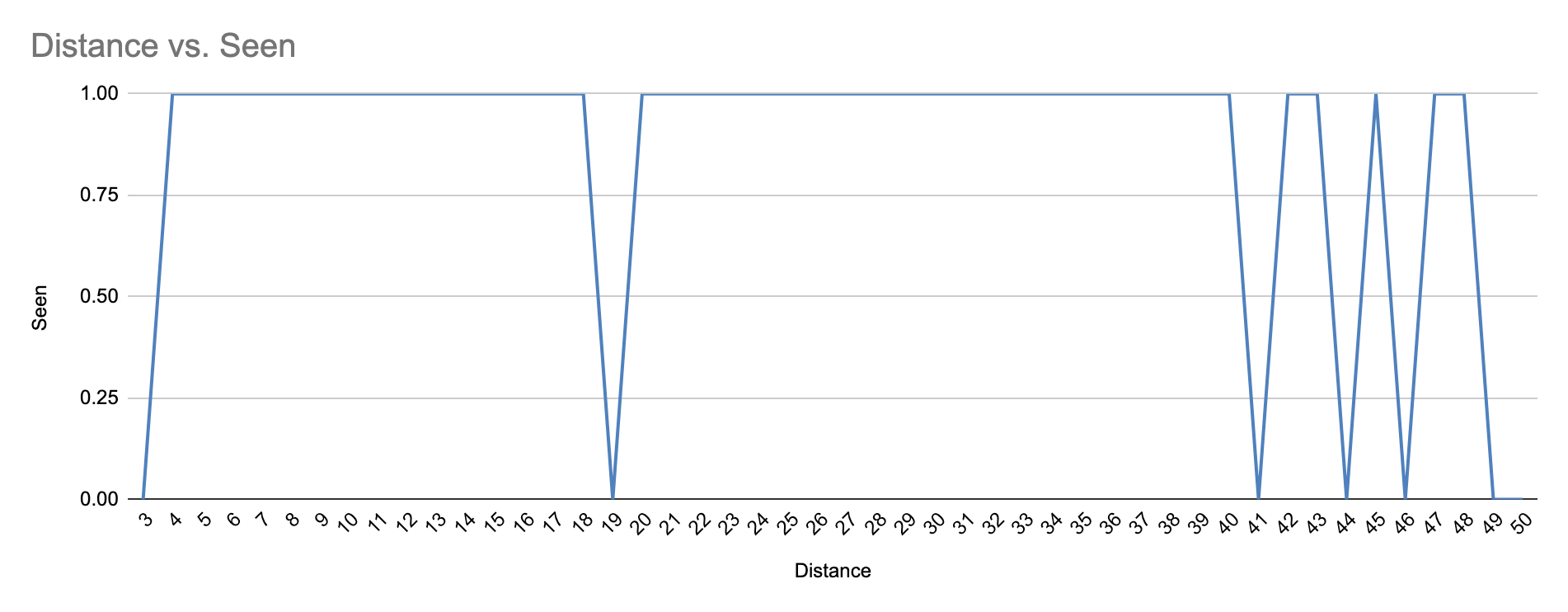
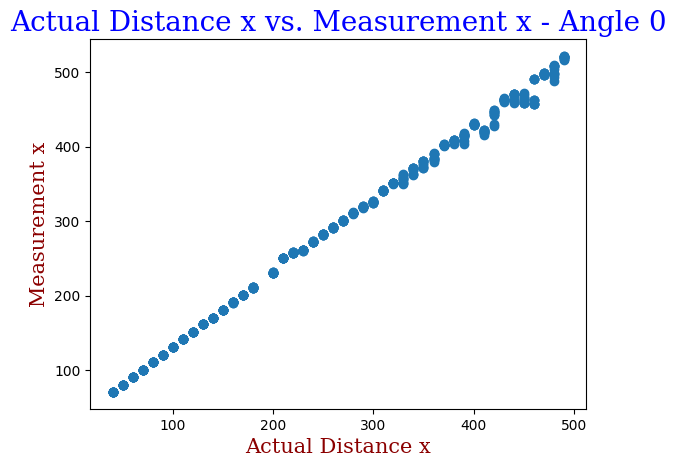
**Report on Sensor Models**

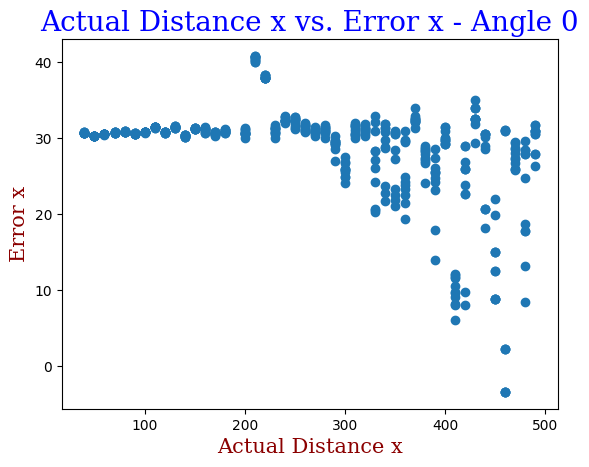
1) Data collection

1. Sensing the presence, and position and orientation of cubes
2. Angle = 0

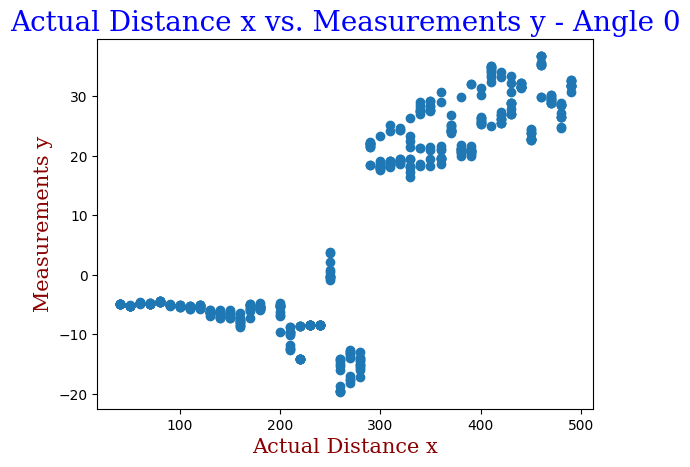
| Angle = 0, 3cm to 50 cm | | |
| --- | --- | --- |
|  | Seen | Not Seen |
| Object = 1 | 0.85 | 0.15 |
| Object = 0 | 0 | 1 |

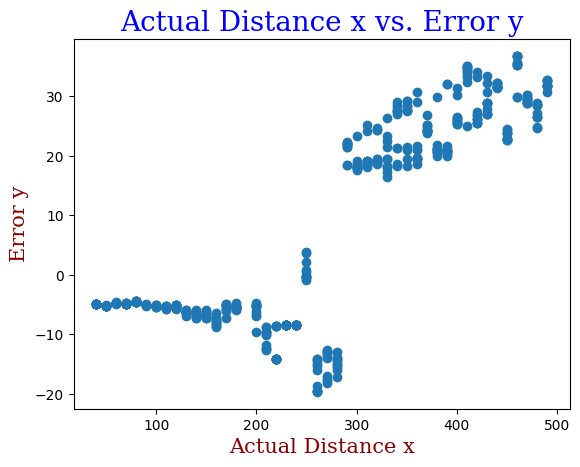
The Anki Cozmo is able to see the cube up to 48 cm when the cube has an angle of 0. With the occasional error where the Cozmo couldn’t see the cube at the distances of 19cm, 41cm, 44cm, and 46cm, however Anki could see the cube up to 48cm. We started measuring the cube from 3cm as the Cozmo couldn’t detect it as it was too close to the sensors. From 3cm to 50cm with an angle of 0 for the cube, the Anki Cozmo was able to detect the cube 0.85% of the time and wasn’t able to detect it 0.15% of the time. However, the Anki Cozmo didn’t detect anything else as the cube so the accuracy of not being able to identify the cube is 1 when there is no cube.



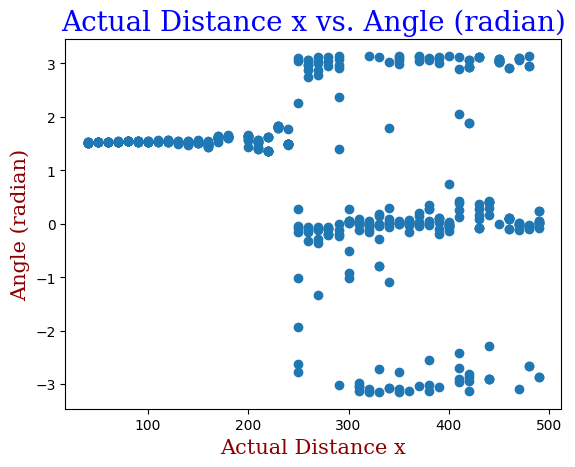


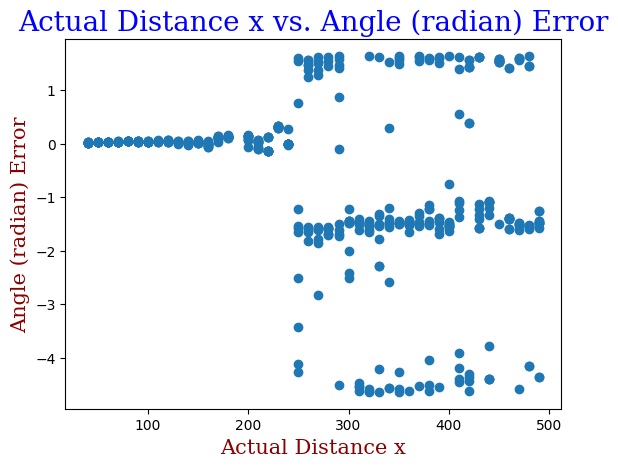
With the cube just being at an angle of 0 there was a high amount of error distribution. There was a measurement error of 30 mm from 0 mm to 200 mm, after 200 mm the error distribution jumps to 40 mm. It can be said that Cozmo has approximately 30 mm systematic error. From 250mm to 400mm the error distribution varies between 35 mm and 15 mm, between 400mm and 500mm the error distribution varies from 0 mm up to 35 mm with the highest amount of errors occuring.



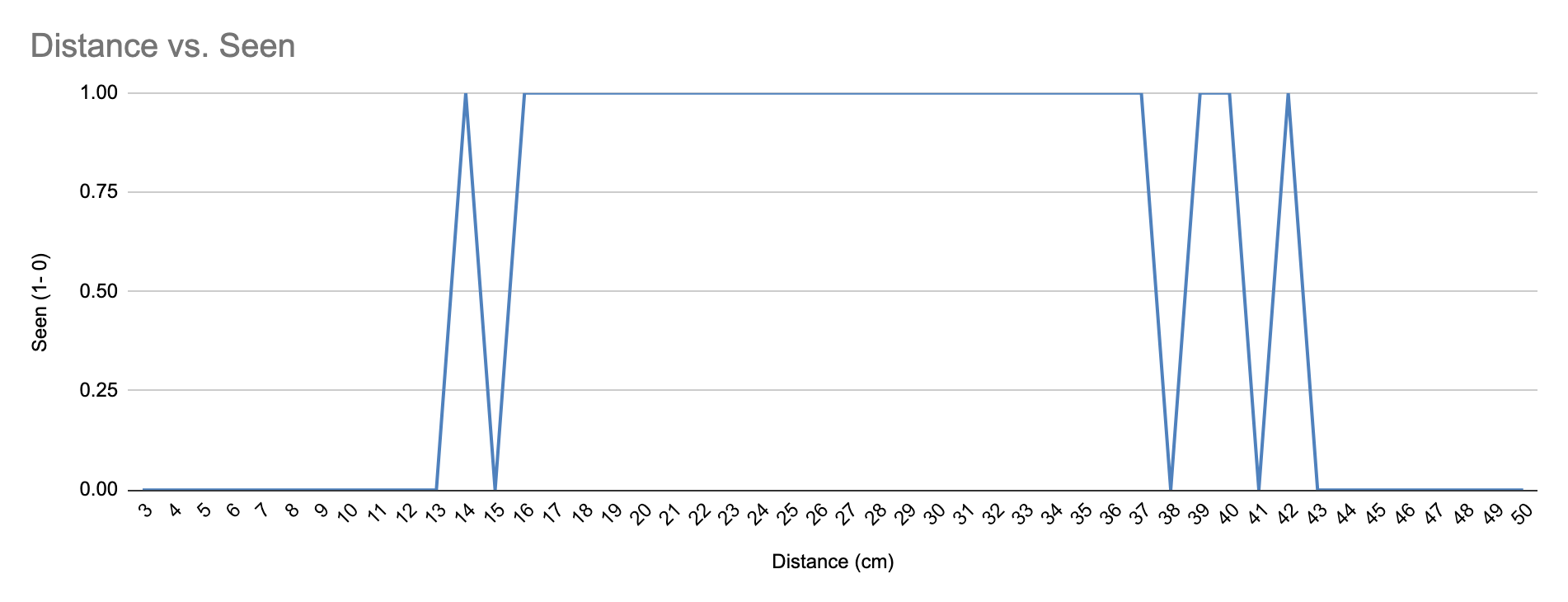


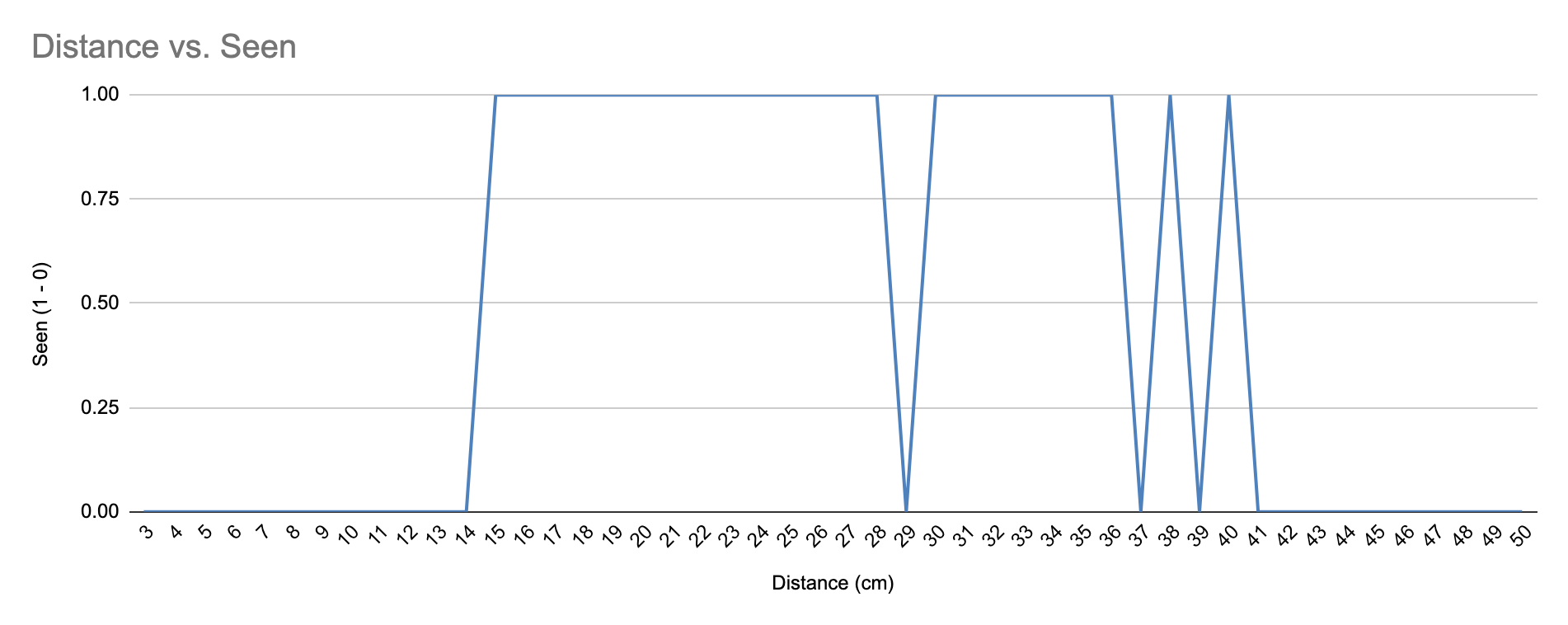
For the y axis, it is clearly seen that before 280 mm Cozmo tends to measure lower than actual and error changes from -20 mm to -5 mm. After 280 mm, the robot's measurements instantly increased and error changes between 18 mm to 40 mm.

Cozmo can measure the angle accurately up to 250 mm with error of almost 0. After this distance, angle varies between -3 radians, 0 radians and 3 radians. It is obvious that Cozmo vision cannot distinguish images clearly after 250 mm. 

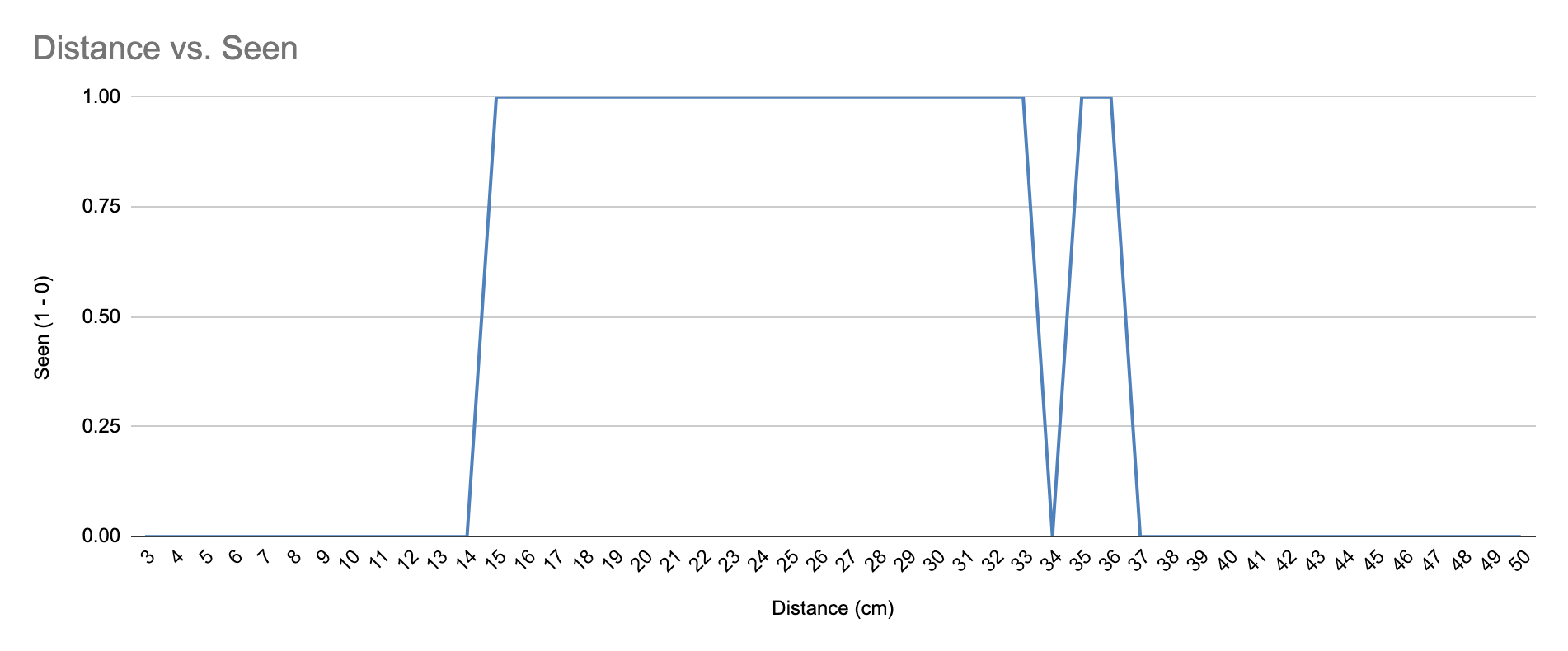


1. Angle = 10⁰

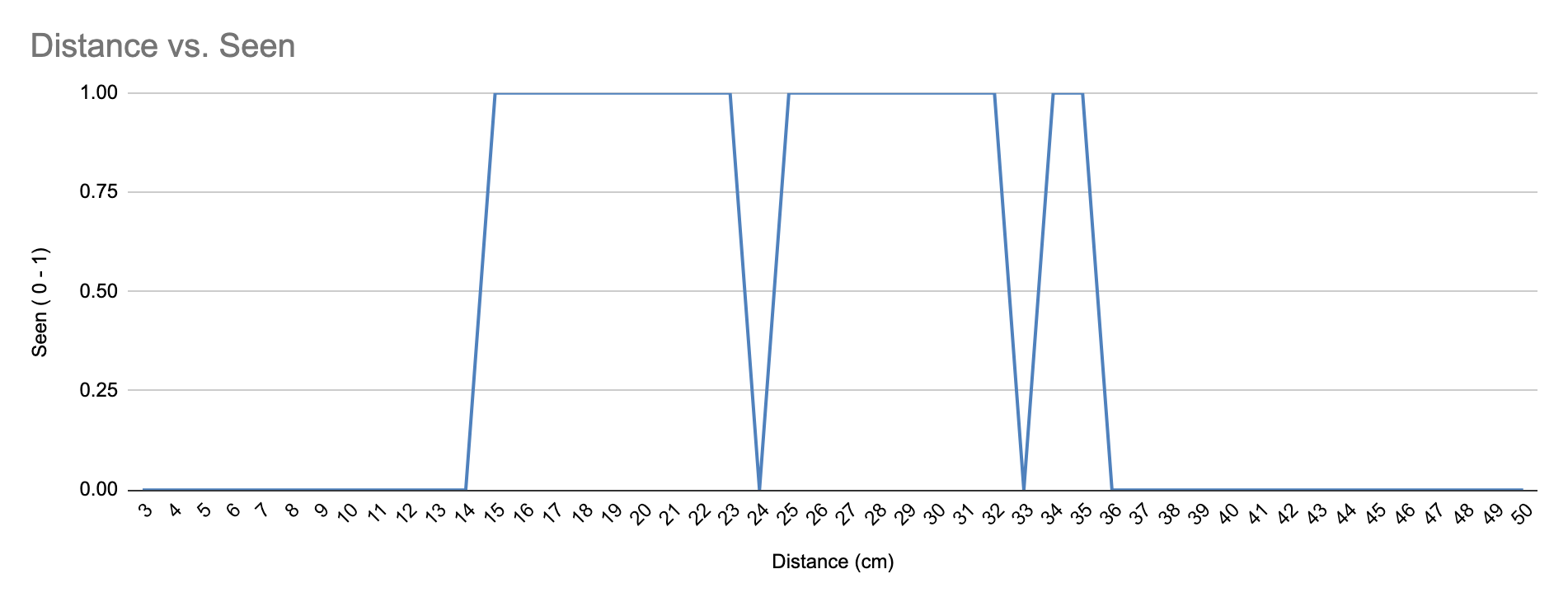
Unlike the cube at an angle of 0 degrees, the Anki Cozmo first detects the cube at 14 cm distance as the cube is at an angle of 10 degrees. Cozmo can only detect the cube up to 42 cm, there are times when anki cannot see the cube for instance up to 14 cm and couldn't see the cube at specific distances such as 15cm, 38cm, 41cm and from 43cm until 50cm. Similarly, to the cube at a 0 degree angle we started testing at 3cm however Anki didn't detect the cube until 14 cm away. From 3cm to 50cm with the cube at an angle of 10 degrees, the Anki Cozmo was able to detect the cube 0.54% of the time, meaning 0.46% of the time it wasn’t able to detect it. However, the robot didn’t detect anything else as the cube so the accuracy of not being able to identify the cube is 1 when there is no cube. 

1. Angle = 20⁰

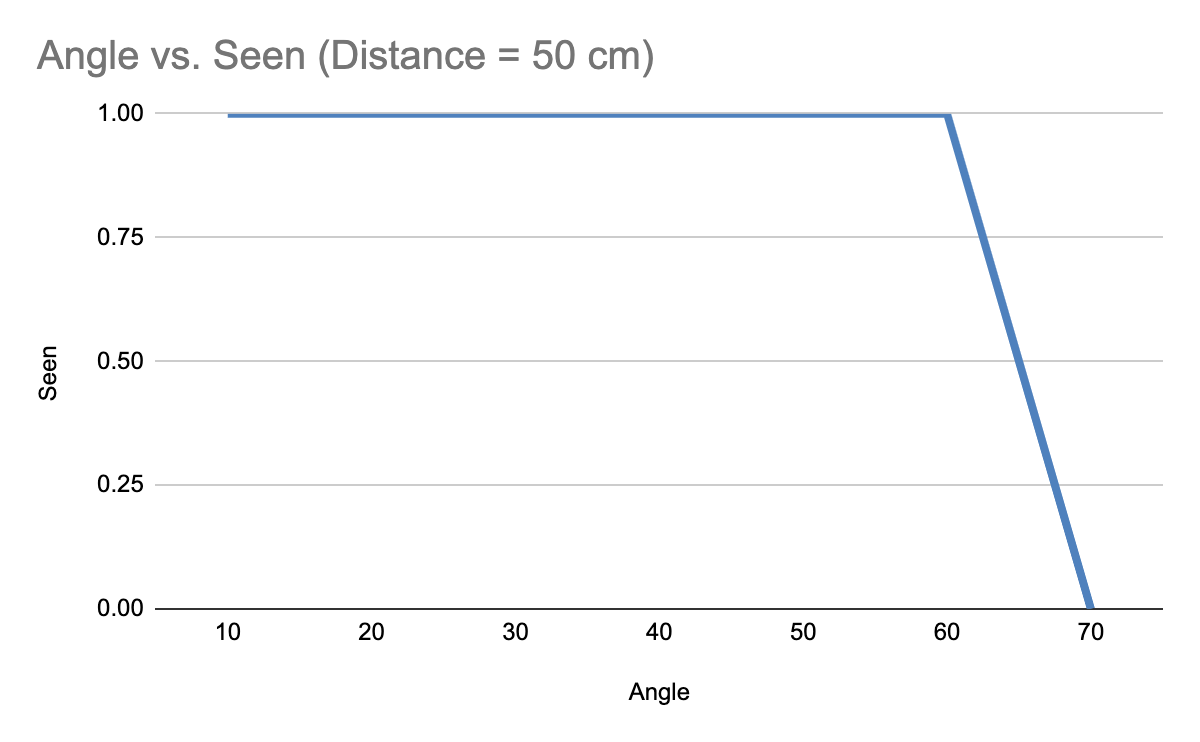
With the cube being at an angle of 20 degrees Cozmo was first able to detect the cube at a distance of 15cm and was able to detect it up to 40 cm. There are specific distances when the Anki Cozmo cannot see the cube; for instance, up to 15 cm the cube cannot be detected, at 29cm, 37cm, 39cm and from 41cm until 50cm. We started testing at 3cm but Anki couldn’t detect the cube until the distance was 15cm. From 3cm to 50cm with the cube at an angle of 20 degrees, the Anki Cozmo was able to detect the cube 0.48% of the time, meaning 0.52% of the time it wasn’t able to detect it which is worse than the 2 previous tests with the cube being 0 and 10 degrees. However, the Anki Cozmo didn’t detect anything else as the cube so the accuracy of not being able to identify the cube is 1 when there is no cube.

1. Angle = 30⁰

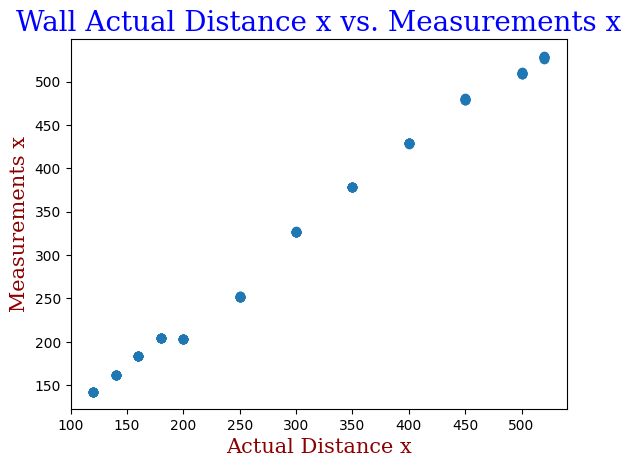
With the cube being at an angle of 30 degrees the robot was first able to detect the cube at a distance of 15cm and was able to detect up to 36 cm. There are specific distances when it cannot see the cube; for instance, up to 15 cm the cube cannot be detected, 34 cm and from 37 cm until 50 cm. We started testing at 3cm but Anki couldn’t detect the cube until the distance was 15cm. From 3cm to 50cm with the cube at an angle of 30 degrees, the Anki Cozmo was able to detect the cube 0.44% of the time, meaning 0.56% of the time it wasn’t able to detect it which is worse than the previous 3 tests with the cube being 0, 10 and 20 degrees. Each time the cube rotates 10 degrees more the detection rate gets worse. However, the Anki Cozmo didn’t detect anything else as the cube so the accuracy of not being able to identify the cube is 1 when there is no cube.

1. Angle = 40⁰

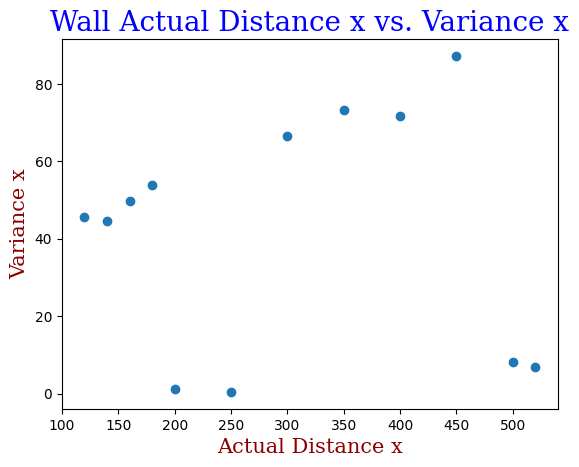
With the cube being at an angle of 40 degrees the anki Cozmo was first able to detect the cube at a distance of 15cm and was able to detect up to 35cm. There are specific distances when the Anki Cozmo cannot see the cube; for instance, up to 15 cm the cube cannot be detected, 24cm, 33cm and from 36cm until 50cm. We started testing at 3cm but Anki couldn’t detect the cube until the distance was 15cm. From 3cm to 50cm with the cube at an angle of 40 degrees, the Anki Cozmo was able to detect the cube 0.40% of the time, meaning 0.60% of the time it wasn’t able to detect it which is worse than the previous 5 tests with the cube being 0, 10, 20 and 30 degrees.

1. Sensing the presence, and position and orientation of markered custom walls 

This graph represents the angle at which the Cozmo stops detecting a custom marked wall. At a distance of 50cm, the Cozmo stops seeing the wall when the angle reaches 70 degrees.



This graph demonstrates the relationship between the actual distance of a wall and the measured distance at which the Cozmo robot can see it. The data appears to be quite accurate, forming a clear and consistent line on the graph, indicating a reliable correlation between the actual and measured distances.



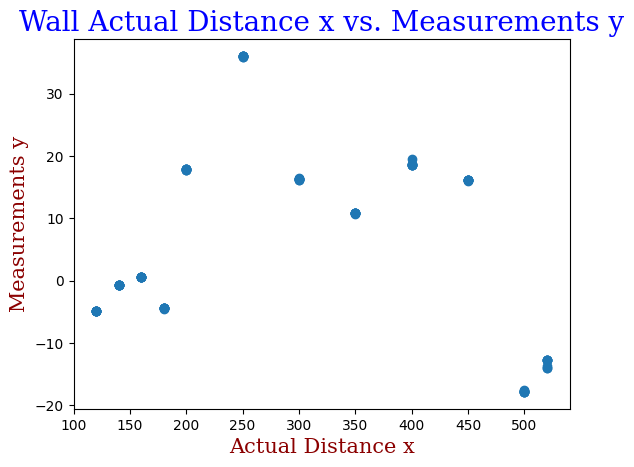
This graph depicts the actual distance of the wall at various x positions and the corresponding variance in the Cozmo robot’s ability to see it.The variance is relatively low for distances between 100 mm and 180 mm, where it reaches 50. It decreases to nearly 0 at distances between 200mm and 250mm. Subsequently, the variance gradually increases as the distance extends up to 470mm, peaking at 80. Beyond 500 mm, the variance decreases to around 10.

The Cozmo robot seems to have an optimal range for accurate detection, which falls between 200 mm and 250 mm, where the variance is minimal. This range may be ideal for consistent and reliable detection.

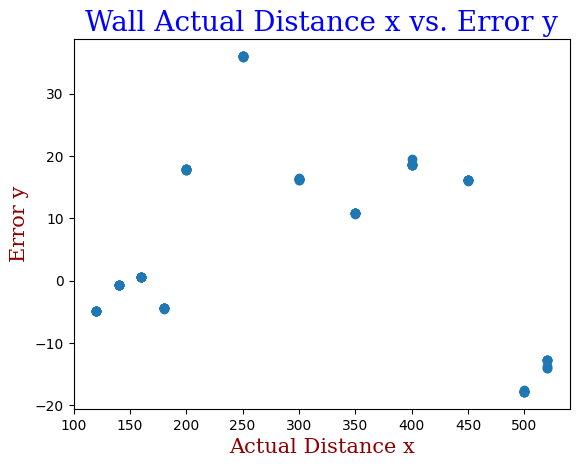
The Cozmo robot seems to struggle to accurately perceive objects which are very close to it.

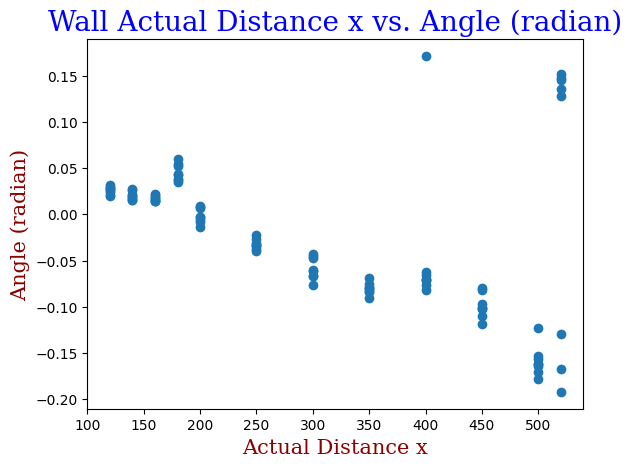
The drop in variance at 500mm might suggest that the robot has a more stable detection performance at this distance compared to the broader range between 250mm and 470mm.

While we measured the wall distance for 50 times, Cozmo could not sense the wall 48 times out of 50 with accuracy of 0.90. However, the Anki Cozmo didn’t detect anything else as the wall so the accuracy of not being able to identify the wall is 1 when there is no wall.



These graphs indicate that Cozmo's ability to sense and measure y-axis values is more accurate when the wall is at distances closer than 200 mm and further than 500 mm. In these two specific distance ranges, Cozmo exhibits better performance and reliability in its y-axis measurements. This suggests that Cozmo's sensors or detection capabilities may have optimal accuracy at both very close and relatively far distances, with less accurate measurements in the intermediate distance range (between 200 mm and 500 mm).

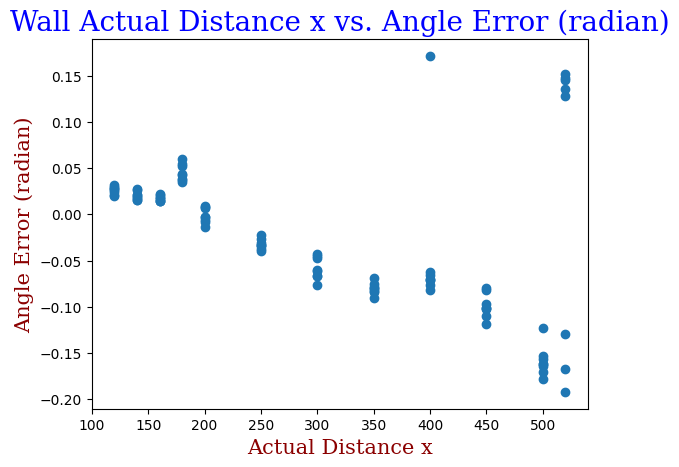




This graph displays the relationship between the actual distance of a wall and the angle error in Cozmo's ability to see it. It reveals that as the actual distance increases, the angle error tends to go into negative values, suggesting an underestimation of the angle.

However, there are two notable exceptions:

1. At a distance of 400mm, the angle error temporarily increases significantly before returning to negative values. This indicates a sudden spike in angle error at this specific distance.
2. Similarly, at 550mm, the angle error experiences a substantial increase and does not recover to negative values, indicating a persistent and significant angle error at this distance.



These graphs suggest that Cozmo's reliability for angle measurements is generally high, as indicated by the consistent negative angle errors at various distances. This consistency implies that Cozmo tends to provide accurate angle measurements in relation to the actual distance of the wall, except for the mentioned anomalies at 400mm and 550mm. These anomalies may be isolated incidents or related to specific factors affecting the measurements at those distances. Overall, the graph suggests a reliable and consistent performance in terms of angle measurements for Cozmo.

1. Sensing cliffs

In order to test the cliff detection, we started with 0 degrees with 80 measurements. Cozmo detected the cliff 78 times out of 80 trials. We realised that the depth of the cliff affects the robot's senses. Cozmo detected the cliff 12 times at 3 cm and 3 times at 2.5 cm. We tested the robot with different angles to the cliff and it identifies cliffs with high accuracy until 40 degrees.

|  |  |  |
| --- | --- | --- |
| Angle = 0, 80 measurements | | |
|  | Cliff Seen | Cliff Not Seen |
| Cliff = 1 | 0.975 | 0.025 |
| Cliff = 0 | 0.9875 | 0.0125 |

|  |  |  |
| --- | --- | --- |
| Angle = 10, 15 measurements | | |
|  | Cliff Seen | Cliff Not Seen |
| Cliff =1 | 1 | 0 |
| Cliff = 0 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| Angle = 20, 15 measurements | | |
|  | Cliff Seen | Cliff Not Seen |
| Cliff =1 | 1 | 0 |
| Cliff = 0 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| Angle = 30, 15 measurements | | |
|  | Cliff Seen | Cliff Not Seen |
| Cliff =1 | 0.93 | 0.07 |
| Cliff = 0 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| Angle = 40, 15 measurements | | |
|  | Cliff Seen | Cliff Not Seen |
| Cliff =1 | 0.67 | 0.33 |
| Cliff = 0 | 0.07 | 0.93 |

|  |  |  |
| --- | --- | --- |
| Angle = 45, 15 measurements | | |
|  | Cliff Seen | Cliff Not Seen |
| Cliff =1 | 0.60 | 0.40 |
| Cliff = 0 | 0.07 | 0.93 |

* 2) Sensor Model Implementation
* We used the previously created map - loadU08520Map and updated the map landmarks including a custom wall.

1. Cliff Sensor Model:

As Cozmo can sense the cliff with accuracy of 0.975 when there is a cliff and 0.025 when there is no cliff. We updated the sensor model with these values.

1. Cube Sensor Model:

We used the data collected in order to calculate variances (x, y, angle) for cube and wall. In order to calculate the variance for multiple measurements of different actual values, we used the Gaussian.py file. This value provides a measure of the spread or dispersion of the measurements from their actual values. The sensor model calculates the probability of the cube being at an actual position when a cube is sensed with measured position.

1. Wall Sensor Model:

Before using the sensor model for cubes, we added a wall to the map with CustomObjectType.007 - CustomObjectMarkers.Triangle3 attributes. For the wall sensor model, we used the collected data to calculate variances for x, y and angle respectively. In order to calculate the variance for multiple measurements of different actual values, we used the Gaussian.py file. This value provides a measure of the spread or dispersion of the measurements from their actual values. The sensor model calculates the probability of the wall being at an actual position when a wall is sensed with measured position.

Sensor model explanation:

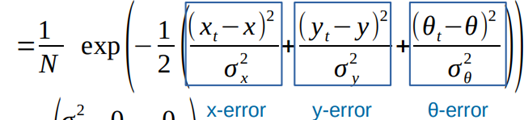
The sensor model code is essential for estimating the probability of a chosen parameter being detected by the sensors. The code is written in Python and encompasses the functions cliff\_sensor\_model, cube\_sensor\_model, and wall\_sensor\_model, all of which are then used in run\_sensor\_model to get the overall probability.

The Gaussian distribution is used for this sensor model due to its ability to represent the uncertainty in the measured positions accurately. The diagonal covariance, with its variances along different axes, allows for the incorporation of independent uncertainties in the measurements. This is crucial for capturing the inherent noise and errors associated with real-world sensor measurements, making the Gaussian distribution a suitable choice for modelling the probabilistic aspects of the sensor data.

The cliff\_sensor\_model function evaluates the likelihood of a cliff being detected by the robot's sensors based on its current pose. It takes three parameters: robotPose, m, and cliffDetected. robotPose represents the current pose of the robot, m represents the map of the environment, and cliffDetected is a boolean value indicating whether a cliff is detected or not. This function checks the validity of the robot's position in relation to the map and subsequently calculates the probability of the cliff being detected, checking if both the robot's pose and the sensor's pose are within the map, with the probability returning as 0 if not. The returned probability of 0.975 or 0.025 indicates the confidence level of the cliff detection if in map.

Moving on to the cube\_sensor\_model function, it determines the probability of a cube being visible and detected at a specific position relative to the robot. The code employs the maximum likelihood to calculate the probability. The function accounts for the errors in the measured position and the true position of the cube, taking into account trueCubePosition, visible, and measuredPosition parameters.

The code first sets p\_vis to 0.85, indicating that there is an 85% chance of the cube being seen by the robot. The variances used were gathered from data measured previously with Cozmo, whereas the errors were calculated from the maximum likelihood formula below, taking into account the x, y and angle dimensions. The probability is then calculated from the formula, where N is the normalisation factor. p\_vis and p\_dis are then multiplied, to get the overall chance of detecting the cube.

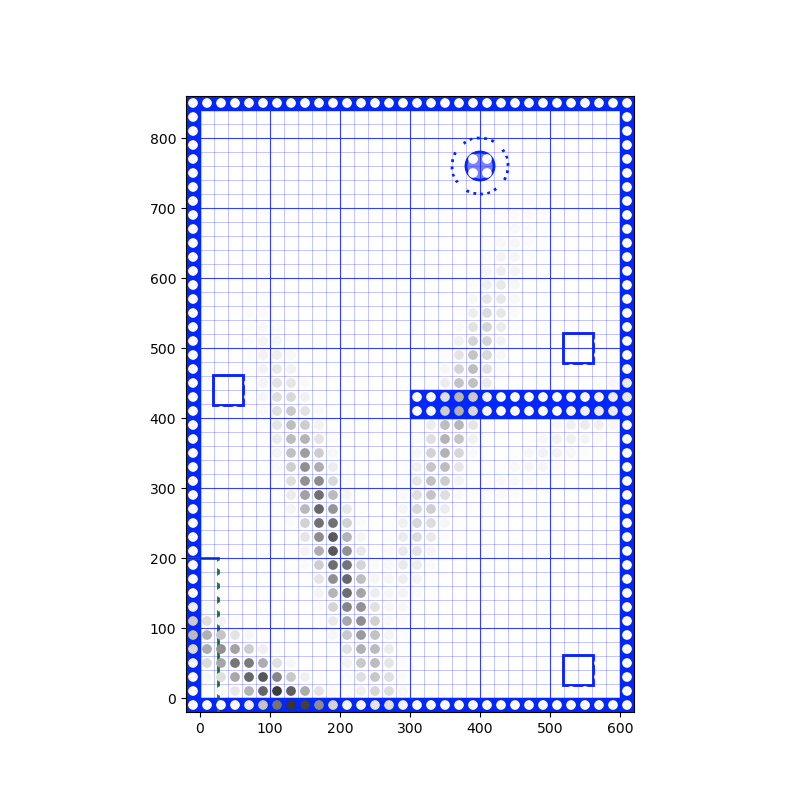


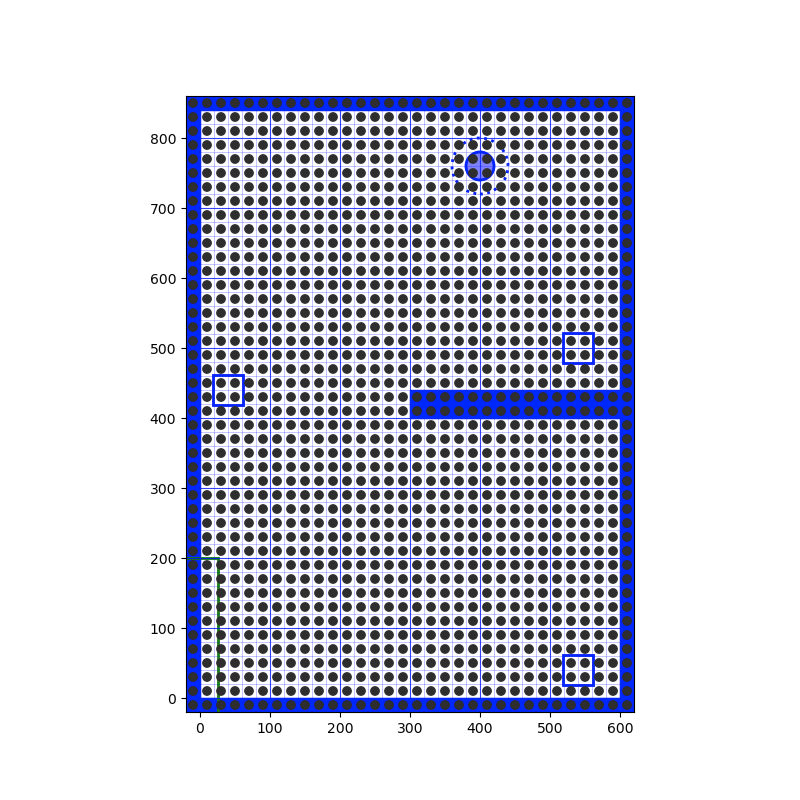
Similarly, the wall\_sensor\_model function calculates the probability of a wall being visible and detected at a specific position relative to the robot, also making use of the maximum likelihood, considering the errors between the measured position and the true position of the wall. The variances and errors were found the same way as for the cube sensor, with the same cal\_variance function being used.

Finally, the run\_sensor\_model function incorporates the results from the cliff, cube, and wall sensor models to compute the overall probability of the sensed environment. By combining the probabilities from each sensor model, this function provides a comprehensive assessment of the surrounding environment, considering both the visibility and the accuracy of detection for various elements such as cliffs, cubes, and walls.

Evaluation of the model:

1. The Cube Sensor Model can show likelihood estimation but the variation is so high that’s why the result is 3 different distributions around the actual position of the cube.





When cosmo cannot see a cube, the likelihood is 1 for the whole map.

1. Wall Sensor Model:

Despite our combined efforts, a successful wall sensor model that reliably detects walls has remained undone. The task has proven to be exceptionally challenging, and despite our best attempts, we have not been able to develop a functioning sensor model for this purpose. This shows the complexity and unique challenges involved in creating a wall detection sensor model.