can result in the ball not bouncing in the correct way. Another issue is that controlling multiple platforms at the same time is difficult for one person.

This leaves the proposed system, self-driving robot platforms which uses a positioning system to drive to the exactly needed spot. It is also useful for presetting routines. Where with a human controller there would be a need to practice on where which platform should move on all stages of the show, a good, robust system would need to only be programmed once to do it and would always do the same.

The goal of this research is thus to make a mobile robot platform, which is capable of precise positioning in a theater environment.

Figure 1.1: A still from the performance.

Figure 1.2: Another still from the performance.

2 Analysis

2.1 Detailed problem description

As discussed before, in the show the robot platforms need to position themselves to precise locations. All of this positioning needs to be done in a theater setting. A theater setting brings multiple interesting problems. Many indoor positioning system use some static infrastructure in the building, like Wi-Fi modems or some other kind of beacons. In a theater setting there can be no reliance on infrastructure, since every theater is different. The positioning system needs be able to be set up in a reasonable time frame. The system can also not interfere with the show, so no bright LEDs which distract viewers and no system that blocks vision. Another issue is that system failures are unacceptable. If the system mispositions a platform then the performance fails, the show would have to be paused to set the platform right. If a platform loses power, or encounters some other problem, it needs to be able restart and function for the rest of the show, after all the show must go on. Another thing to consider is that there is also a lot of electromagnetic interference from stage equipment. Systems reliant on signal strength will have issues.

Further information is that the playing field for the show is about 7m x 7m. Walls and ceilings can not be used, since it's unreliable that they are available in every theater. The platforms also all have different sizes. An example map with positions for the platforms is depicted in figure 2.1. It is assumed that the surface is flat so that only 2-dimensional positional data is necessary. So to position the platform correctly three data are needed, the x position, the y position and the orientation of the platform. If these three data are known exactly the platform can position itself correctly.

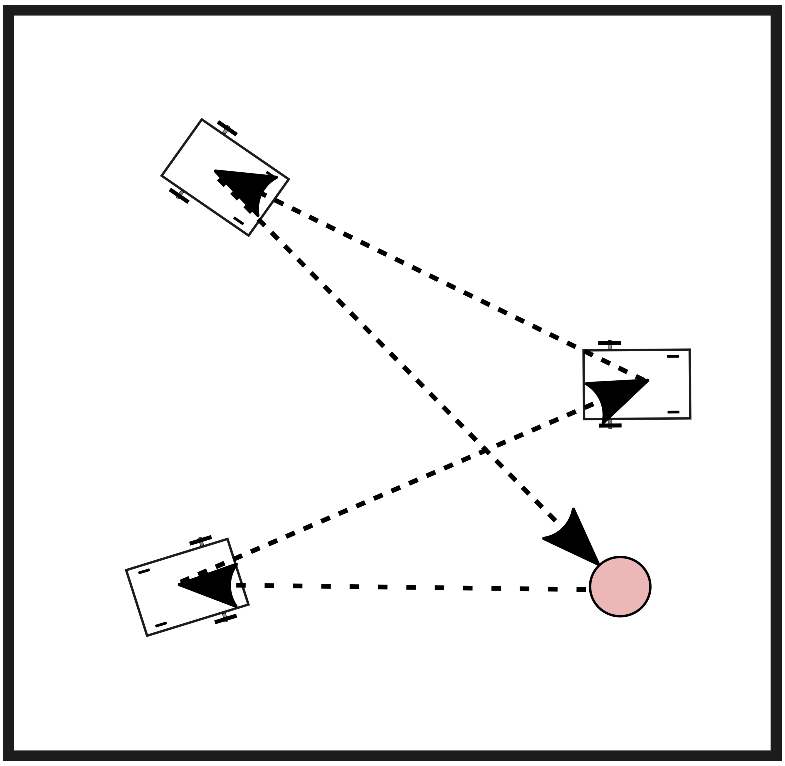

Figure 2.1: A map of the setup for the platforms. The circle represents the juggler, the squares the platforms and the arrows a path for the ball.

2.2 Current positioning systems

Over the years many different kinds of positioning systems were conceived. In "A survey of indoor positioning systems for wireless personal networks"[1] and "Overview of current indoor positioning systems"[2] many different positioning systems are evaluated. A nice visualization is depicted in figure 2.2. A number of them can be easily identified to be not usable based on some simple requirements. Any positioning system that is above decimeter level precision can immediately be discarded as unfit for the system. This immediately eliminates the systems W-LAN, APGS and GSN. As discussed before in theaters there is a lot of electromagnetic interference system which rely on signal strength will have a notable drop in their effectiveness. This eliminates Locata and Geodetic GNSS. These two systems rely on ground based pseudo-satellites to function. This kind of system is usually setup outside, but can work inside buildings. These pseudo-satellites are then setup in corners and measure the signal strength to determine the position of the target. This is system is deemed unfit because of the previously mentioned issue with system relying on signal strength.

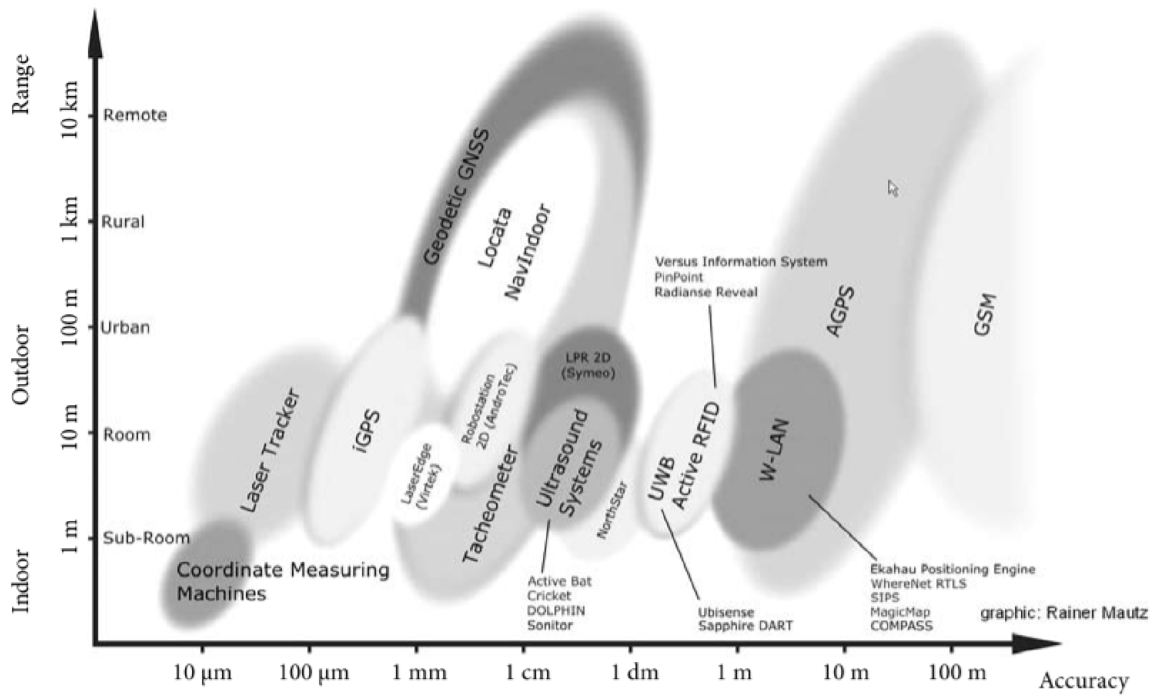

Figure 2.2: Overview of multiple positioning systems, from [2]

There are two main groups of positioning systems. Systems that measure the absolute position and system that measure the change in position to calculate the next position, this is known as dead reckoning. Usually a combination of these two systems is used to gain more accuracy. The absolute position measurement is more accurate after long periods of time, and the dead reckoning system is more accurate short term. Some positioning system are discussed next.

2.2.1 Ultra-wideband

Ultra-wideband(UWB) uses short radio pulses send over a wide frequency to send information. Because it uses such a wide frequency band it can send huge amount of data in low amount of time. This technology can also be used for positioning systems. There are two main methods of getting accurate absolute positional data with UWB, using time of flight(TOF) or time difference of arrival(TDOA). Alternatively the signal strength can also be measured to calculate the position, but as discussed before this is deemed unreliable.

TOF

The concept of TOF is simple. Two nodes are an unknown distance apart. First the time it takes for the signal to travel from node to node is measured. The distance can now be calculated by multiplying this time with the speed at which the signal travels, the speed of light. This does leave the issue of how the travel time can be measured. The nodes seemingly would need to have the exact same clock times. But this is not necessary. The protocol is as follows: node one sends out an UWB pulse with a time stamp of at what time it was send. The node 2 receives it and measures the time. Node two then answers with sending an UWB signal back, with the time it measured when the signal was received and the time at which it sends this signal. Now when node one receives this signal it can calculate the TOF with the following formula:

$$TOF = \frac{T_{1receive} - T_{1send} - (T_{2receive} - T_{2send})}{2}$$

As can be seen it does not matter if the clocks of the two nodes are not in sync. As long as they run at the same speed the time will be received correctly. The distance can now be calculated by multiplying it with the speed of light.

To actually measure the position of a node at least 3 anchors with known positions are necessary for 2 dimensional positioning. Then with the use of a trilateration algorithm the position can be calculated. The concept is shown in figure 2.3. A TOF system for UWB can reach 10cm accuracy in line of sight cases[3]. In non line of sight cases the accuracy suffers greatly. So line of sight is required for this system to function.

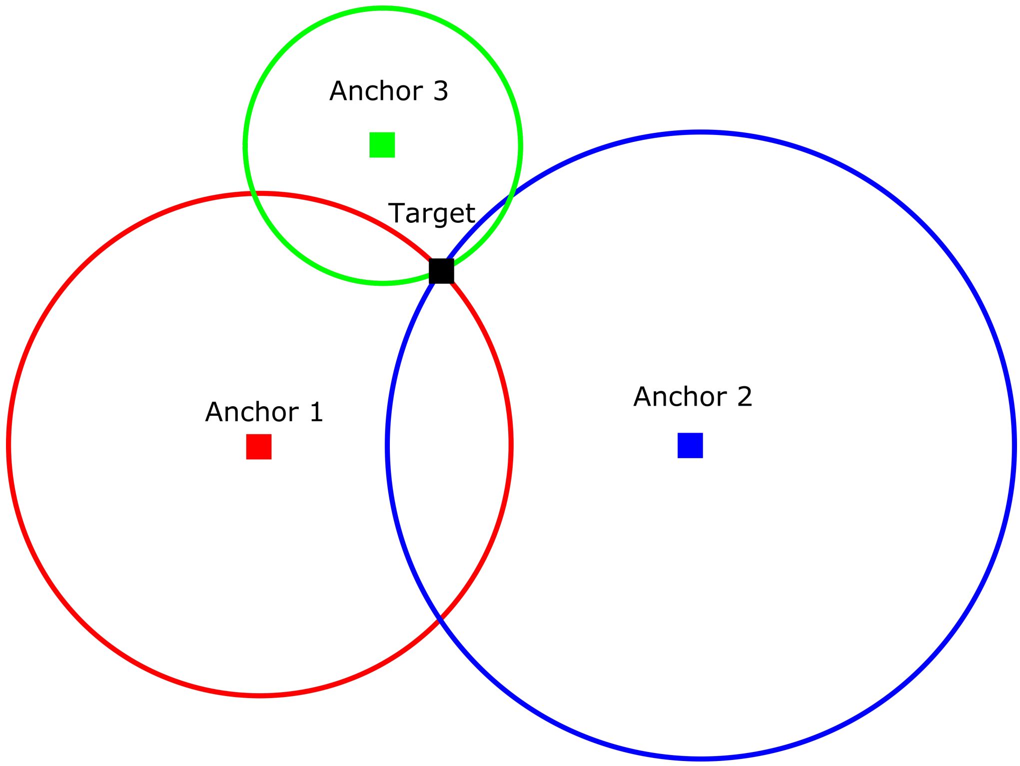

Figure 2.3: The principle of a TOF positioning system

TDOA

In a TDOA system the anchor nodes do not need to send a signal, making it a more versatile system and giving the system more scalability. In this system the target sends out a signal. The anchors measure the time at which they receive the signal. Now two anchors can compare the times measured and determine the relative distance to the nodes. This will result in a hyperbole between the 2 anchors. With 3 anchors there will be a total of 3 hyperboles. Similar to the TOF system the location can now be determined with the cross point of the hyperboles. The main advantage of this system over TOF is that it scales better, multiple targets do not reduce the frequency of which a TDOA updates. Disadvantages of the system compared to TOF, is that the anchors need to have synced clocks and it is slightly less accurate.

2.2.2 Ultrasound

Ultrasound is sound which has a frequency above the human hearing frequency. Ultrasound can be used for a positioning system. Ultrasound modules are often used on the side of a robot. Such an ultrasound module will send out a signal and wait for the reflection. In doing so it can be determined at what distance an object is from the sensor, like a radar. But this is of little use for our case, the playground is entirely flat with no reliable objects for reflection. Luckily different systems exist too. Cricket[4] uses a combination of a 433 MHZ radio signal and ultrasound. An anchor node with a known location sends out a radio signal, and at the same time it sends out an ultrasound wave. The receiving node will receive the two signals at two different times. The radio signal moves at the speed of light and the ultrasound signal at the speed of sound. The time traveled by the radio signal is negligible compared to the time traveled of the ultrasound signal. So the distance can be calculated according to the following formula:

$$d = (t_{ultrasound} - t_{radio}) * c_{sound}$$

with $t_{ultrasound}$ being the time at which the ultrasound is received, t_{radio} the time at which the radio frequency is received. This system requires line of sight. With the distances from the anchors known the position can be calculated with the TOF scheme.

Ultrasound systems can be very accurate, an accuracy of 2 cm is possible[5]. As a trade-off for this accuracy, the system has very low latency(1-2hz). This is caused by the slow speed of sound. The system also requires line of sight. Furthermore noisy environments can disturb the sensor from detecting the ultrasound signal. This might cause issues in a theater.

2.2.3 Reflectance sensor

Another concept is to use reflectance sensors on the bottom of the platform. An array of these sensors is commonly used for line following robots[6]. A simple tape path could be made on a ballet floor. This could be placed down on stage, with the route marked in tape. The tape could run from beginning to the end, but this would result in limited options for the show. The robots will only be able to follow the line during a show. And since overlap of paths results in problems, only a limited amount of paths can be laid down. Potentially only a few set parts of the floor could be marked with tape. The system could use these set parts to determine exactly where it is. This would require the system to be combined with another approximate positioning system, so that the platform knows what tape marker it is detecting.

2.2.4 Camera tracking

A camera system could be used. A tracking camera could be used[7]. This camera is setup on a stand and follows an identifiable object. The camera can calculate the location of the object with the knowledge of it's own coordinates. The object can be identified in a multitude of ways. A color or a recognizable pattern could be used. The article mention a precision in the range of 5 cm. A big issue with this system is that platform needs to be recognizable with variable lighting conditions. This can be a big issue in theaters since the lighting is different in different theaters. Other disadvantages are that the system requires line of sight and that it can ruin the aesthetic of the platform.

2.2.5 IMU

An IMU(inertial measurement unit) is a sensor package that is used for measuring the acceleration, angular rate and magnetic field, a body experiences. This is measured using a combination of an accelerometer, magnetometer and a gyroscope. An IMU can used for short term precision measurement while another system measures the position accurately long term. An example of this is a combination of gps and IMU for vehicle navigations[8]. The separate sensors in an IMU have different functions.

Accelerometer

The accelerometer measures the acceleration in 3-axis. To get the position out of an accelerometer it has to be integrated twice. Integrating something twice results in very inaccurate results. Because it is so inaccurate it is deemed unusable for positioning. It can potentially be used for measuring if a ball hits the platform, since it will detect an abrupt movement of the platform.

Gyroscope

A gyroscope measure the angular change over time around 3-axis. A digital gyroscope works as follows. Two capacitive elements are placed next to each other and when the module is tilted, due to the Coriolis effect the capacitance of the capacitors changes, proportional to the amount of change. The gyroscope measures this over 3-axis. But since the plane on which the platform drives is assumed to be 2-dimensional only 1-axis is interesting. This axis the turning around the z-axis, also known as the yaw-axis. To make the assumption that only 1-axis is need the gyroscope has to be installed perpendicular to the ground plane. Otherwise the measurements will have a small offset, since the gyroscope will know not just turn around the z-axis, but also a bit around the y-axis and x-axis depending on the offset. The measured change in angle can be used for dead reckoning. Without a second system in place to return the absolute orientation the gyroscope will drift off over time.

Magnetometer

A magnetometer measures the direction of the magnetic field. Like the gyroscope and the accelerometer it does this around 3-axis. A magnetometer can be used to gain the orientation, assuming there is no electromagnetic interference. Since there is a magnetic field around the world that points north, the sensor can see where the north is, like a compass. Using this it can always determine in what orientation it is. Sadly enough magnetic interference is an issue in a theater. There is a solution to mitigate some of the magnetic interference with the use of magnetic field mapping [9]. This would require some extra steps in setting up the system, the vehicle would need to drive around on the theater before the show to map the magnetic field. But if it works it can be used for measuring the absolute orientation. This measurement can be used for stopping gyroscope drift.

2.2.6 Odometry

Odometry is the concept of using measurements of the wheel rotation to determine the systems position. For a twee wheeled vehicle if the wheel rotation of both wheels is known, the change in orientation and the positional change can be calculated. A way for measuring the wheel rotation is with the use of quadrature encoders. For this an encoder disk with multiple notches is placed on the axle. With a ir light sensor it it can be measured each time the disk passes. This can be counted and the change in position can be measured. A quadrature encoder does have the problem of having a limited resolution. Odometry systems can be made very accurate with careful tuning. An accuracy of 30mm is possible after driving a path of a 4m x 4m square[10]. But this requires high encoder resolution, slow speed, a flat no slip surface and a lot of tuning.

2.3 Full system concepts

Most of the single systems have one or more weaknesses. So it is better to combine systems to get an improved result. In general it seems a good idea to use a dead reckoning system with another system that returns the absolute positions. This ensures that the drift error from the dead reckoning is fixed by the absolute measurements. Some concept that were thought of are written below.

2.3.1 Tape matrix

One idea is to make a matrix on the ground with tape on a ballet floor. In the matrix the squares are given coordinates, in a similar way to a chessboard, in the form of (x,y). In each of these sectors a cheap RFID tag with a unique identifier is placed. The platform can now always know in which sector it is. If this system is realized then the size of one square will be somewhere around 50cm x 50cm. With squares of 50cm x 50cm the amount RFID tags needed is $(700/50) * (700/50) = 196$. A high amount, but RFID tags are very cheap, only costing about 15 cents. This system alone gives an accuracy of 50 cm, but with the use of reflectance sensors it can be made more accurate. Whenever a line is measured with a reflectance sensor it can be determined exactly where one coordinate of the platform is. For example when the platform crosses from sector(1,1) to (2,1), The system know that when the reflectance sensor turns on then the x coordinate of the platform is 50 cm. The orientation of the platform can also be fixed, when it reaches an edge the platform can use it to move precisely to either exactly 0,90,180,270 degrees, by setting the platform straight. The system in work is depicted in figure 2.4.

This system gives exact absolute measurements periodically. Combined with the use of dead reckoning with either odometry or an IMU can lead to accurate results. The system does have a some issues. It is unclear what the platform should do in crosspoints of x and y lines. But this can be solved by either avoiding those areas or ignoring the reflectance sensor when the system thinks it is close to a crosspoint.

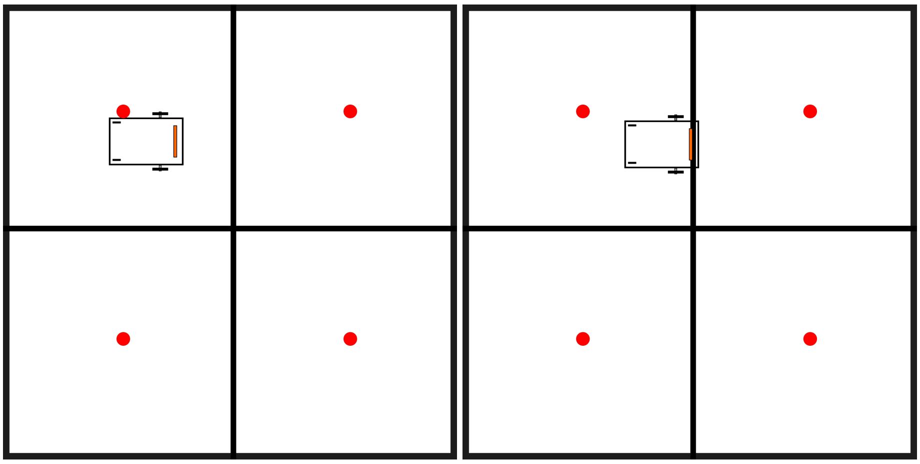

Figure 2.4: Figure of the tape matrix in progress. **Left:** The robot platform knows it is in sector (1,1) because of the rfid tag(red circle). **Right:** The robot platform crosses the vertical line, with the knowledge of its approximate position, it can now assume that the x position is 50. If the platform is unsure about it's orientation it can also set itself to 90 degree.

2.3.2 Marking end position

With this system a previously discussed positional system is used to get an estimate of the position. The end positions will be marked with a tape pattern. The reasonable accurate positional system is used to drive to the end position. With the use of the tape pattern the platform is set in an accurate position. It does this using reflectance sensors. This system can accept some inaccuracies from the positional system, since it will reposition exactly when it reaches the tape. The pattern has to be made in such a way that the platform can position correctly when it detects it. Some concepts for this are shown in figure 2.5 and figure 2.6. The main advantage of this system is that the platform will be set precisely on the position of the tape, drastically increasing accuracy. A disadvantage is that if the positional system is too inaccurate it can miss the tape and fail to position completely. Another disadvantage is that the robot platform needs to make an approach from a certain angle for the system to work.

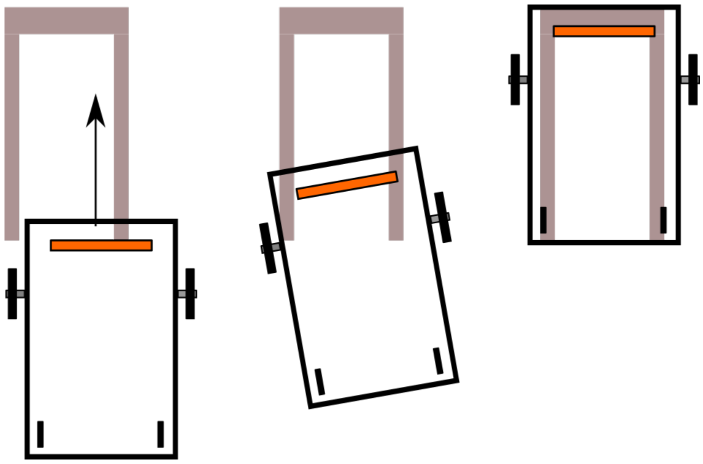

Figure 2.5: A figure showing the robot platform positioning on tape. The bar on the platform are reflectance sensors. Steps with this tape pattern are: 1. Detect which side the tape is. 2. Turn to the other side till only the outer reflectance sensor is above the tape. 3 Turn the other way till the tape is seen on both sides. 4. Move forward till all sensors detect tape.

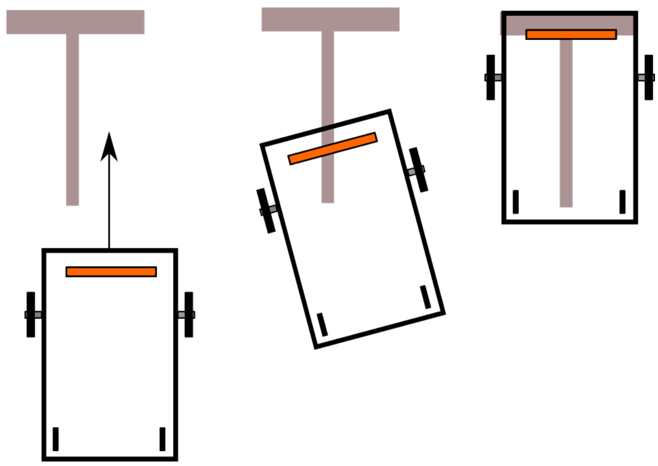

Figure 2.6: A figure showing the robot platform positioning on tape. The bar on the platform are reflectance sensors. Steps with this tape pattern are: 1. Detect the distance between the measured tape position and the center. 2. Turn to position the tape in the middle with the use of a closed loop controller. 3 Stop when all sensors turn on.

2.4 Requirements

To be able to judge what system will function best some requirements for the system need to be set. The most important requirements are listed below, The system must:

- be able to drive to a precise location and end in a precise orientation.
- be able to handle multiple mobile platforms
- be able to be setup and function in a theater environment
- have a functioning control system
- be robust

These are the most important requirements. The system is considered a failure if it doesn't fulfill these requirements. But there also some less important requirements. These are sum up below:

- Reposition as fast as possible
- Handle as much weight as possible
- look Aesthetically pleasing
- be Cheap

2.4.1 Be able to drive to a precise location and end in a precise orientation.

If the system is not precise enough then the system can be considered useless. Since if the positioning is not precise enough then the ball can miss after bouncing. It's hard to estimate how accurate the platforms need to be. If the orientation is wrong on a platform it will cause a cumulative error. If the orientation is off by 1 degree it will result in an offset of $d * sin(1)$. In the worst case scenario the next two platforms also have a 1 degree offset. With the assumption that there is 3 meter of distance between each bounce the total offset will be:

$$3 * sin(3) + 3 * sin(2) + 3 * sin(3) = 0.30m$$

This is already a lot, so the goal for orientation is set to 1 degree offset in the worst case scenario.

The position precision requirement has a little more leeway. The offset caused by a positional change will not increase with time or distance, it can be considered flat. It still should be as accurate as possible, therefore the goal for accuracy in position is set to 3 cm.

2.4.2 Be able to handle multiple mobile platforms

During the performance it is desired to have 3 platforms moving. The made system should thus be able to handle 3 platforms.

2.4.3 Be able to be setup and function in a theater environment

The system of course has to work in a theater. If the system doesn't work in a theater it can't be used. But every theater is slightly different. Some have more space then others, some have more electromagnetic interference. So the system should not rely on things that can easily be disturbed. Furthermore the system needs to be setup in a theater, in a reasonable time frame. So systems that would require a lot of effort to setup or rely on very precise placement of sensors are off limits.

2.4.4 Have a functioning control system

The platforms need to move to right places on the right moments. For this a control system is necessary. In this control system multiple commands should be possible for the platforms. It should also be possible to interrupt the platforms when things go wrong. It should also have some backup methods like being able to take manual control.

2.4.5 Be robust

The system should never fail. During a performance it is unacceptable that the platforms don't function correctly. If the system has a high failure rate then it can't be reliably used in a performance. The goal is thus to have the failure rate as low as possible. A reasonable goal would be a failure rate below 1%.

2.5 Final system choice

Since the orientation needs to be very precise, the best solution is to mark the end position with tape. When the position is marked with tape the platform can position precisely on it. To ensure the reflectance sensor array hits the tape when approaching, the reflectance should cover over as much of the width of the platform as possible. This concept will be combined with 2 other positional system, dead reckoning and UWB TOF positioning. The absolute positioning is chosen to be UWB because it is influenced the least by the setting. The line of sight issues can be solved by doing the following things, putting the UWB modules on top of stands of 1 meter high and using 4 anchors in the corners. When the UWB modules are on stands they will be high enough that line of is sight is not broken by other platforms. This does require that the antenna of a platform is on top of it. The person on stage can still block the signal, that's why there are 4 anchors. If one has broken line of sight the other 3 are still accurate. Because the other 3 are still accurate, the position can still be determined accurately. There should always be 3 anchors with line of sight unless the person is really close to the platform.

The dead reckoning will be done with odometry, with extra help from an IMU. Odometry is chosen because it can be made very accurate. The IMU is added because it is cheap and its additional data can increase the accuracy of the odometry. The full theoretical system in work is depicted in figure 2.7

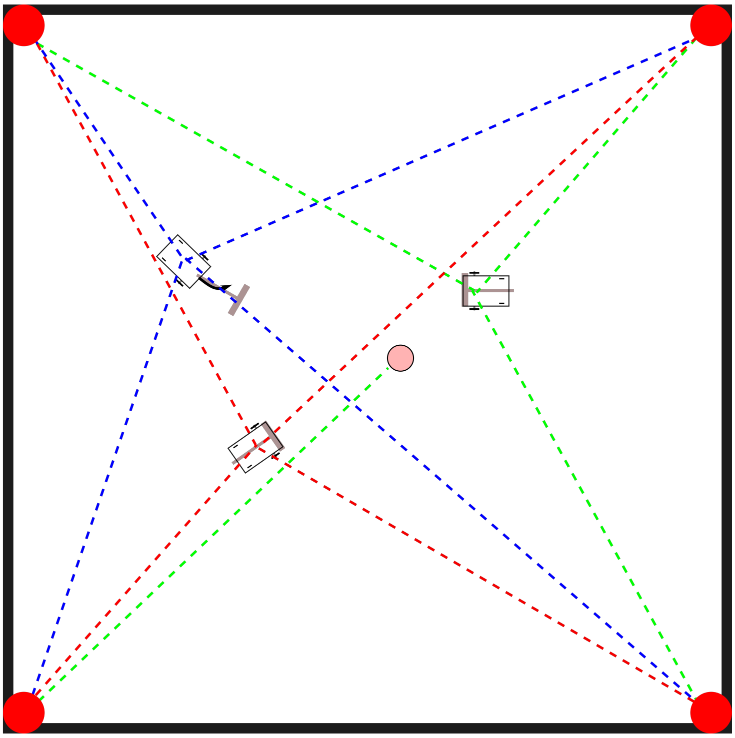

Figure 2.7: The full desired system. The anchors are setup in the corners marked with a red circle. The measurements are shown with striped lines. The platforms are positioned exactly on top of the tape, besides the upper left one which is positioning on top of the tape.

3 Implementation

3.1 Material choices

With the analysis done the system can actually be realized. For testing purposes a prototype is made. To make this prototype all the parts are required. So parts have to be selected for the multitude of roles.

3.1.1 Frame

The frame is chosen to be made from aluminum bars. Aluminum is light, sturdy and can easily be worked with. The platforms are square so the prototype platform is also modeled square. Size of the frame is modeled to be about the same as the ones used in the show. This gives a size of around 32cm x 45cm. The frame is supported with 4 wheels, 2 of which are motor controlled. The used motors is the parallax motor and wheel kit[11]. These wheels support up to 30 kilo, more than enough for the frame. The other two wheels are simple rotational wheels. The platform will be able to steer by changing the power of the separate wheels. At maximum speed a motor was measured to require about 3,6A. So for two motors the required current is 7,2A. The motors require decent amount of current so a H-bridge is necessary. The chosen H-bridge for this is a HB-25. This H-bridge can deliver up to 25A, which is more than enough. To deliver the amount of power needed, a power source is required. For this a 12V lead-acid rechargeable car battery is chosen. This car battery stores up to 12 Ah. This should be enough for a show, since the platforms are static most of the time.

3.1.2 Microcontroller and sensors

To make the positioning system discussed in analysis, a lot of different sensors are necessary. The chosen products are put in table 3.1. The microcontroller is chosen to be an Arduino Mega 2560. The arduino mega is chosen because a lot of digital input pins are needed. Since every reflectance sensor needs 1 pin. Furthermore with a 16MHz clock speed it should be fast enough to do the computing for a robotic platform. For the IMU a GY-85 is chosen. It is mainly chosen for being cheap and being 'good enough'. More performance could be gained from a more expensive IMU, like a Xsens IMU. But those are 40 times more expensive. The IMU module includes 3 sensors, the ADXL345 three axis accelerometer, the ITG3205 three axis gyroscope and the HMC5883L magnetometer. Although most likely only the gyroscope will be used.

The reflectance sensor array used is the QTR-8RC. This is a reflectance sensor of which the outputs can be read by digital pins. These arrays each have a width of 8 cm, with a sensor each cm. It is important that these sensors are attached as close to the ground as possible. To ensure this, the reflectance sensors are attached in such a way that the height of the sensors can easily be modified.

For the UWB module a DWM1000 is chosen[12]. According to the data sheet it can locate objects which move at 5m/s with an accuracy of 10cm. The module also has a range of 20m, which is enough since in a space of 7m x 7m, the maximum possible distance between any point and an anchor in the corner is about 10m. Finally the used motor set includes 36 position encoder. With a 36 position-encoder a resolution of 144 can be reached. This results in that the position of a wheel can given to the precision of 2.5 degrees.

Sadly enough the UWB module arrived to late to be integrated. So the system has to function without an absolute position measurement. The system can still work if the drift error from the odometry is not too high.

Table 3.1: The chosen sensors and microcontroller

Microcontroller	Arduino Mega 2560
IMU	Gy-85
Reflectance sensor array	QTR-8RC
UWB module	DWM1000
Quadrature encoder	36 position-encoder set

3.2 Physical platform design

With all the needed sensors and materials known the platform can be made. As example there already are some remote control platforms. In figure 3.1 the bottom of such a remote control platform can be seen. For testing purposes it seemed easier to make a new one, since these platforms are still in use. So a prototype robot platform was made. This platform is designed to be similar to those remote control platforms, so that those could potentially be modified to save cost and effort. The final prototype frame can be seen in figure 3.2.

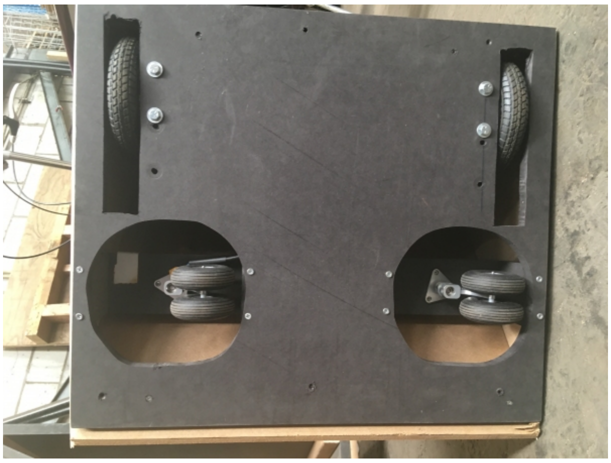

Figure 3.1: The bottom of a remote control platform

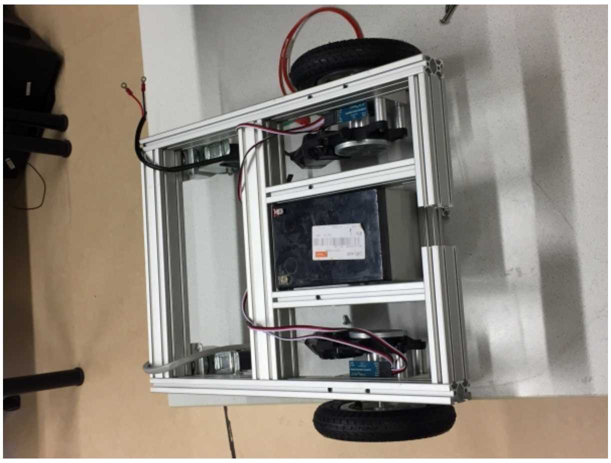

Figure 3.2: The frame of the prototype platform

3.3 Utilizing the sensors

With sensor choices made they need to be calibrated. The data measured also needs to be converted to useful information. This is done in different ways for the sensors

3.3.1 Odometry

Considering the entire system there are 3 values which are interesting, the x-coordinate, the y-coordinate and the orientation of the platform. A few things have to be defined and measured first. In figure 3.3 a top view of the platform can be seen with some measurements. To calculate the change in x and y position, the rotational speed of the wheels has to be converted to linear speed. The diameter of the wheels is 152mm(6 inch). So:

$$C = \pi * D$$

$$C = \pi * 152mm = 479mm$$

With C for circumference and D for diameter, the linear speed given the angular speed of a wheel is then ω(in radians). So the linear speed is given by:

$$V = (C * \omega)/2\pi$$

With the linear speed known the change in x-position, the change in y-position and change in orientation over time can be calculated:

$$V_x = (V_L + V_R)cos(\theta)$$

$$V_y = (V_L + V_R)sin(\theta)$$

$$\omega = (V_L - V_R)/b$$

With θ being the heading of the platform and b the axle length. With the old position known the next position can simple be calculated like:

$$x(t) = x(t-1) + \delta t * V_x$$

$$y(t) = x(t-1) + \delta t * V_y$$

$$\theta(t) = \theta(t-1) + \omega$$

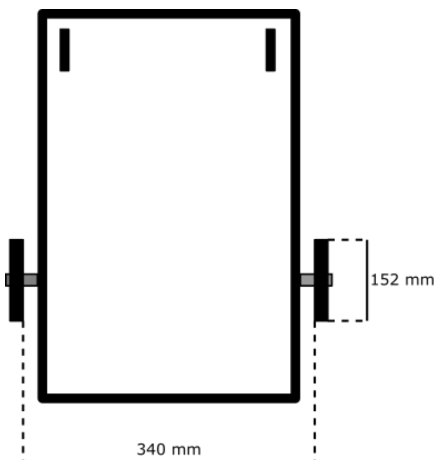

Figure 3.3: A top view of the platform showing the axle length and wheel diameter

Quadrature encoder

The actually get the angular speed of the wheels a sensor is needed. In the motor set such a sensor was already available, a quadrature encoder. The used quadrature encoder works in the following way: there is rotary disc with increments(36 in this case), this disc turns through two optical detectors. When the disc is in front of one of these detectors it will block the light en the signal goes from high to low. With a 36 position encoder a resolution of 144 can be achieved. This gives an accuracy of 2.5 degree for each wheel. This results in a distance resolution of $C/144 = 479/144 = 3.33mm$ per wheel.

Errors

The main causes for errors in odometry can be generalized in to two groups. Systematic errors and unsystematic errors. Systematic errors are categorized by inaccurate readings, where as unsystematic errors are caused by unexpected or immeasurable things. Common systematic errors are[10]:

- Unequal wheel diameters
- Average of both wheel diameters differs from nominal diameter
- Misalignment of wheels
- Uncertainty about the effective wheelbase (due to non-point wheel contact with the floor)
- Limited encoder resolution
- Limited encoder sampling rate

Common unsystematic errors are:

- Travel over uneven floors
- Travel over unexpected objects on the floor
- Wheel-slippage

Most of these systematic errors are preventable or the effect of them can be minimized with fine tuning. Unsystematic errors on the other hand are harder to prevent. Luckily travel over uneven floor and unexpected objects should not cause great issues, since the stage will be flat and clear. Wheel slippage can be minimized with low acceleration and low deceleration of the wheels, but it is still not completely preventable.

3.3.2 Gyroscope

The gyroscope measures the change of angle per second. This data is already calculated from the odometry, but a gyroscope has the potential to be more accurate. The gyroscope does not have the issue of the limited resolution that odometry has. It does still have the issue of drifting of, but to a lesser degree then odometry.

The used gyroscope is a ITG-3200[13]. Before the gyro can be used it needs to be calibrated. Calibration is first implemented using the scaling factor on the datasheet. But this scaling factor is never 100% accurate. So the gyroscope is calibrated using the calibration technique described in[14]. The steps taken are depicted in figure 3.4. After calibration the drift after turning it around 10 times was 0.9 degree. Or a drift of 0.09 per 360 degree turn. Which is an acceptable error.

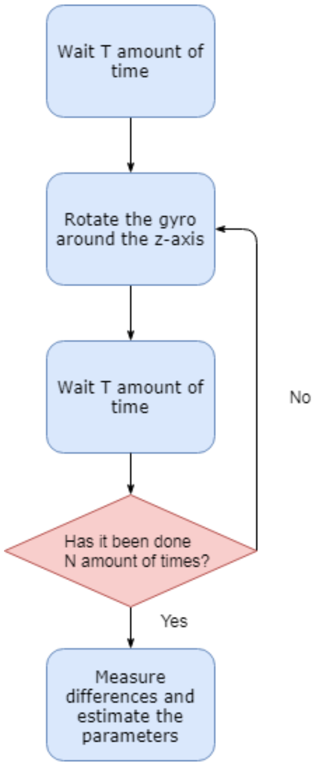

Figure 3.4: A drawing of the platform showing the axle length, wheel diameter and frames.

3.3.3 Reflectance sensor

In the system the end position will be marked with a tape pattern. As discussed before this will be measured with a reflectance sensor array on the bottom of the platform. This sensor detects the tape and positions the platform on top of it. Making sure that the platform ends on the desired location and in the right orientation.

The chosen shape for the tape is the T-form. This makes the functioning of the system when it detects the tape simple. The principle is simple. One or more of the reflectance sensors detect the tape. Knowing which of the sensor(s) detects the tape, the error can be determined. This is then inserted in a control loop that controls the motors. The chosen controller is a PID controller. It will straighten the platform on the tape, then when all sensor turn on at the end the platform will stop.

The used reflectance sensor arrays have 8 reflectance sensors, with a separation of 1 cm. The sensors return a reflectance value. With low values for strong reflectance and high values for low/no reflectance. An important things in the used tape and floor will be that either the floor has low reflectance and the tape has high reflectance, or vice versa. The reflectance is also dependent on how high above the ground the sensors are. If they are higher of the ground then the reflectance is lower. Before the sensors can be used they have to calibrated. A reflectance value needs to be chosen. For non reflective tape and a reflective floor, if the measured value is above the reflectance value then the sensor is above tape.

If the platform misses the tape the system fails, since it will not find the tape and can't position correctly on it. To reduce the chance of this happening, instead of only using one sensor array, 3 are used to increase the width. This covers a total of 23 cm. So the maximum error the platform can have when approaching the tape is $maxerror = 11.5cm + tapewidth/2$. Assuming that the width of the tape is around 2 cm gives a maximum error of 12.5 cm.

Since 3 reflectance sensor arrays are used there is a total of 24 separate values. Measuring them all will result in a boolean array of 24 values. With the knowledge of the position of each of the sensors the error can be easily calculated.

With the error a PID controller can be implemented. The basic formula for a PID controller is:

$$u(t) = K_p * e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$$

The K values have to be tuned to get a good result.

3.4 Attaching the sensors

With all details of the sensors known, they can be connected to the main frame. The Arduino mega requires 5 volt for power. With the usage of an adapter, the 12 volt of the battery can be converted to 5v. The Arduino provides power for the rest of the sensors. The reflective sensors are set on a bar of which the height can be modified. The platform with everything connected can be seen in figure 3.5. There is no slope attached to it but this can be easily fitted on top of the platform.

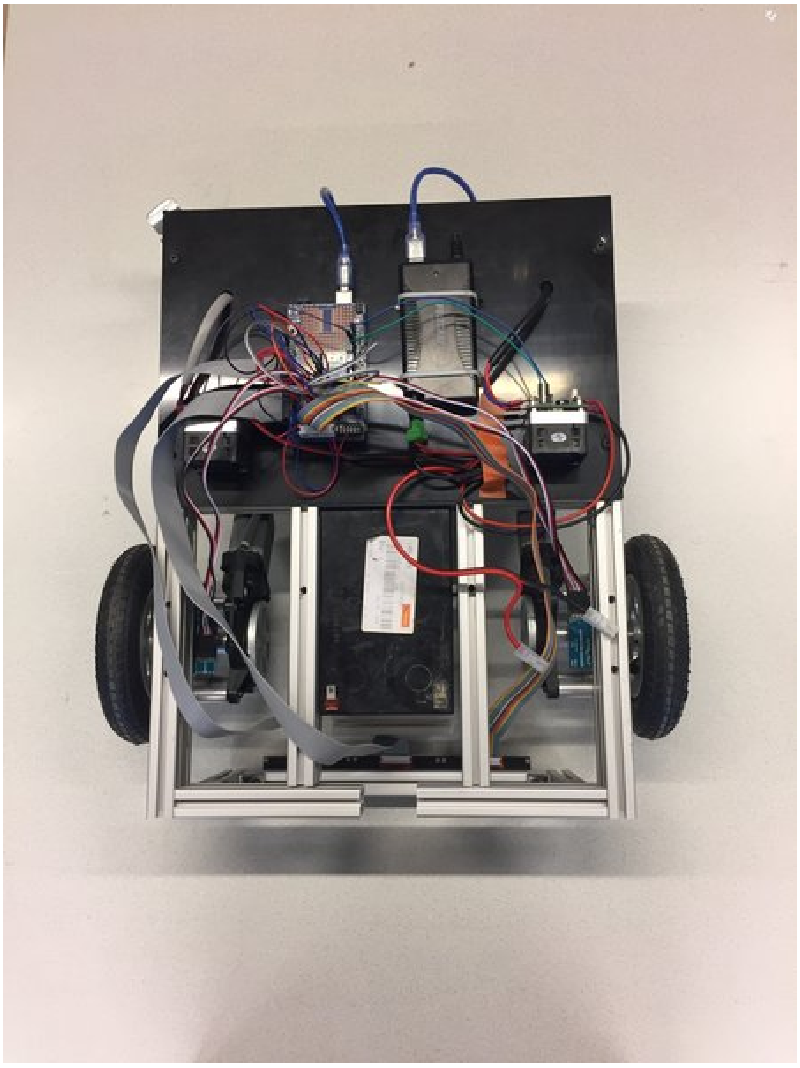

Figure 3.5: The final look of the platform

3.5 Control system

With all the sensors and measuring in place a control system is needed. The platform will be given commands through the main console. The main console is controlled by someone offstage. With the main console commands are send to the multiple platforms. With the use of identifiers it is easy to send a command to single platform, or all platforms. Commands to the platform are put in a queue. After finishing a command, the program will check if there is another command in the queue. It will then perform that command. If the command list is empty the platform becomes idle. Before the show an entire routine can be preset with the use of these command lists

Furthermore the main controller has the ability to interrupt the command or change it. This is necessary for during a show, if a platform doesn't do what it is supposed to do it can quickly be stopped, reset and be put back on track. For this there is a list of control options. When the platform receives such a control from the main console it will interrupt the current action and do the control action. The program is visualized in figure 3.6. Each cycle the system will also update its position by measuring the sensors.

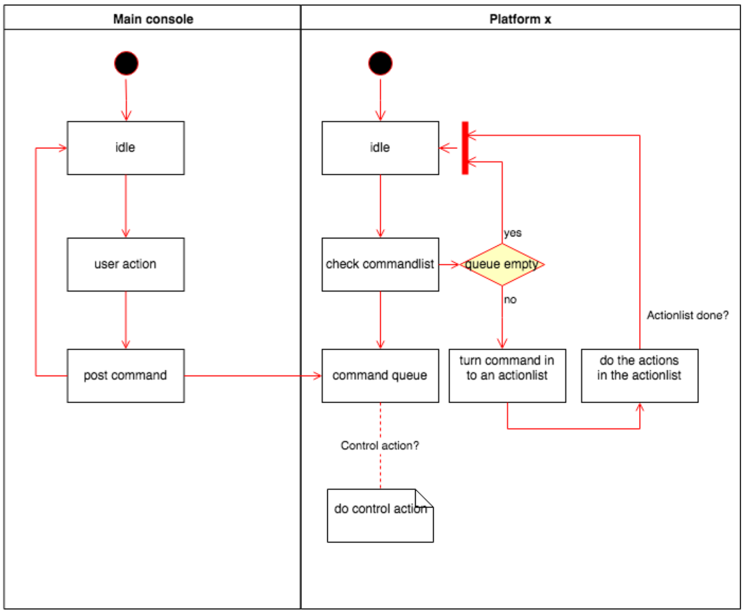

Figure 3.6: The basic control loop.

For a lot of freedom in what the system can do in a performance, the system should have commands for as many different things as possible. In table 3.2 the list of all the commands for the platforms is specified. Not all of them could be implemented due to time constraints. The full list of desired controls can be seen in table 3.3.

When the platform performs a command it will convert the command to a list of actions. For example for move(x,y,θ) the actions are:

1. Calculate needed angle to be able to move straight to the position
2. Turn to the needed orientation
3. Drive forward till the position is reached
4. If an orientation is specified for the endposition, turn to that orientation

The standard procedure for finding tape is depicted in figure 3.7. As can be seen in the figure the platform first goes to a position in front of the tape. It will then turn to the right angle, and make a straight approach. When one of the reflectance sensors on the bottom of the platform detects tape it will try and straighten the platform with the pid controller.

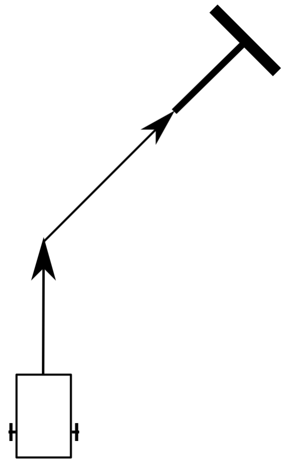

Figure 3.7: The standard procedure from moving from a position to a marked endposition

For the turning command, a simple p controller is used. This controller slows down the platform when it is near the needed orientation. This controller reduces the chance that the platform overshoots, or slips when breaking.

For driving straight to a point a similar system is used. The platform slows down when it is close to the desired point. A problem is that since the motors are not exactly the same, the platform will not drive exactly straight, instead there will a small deviation to one side. To make sure that the platform moves correctly to the desired point, it makes in between calculations of what orientation it needs have to reach the point. For example the platform drives straight from point (0,0) to (500,0). Through some inaccuracy the platform measures that it is on (100,3). Now the error in angle is calculated with $tan^{-1}((0-3)/(500-100)) = 0.43$. This is inserted in a pid controller and the platform is moved to the correct angle.

3.6 Setup

Setting up the system with this design should not take too much effort. On a ballet floor all the needed positions can be marked and taped beforehand. The positions that need tape are the initial positions of the multiple platforms and the end positions. After the ballet floor is placed, the platforms can be placed on their initial positions. They can then be given a preprogrammed list of commands. To then start the program, a start command has to be given to all of the platform to start their program.

Table 3.2: List of commands the main platform has for a platform

Move(x,y,θ)	The platform will go to the position from their current position. It will end in the specified orientation(is optional).
Moveprecise(x, y,θ)	This time the platform will expect tape. So the platform will first go 1m straight in front of the tape and make a straight approach. After stopping it will also assume the specified position and orientation as its real position, fixing drift errors.
Waitfor(trigger)	The platform will not do its next command till the trigger is done. Current triggers are: time(x), the system will wait x amount of seconds. mainsystem(), the system will wait till the main controller gives the sign to go. throw(n), the system will perform it's next command after it has detected n balls being thrown against it.
Turn(t,θ,direction)	Turn for t amount of time or for θ amount of degrees. The system will turn in the specified direction(clockwise or counterclockwise). If both degree and time are specified it will stop after doing one of them.
Turnto(θ)	The platform will turn till it has the orientation θ(in the quickest direction).
Drivestraight(t,d,direction)	Similar to turn only for driving straight. Direction is either backwards of forwards.
Setmotors(m1,m2,t,d)	Direct control of the motors. Will keep the specified power, for t amount of time or if d amount of distance is traveled.

Table 3.3: List of controls the main platform has for a platform

Pause	Pause the system. System will perform no action till it gets the go
Go	Stop pause, and start or continue doing things in the actionlist
Clearlist	Clear the list of commands, also clear the list of actions
Skip	platform will stop it's current command and move to the next
Insert(command)	A command will be placed at the front of the queue
Reset	Resets the entire system

4 Results

4.1 Physical results

After building some details about the platform were measured. These can be found in table 4.1. The max speed is a little lower then desired. The battery has 12Ah, this means the platform can only run on max speed for 1.5 hours. But most of the time the platforms will be static, so the platform should have enough power for an entire performance. The weight of the platform is also under the max allowable weight of the motors(30kg). There still needs to be a sloped platform put on top of it, but as long as that weighs under 20kg the system is fine.

Table 4.1: Details about the platform

Dimensions	45cm x 33cm x 14cm
Weight	10.45kg
Current usage	Idle: 0.4A. Max current usage ~8A
Max speed	0.8m/s

4.2 Positioning results

To test the accuracy of the system it was tested. For this the vehicle started in position (0,0). The system was told to move to (0,500) and after that to (100,500). The system converts this into 4 actions

1. Turn to 90 degrees
2. Drive to(0,500)
3. Turn to 0 degrees
4. Drive to(100,500)

This test was done 5 times to measure the accuracy and reliability of the system. The results are plotted in figure 4.1. The average absolute error in position is 3.7 cm, and the biggest difference is 6.8 cm.

Multiple data of the system was measured during a run. In figure 4.2 the total distance traveled over time per wheel can be seen. The speed of the separate wheels is depicted in figure 4.3. Other interesting data is the measured orientation of the platform in figure 4.4 and the traveled path according to the system figure 4.5. The actual ending position of the platform was (490.3, 105.2). This is more then in the other tests. This is due to the Serial.print function of the Arduino influencing the frequency at which the program runs. The print function approximately decreases the frequency at which the program runs with a factor of 5.

The 4 actions are clearly visible in all the graphs. First the platform is turned counterclockwise to 90 degree. At t = 2 the platform starts driving straight. At a speed of about 0.7m/s. As can be seen in figure 4.5 the platform does not drive exactly straight to the point, it makes a bit of a curve. The platform does end in the correct position. This is due to the program, which fixes errors in positions by moving the platfom to a certain angle. The effect can also be seen in figure 4.4 where the angle slighly oscillates. At t = 9.5 the platforms turn clockwise, and at t = 11.2 the platform drives straight again. Until finally stopping at the endposition.

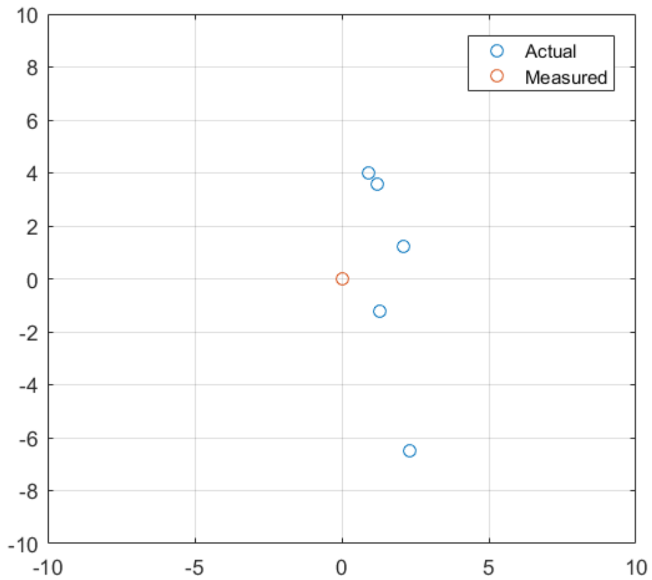

Figure 4.1: The end positions of the platform are marked with blue. The desired end position is marked in red. The x-axis and y-axis are in cm.

4.3 Tape positioning results

To test how well the positioning of the system on the tape works, a test was done. The platform was driven to the tape with 3 different starting deviations. One with a deviation of 10cm, one with -10cm and one with no deviation. Basically one where the left side touches the tape first, then one where the right side touches first and finally one where the center touches first. When the platform stops, the deviation of the wheels are measured compared to the desired position. When the deviation of the both wheels is known the deviation in angle can be calculated with:

$$sin^{-1}((D_L - D_R)/W)$$

With W being the distance between the wheels.

The results are displayed in table 4.2, 4.3 and 4.4. The average error in angle for all results is 0.47 degrees. The biggest outlier has an error of 1.69 degree. It seems that the center hitting first results in less average error in degree. Furthermore the deviation in wheel position is low, under a 1 cm absolute error in all cases.

Table 4.2: Results of the left side hitting the tape first.

	Left wheel deviation in cm	Right wheel deviation in cm	Absolute angle deviation in degrees
Measurements	0.3	0	0.51
	0.5	-0.3	1.35
	0.2	0	0.34
	0.1	0.3	0.34
	0	0	0.00
Average	0.22	0	0.51

Table 4.3: Results of the right side hitting the tape first.

	Left wheel deviation in cm	Right wheel deviation in cm	Absolute angle deviation in degrees
Measurements	-0.7	0	0.1
	0.1	0	0.17
	0.2	0.7	0.84
	0.1	-0.1	0.34
	-0.8	0.2	1.69
Averages	-0.22	0.16	0.84

Table 4.4: Results of the center hitting the tape first.

	Left wheel deviation in cm	Right wheel deviation in cm	Absolute angle deviation in degrees
Measurements	0.1	0.1	0.00
	0.3	0.5	0.34
	-0.3	0	0.51
	0.2	0.4	0.34
	0.4	0.1	0.51
Averages	0.14	0.22	0.34

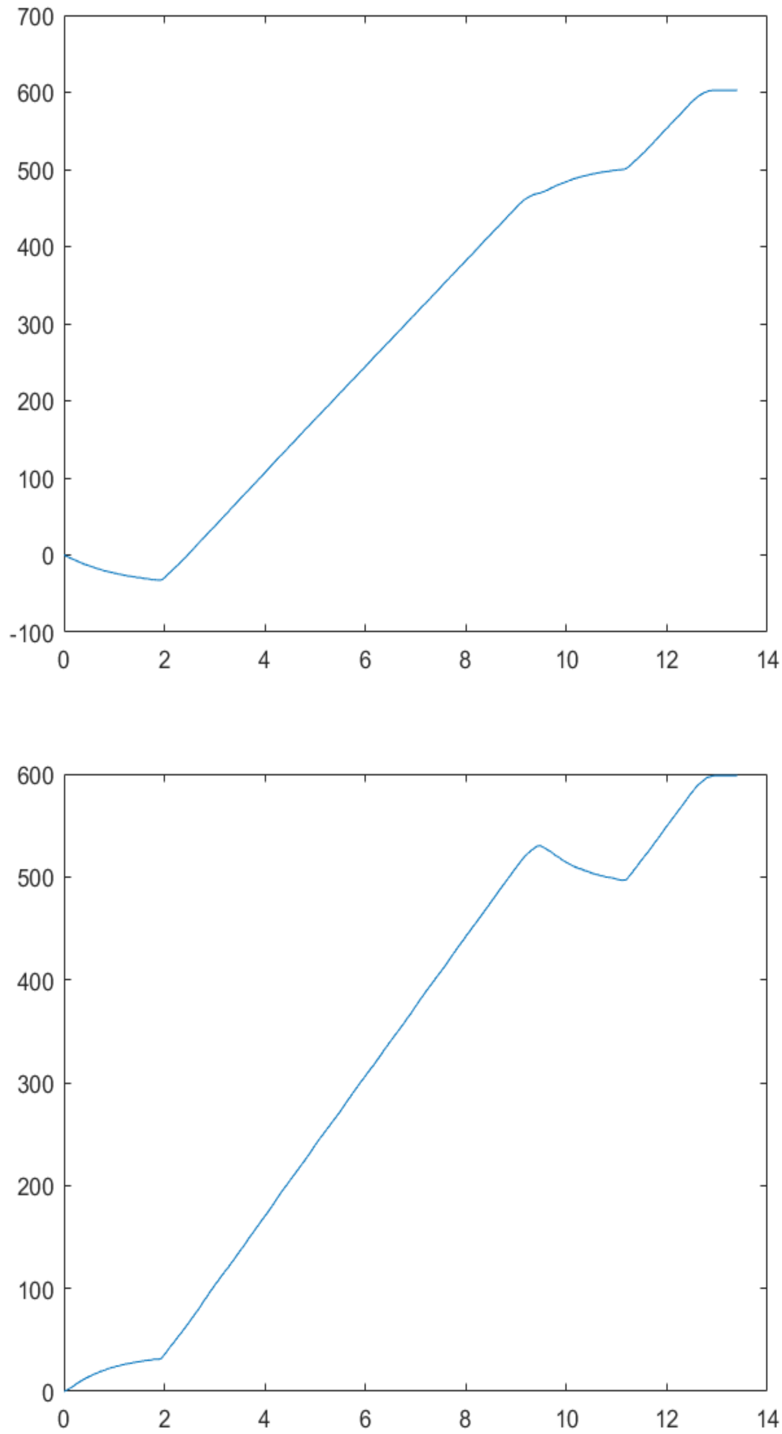

Figure 4.2: Top picture is the distance traveled by the left wheel, bottom picture is the distance traveled by the right wheel. The x-axis is time in seconds and the y-axis distance in centimeter.

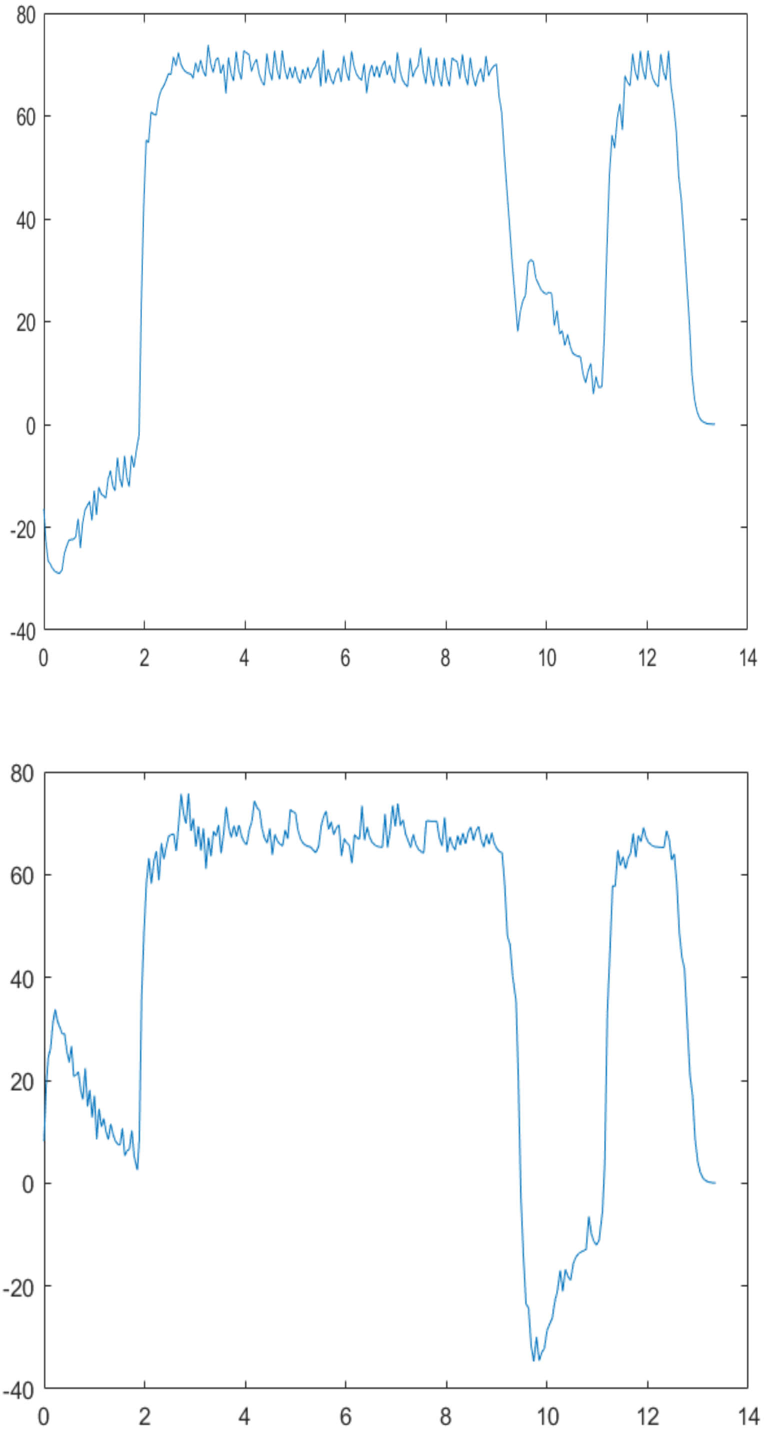

Figure 4.3: Top picture is the speed of the left wheel, bottom picture is the speed of the right wheel. The x-axis is time in seconds and the y-axis distance in centimeter/second.

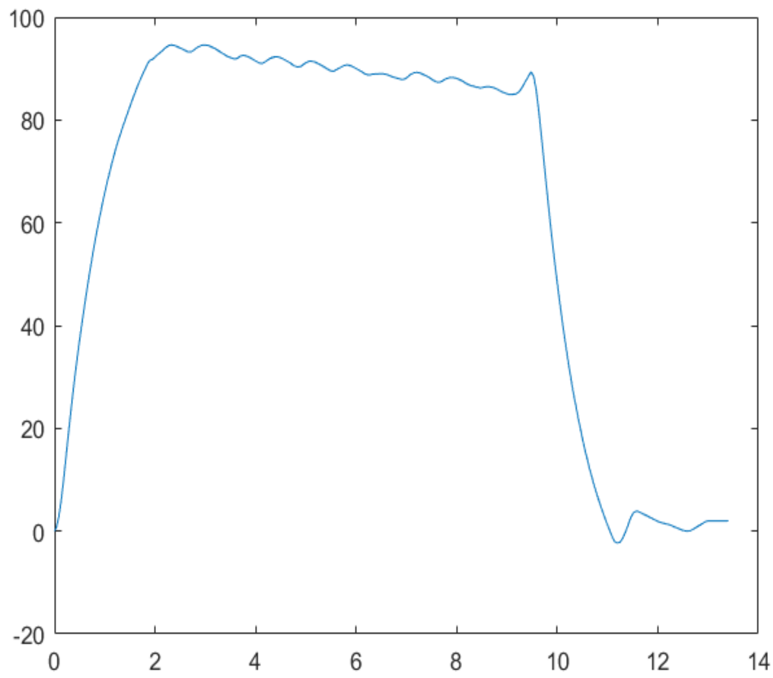

Figure 4.4: The measured orientation of the platform. The x-axis is time in seconds and the y-axis is the orientation in degrees.

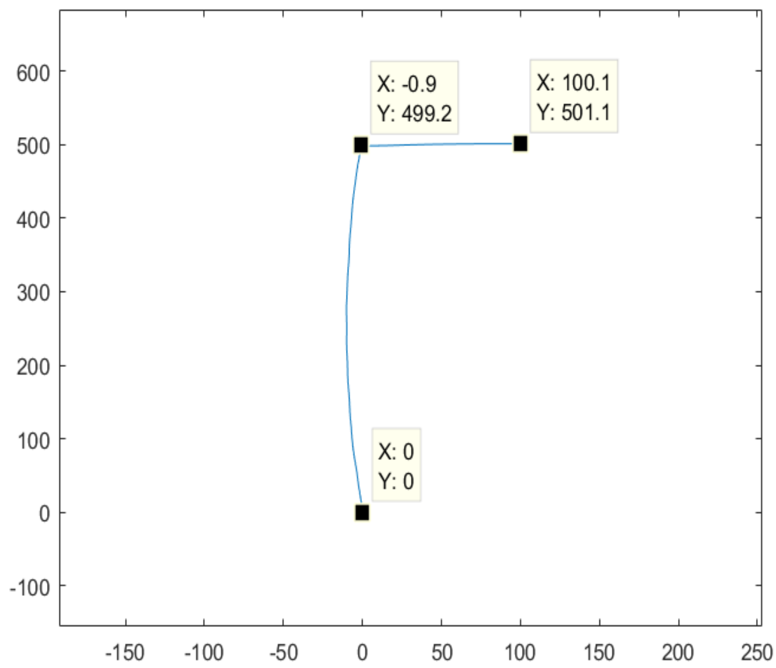

Figure 4.5: The travel path measured by the platform. The x-axis and y-axis are in centimeters.

5 Conclusions

5.1 Conclusions

The system is shown to be capable of positioning on a precise location. The methods used are unaffected by the disruptions from a theater setting. But not all requirements were completely fulfilled. The average error in orientation is 0.47 degrees, which is under the desired 1 degree. But there were outliers with a 1.67 degree error. Because of this there is a chance that the system fails, if the other platforms get the same error. The positional error was better, always within the wanted maximum error of 3 cm after positioning on the tape. Furthermore the positioning system of odometry was always within the maximum allowable 12.5 cm error. So the tape will be hit. Still the error with odometry was higher compared to the possible 30mm accuracy[10]. This is mostly due to the low quadrature encoder disk resolution, and fairly slow sample time. Another issue is that since no absolute positioning could be implemented, the platforms will drift if no tape marker is hit in due time. Furthermore the total time it takes to position over 5 meter and 1 meter is about 14 seconds. This is not slow but it could be faster. Furthermore the platform does have the power capacity to make it through one show. And it can also hold the weight of the ramp put on top of it.

5.2 Recommendations

The made system has promising results, but there is a lot of room for improvement. To improve the accuracy of the odometry, quadrature encoders with a higher resolution should be used. Other improvements are that the sampling rate for measuring should be higher. This can be accomplished by using a faster microcontroller.

The speed at which the vehicle moves can be increased by using more powerful motorized wheels. Furthermore a potential way to reduce the error in orientation on the tape is to use two extra reflectance sensor arrays on the side of the robot, perpendicular to the other reflectance sensors. With this the platforms orientation can be set even more precise. Another thing to improve the positioning is to add an UWB system as discussed before. This would help negate the drift errors of the platforms after long driving without hitting a marker.

Bibliography

[1] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 13–32, First 2009.

[2] R. Mautz, "Overview of current indoor positioning systems," *Geodezija ir Kartografija*, vol. 35, no. 1, pp. 18–22, 2009. [Online]. Available: https://www.tandfonline.com/doi/abs/10.3846/1392-1541.2009.35.18-22

[3] N. Ruangpayoongsak, "Algorithms for indoor positioning systems using ultra-wideband signals," TUDelft, 12 2010. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid:013544ae-6157-4479-bb1e-be81f63461f2?collection=research

[4] H. B. Nissanka B.P., Anit C., "The cricket location-support system," in *2009 Second International Conference on Computer and Electrical Engineering*, aug 2000, pp. 32–43.

[5] C. Z. S. G. Jian li, Guangjie Han, "An indoor ultrasonic positioning system based on toa for internet of things," 2016.

[6] M. Pakdaman and M. M. Sanaatiyan, "Design and implementation of line follower robot," in *2009 Second International Conference on Computer and Electrical Engineering*, vol. 2, Dec 2009, pp. 585–590.

[7] N. Ruangpayoongsak, "Mobile robot positioning by using low-cost visual tracking system," 2017.

[8] J. Kim and S. Lee, "A vehicular positioning with gps/imu using adaptive control of filter noise covariance," vol. 2, no. 1, 2016, pp. 41 – 46, special Issue on Positioning Techniques and Applications. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405959515300849

[9] E. C. Frick, "Mitigation of magnetic interference and compensation of bias drift in inertial sensors." University of Iowa, 2015. [Online]. Available: http://ir.uiowa.edu/etd/5472

[10] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 869–880, Dec 1996.

[11] "Motor mount and wheel kit," Parallax, 4 2013. [Online]. Available: https://www.parallax.com/product/28962

[12] "Scensor dwm1000 module," Decawave. [Online]. Available: https://www.decawave.com/products/dwm1000-module

[13] "Itg-3200 product specification," Invensens inc., 3 2010. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf

[14] A. P. David Tedaldi and E. Menegatti, "A robust and easy to implement method for imu calibration without external equipments," p. 8, 2014. [Online]. Available: https://www.dis.uniroma1.it/~pretto/papers/tpm_icra2014.pdf