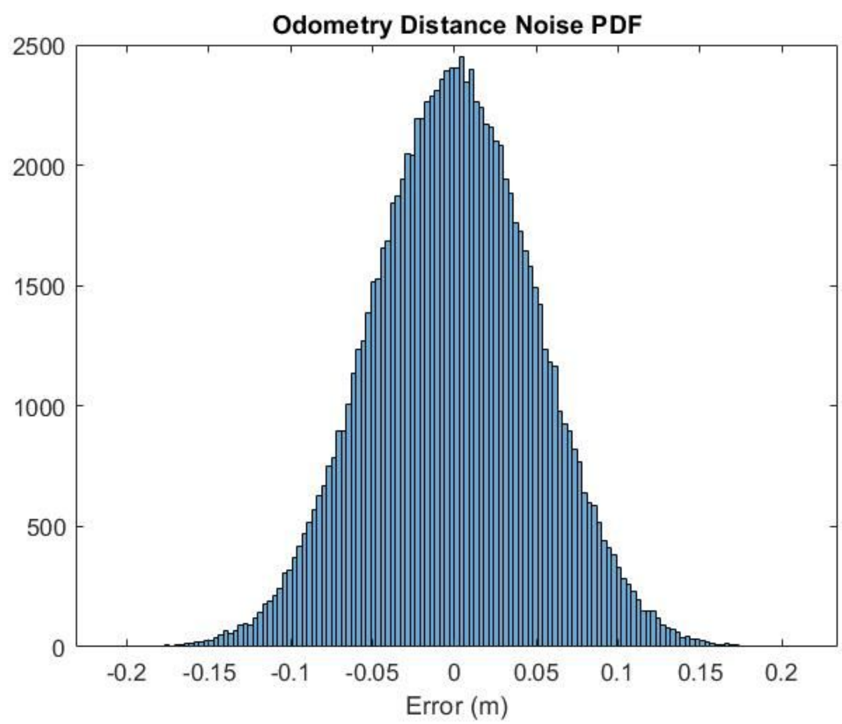
EBS 289K - Agricultural Robotics and Automation
Spring, 2019
Final Project - Nursery

Team: Bennet Evans
Guilherme De Moura Araujo
Nicolas Buxbaum

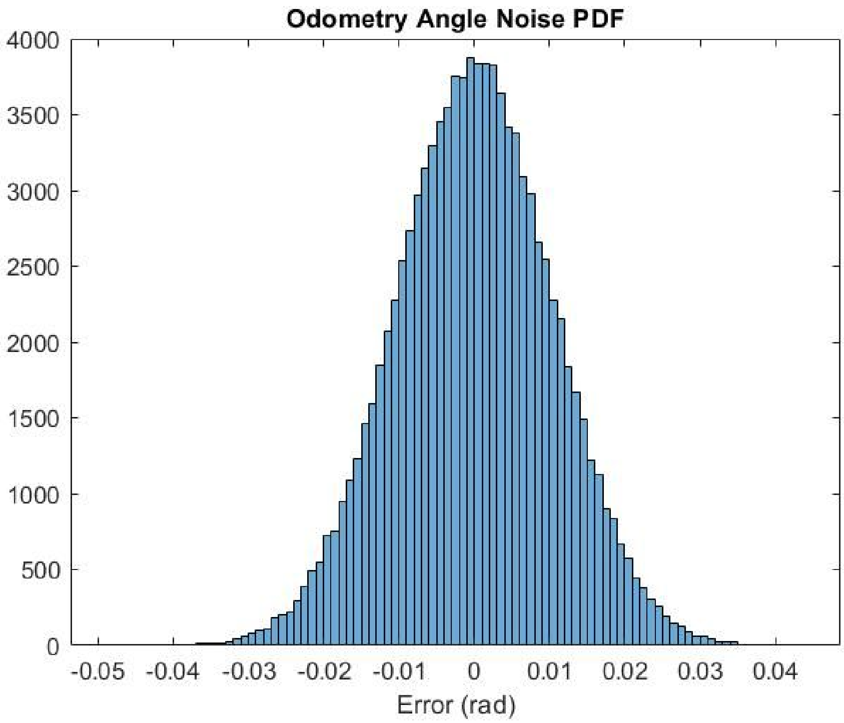
Instructor: Stavros G. Vougioukas

Estimating Sensor Noise:

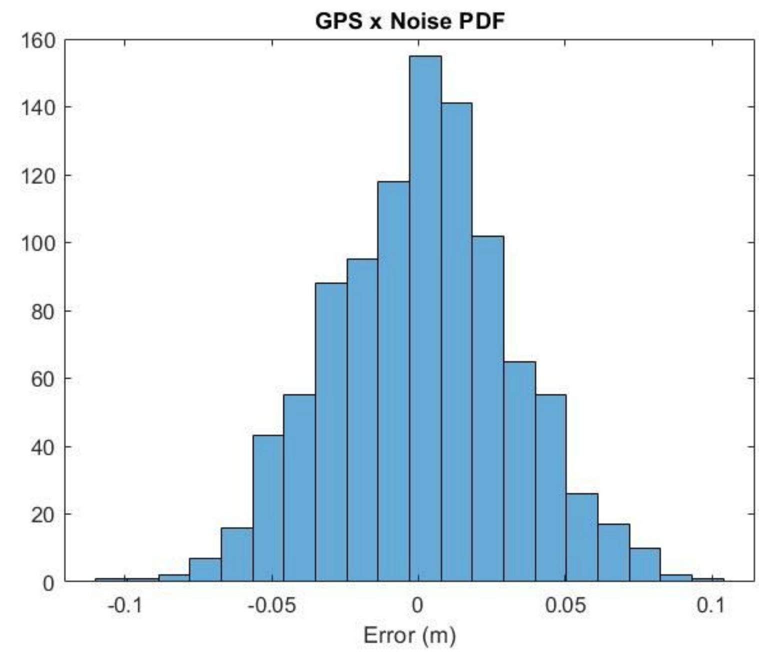
The script “covarianceCalculation.m” was used in order to estimate the covariance matrices of the odometer and GPS. The script runs “robot_odo” and “GPS_CompassNoisy” using a random driver. The difference between the sensor outputs and ground truth for each iterative step is calculated. The variance of this difference for each sensor output is calculated, and is used as an estimate of actual sensor noise. Sample error distributions for each sensor are shown below with their standard deviations.



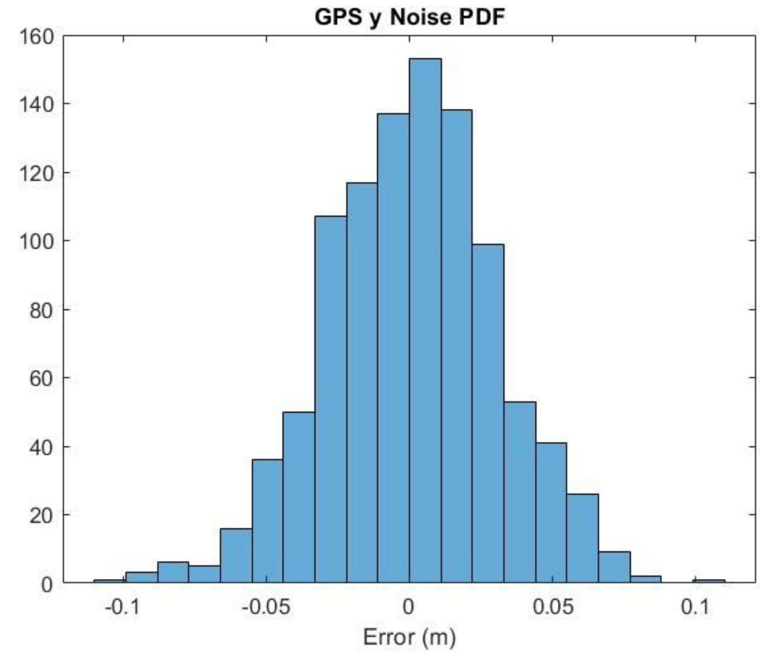
Stand Deviation = 0.0501 m



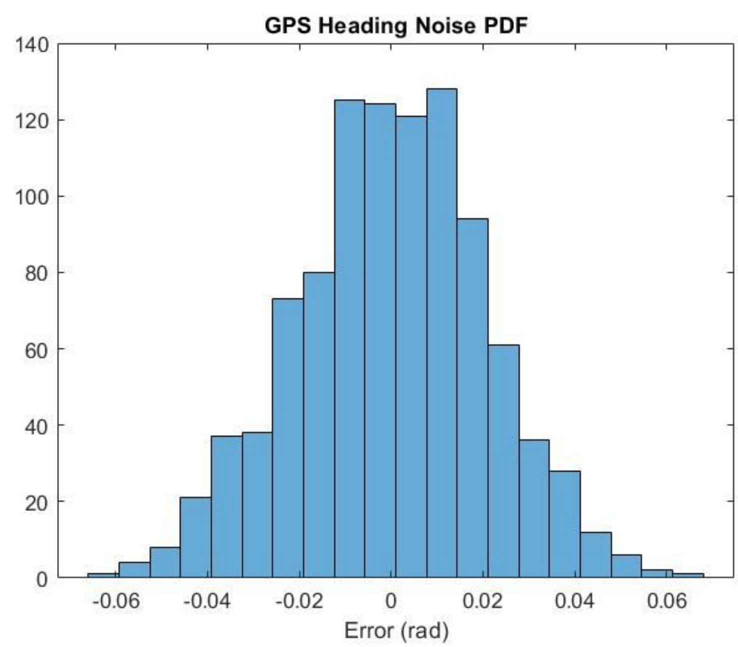
Standard Deviation = 0.0104 radians



Standard Deviation = 0.0311 m



Standard Deviation = 0.0296 m



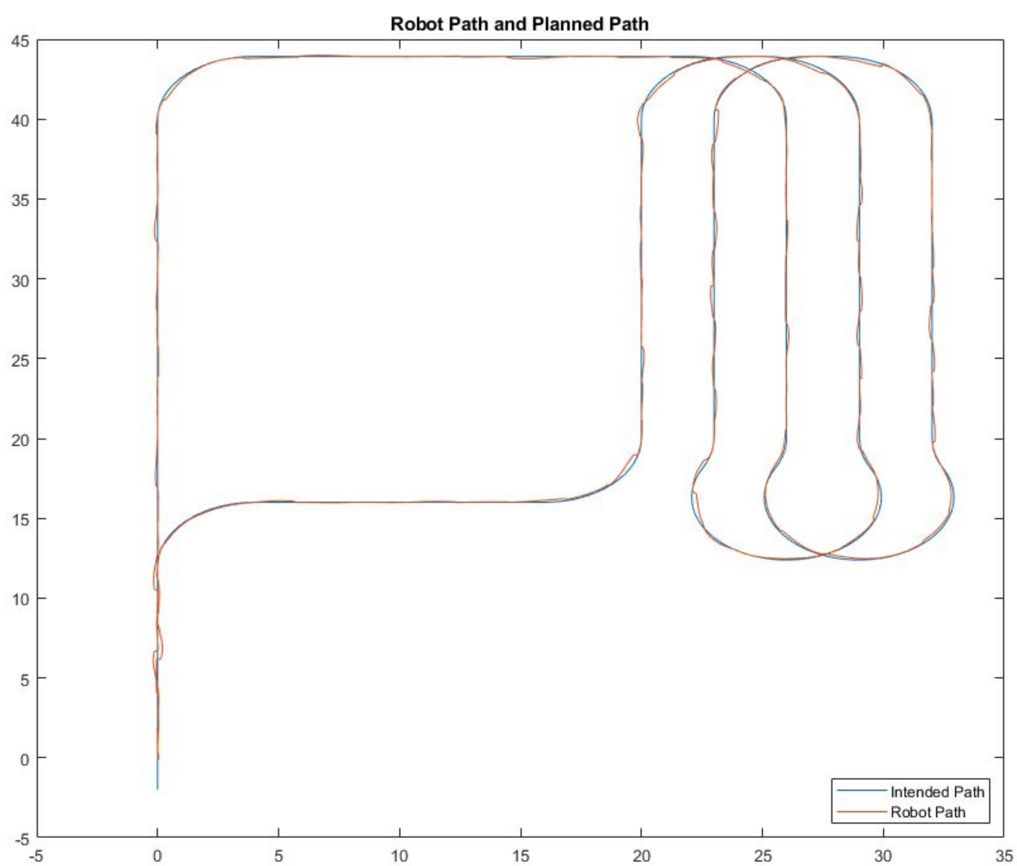
Standard Deviation = 0.0206 radians

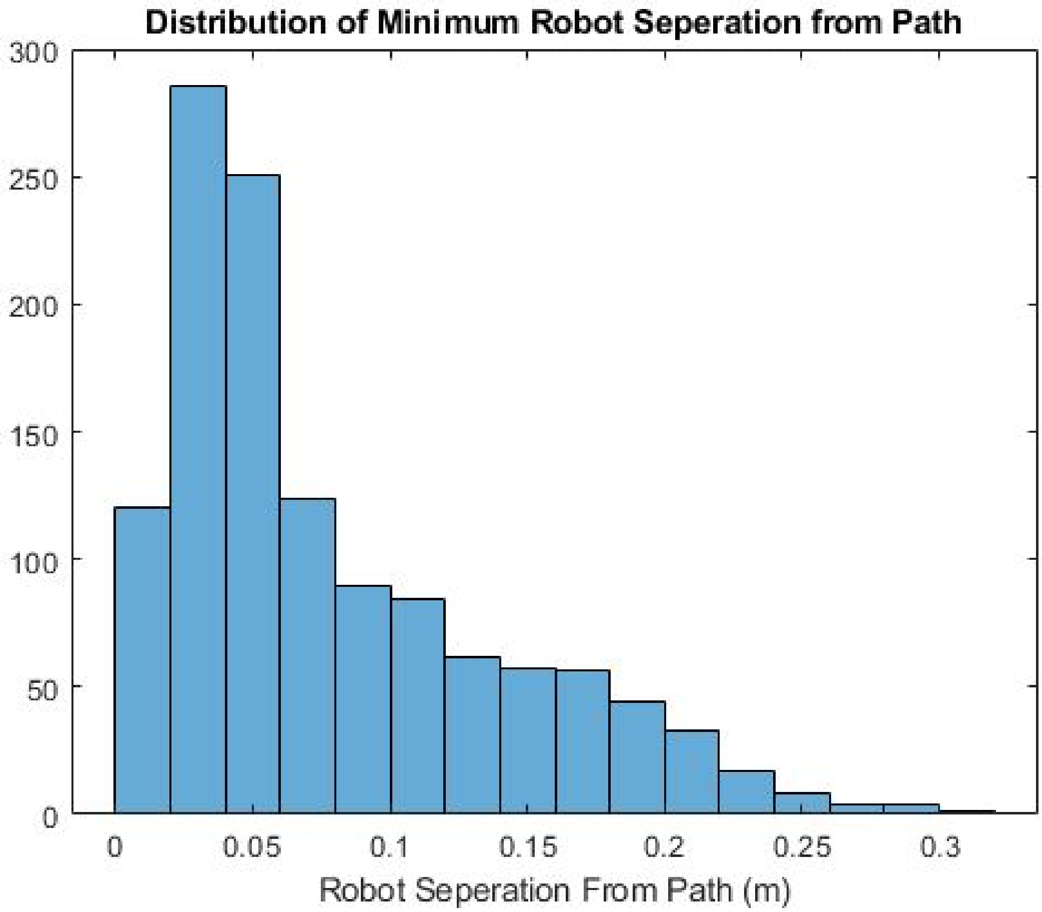
Path Generation:

To generate a path for the robot to follow through the nursery, a set of nodes was mapped to the field. Each row had three nodes mapped to it; one at each end and one at the center. The center nodes ensured that the robot would travel through every row. A node was also mapped at the robots start and end point. A traveling salesman algorithm (“tspof_ga.m”) was used to generate an optimized path between the nodes. The cost of traveling within rows was set to zero. The cost to travel between nodes on the same side of the field was set to the distance of the required maneuver between the nodes. The cost of travel between nodes on different rows and opposing sides of the field was set to an arbitrarily large value. After the algorithm had selected the order of travel between nodes, the main script generated the actual path to be followed through the nursery. Headland maneuvers were restricted to pi and omega turns.

Path Following:

To follow the generated path, a pure pursuit algorithm was implemented. The algorithm relied upon an estimated robot pose supplied by an Extended Kalman Filter. The pursuit algorithm was fed a subset of the path to ensure the robot followed the correct portion of the path. A sample of the robot path and the distribution of path deviations are shown below.



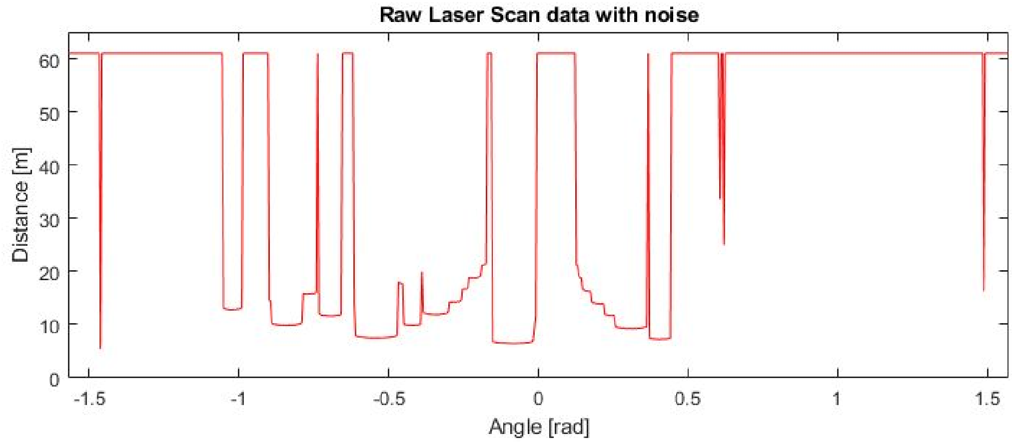


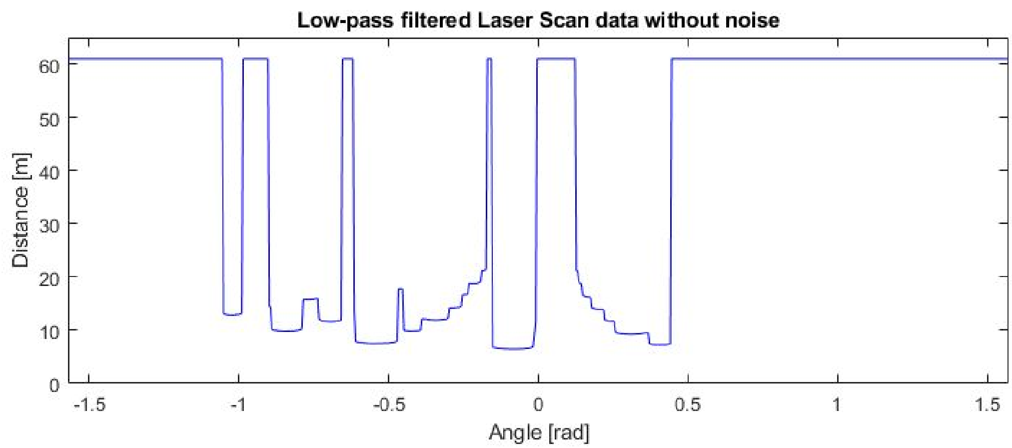
The error consists of mostly small deviations, but the path does show some larger deviations that could potentially lead to impact with trees (especially if the row with were smaller). To prevent this a more frequent gps sampling frequency could be used or alternatively the robot speed could be reduced so odometry errors would have less time to propagate between gps updates.

Image Processing:

In order to properly identify the trees, their respective coordinates, and their trunk diameter a 2-step image processing algorithm was created.

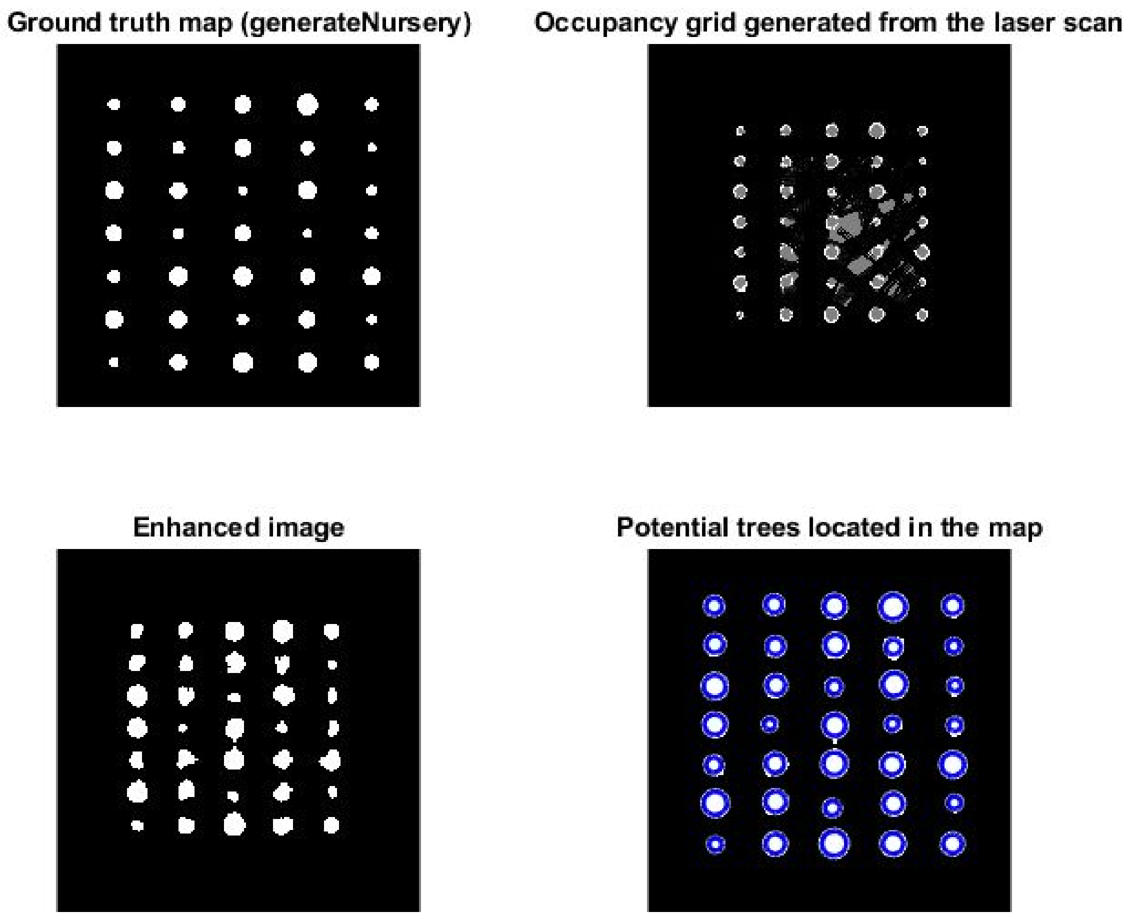
The first step of the algorithm acts on the noisy outputs from the laser scanner. A median filter (low-pass) was used to get rid of the bad measurements. The window size selected to build the low-pass filter was 1x5. It is also important to highlight that all laser scan measurements were taken simultaneously with the gps measurements because those are the most accurate pose tracking states. The figure below gives a better idea of the low-pass filter functioning.





We can see in the picture above that the filter successfully got rid of seven noisy measurements (approximately at angles = -1.48; -0.75; -0.40; 0.45; 0.60; 0.62; 1.49) from the Laser Scan, which are represented by the random spikes occurring in the top image. The filter gives us more reliable data and helps the robot to build a more realistic occupancy grid.

The second step consists on enhancing the occupancy grid generated by the laser scan and using the image processing toolbox from matlab to identify and label all trees. In order to ease the image processing and get rid of irrelevant information the occupancy grid was binarized (i.e., transformed into a B&W image). The threshold chosen to binarize the image was 0.55 (unknown objects have a probability no greater than 0.5). Aiming to enhance and recognize potential edges in the B&W image the technique described by Canny (1986) was used. The final step of image enhancing aims to fill “holes” in the image. It is done by means of the morphological operation of closing an image (erode and then dilate the image). Finally, in order to detect circular shapes and their respective radii the built in Matlab function imfindcircles was used. The figure below illustrates the steps taken.



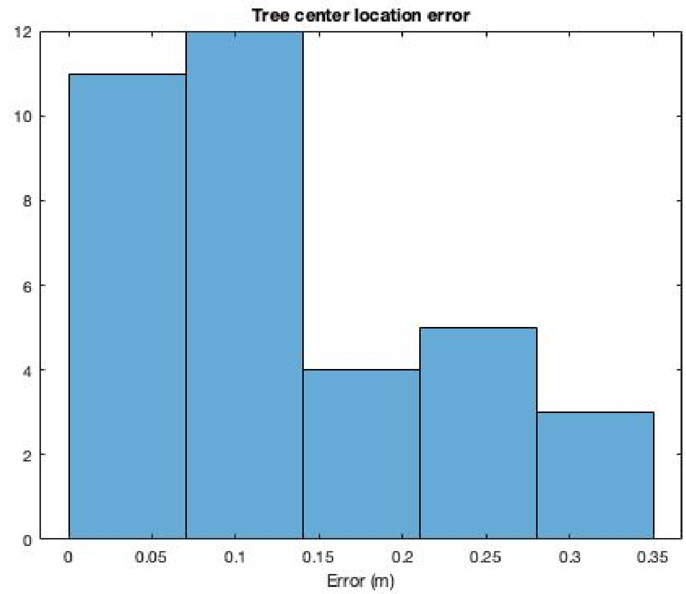
The last but one step consisted on identifying if the found trees indeed belonged to nursery. This step consisted basically on checking whether the trees coordinates matched the expected coordinates of trees belonging to nursery physical area.

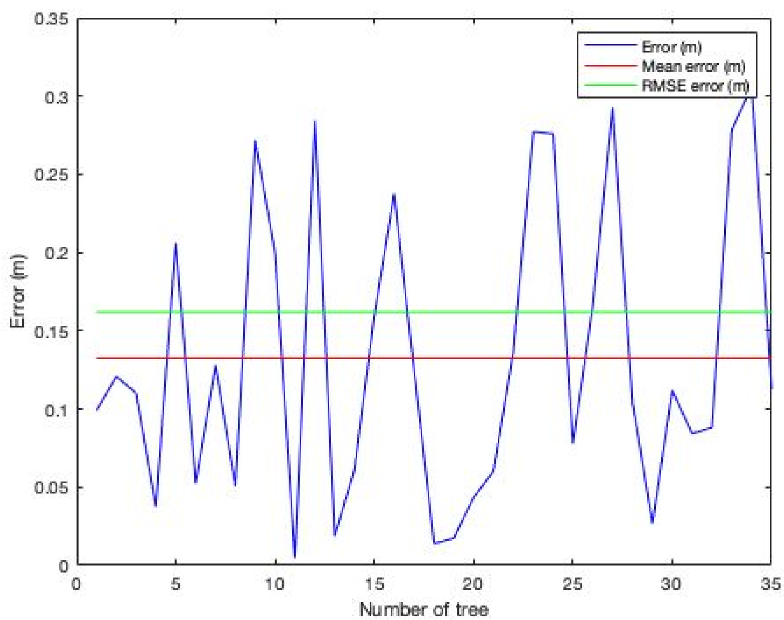
Once the “trees” were all identified they were grouped into clusters representing each of the individual rows and labeled accordingly to the desired output format (Row, X, Y, Diameter, number of the tree in the row). Finally, the algorithm returns a text file containing all relevant information.

Image Processing Performance Analysis

The analysis was divided into two categories, tree location and diameter estimation. The file Image_Processing_Error calculates all statistics and presents them in graphical form and outputs a summary in the Matlab command window.

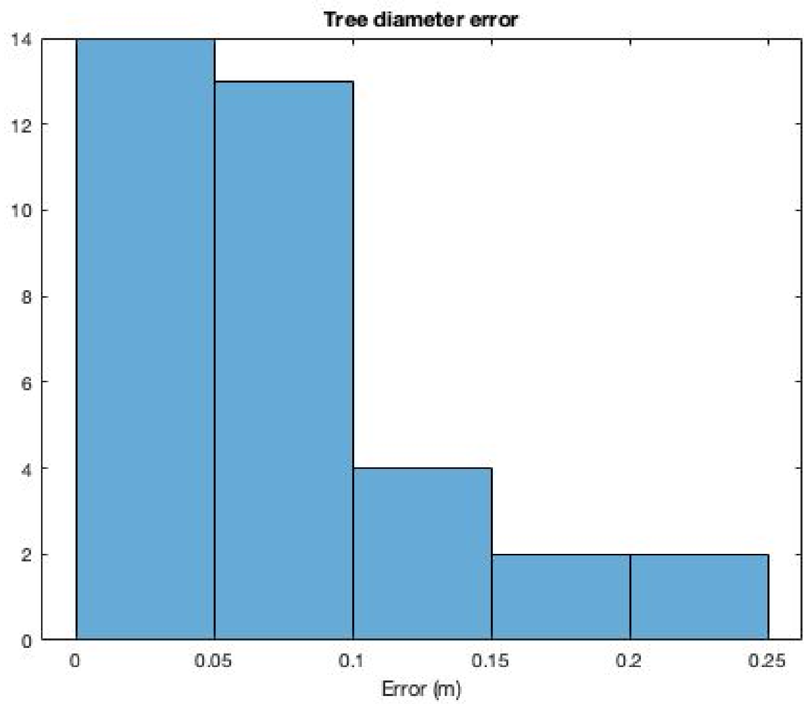
I) Position estimation

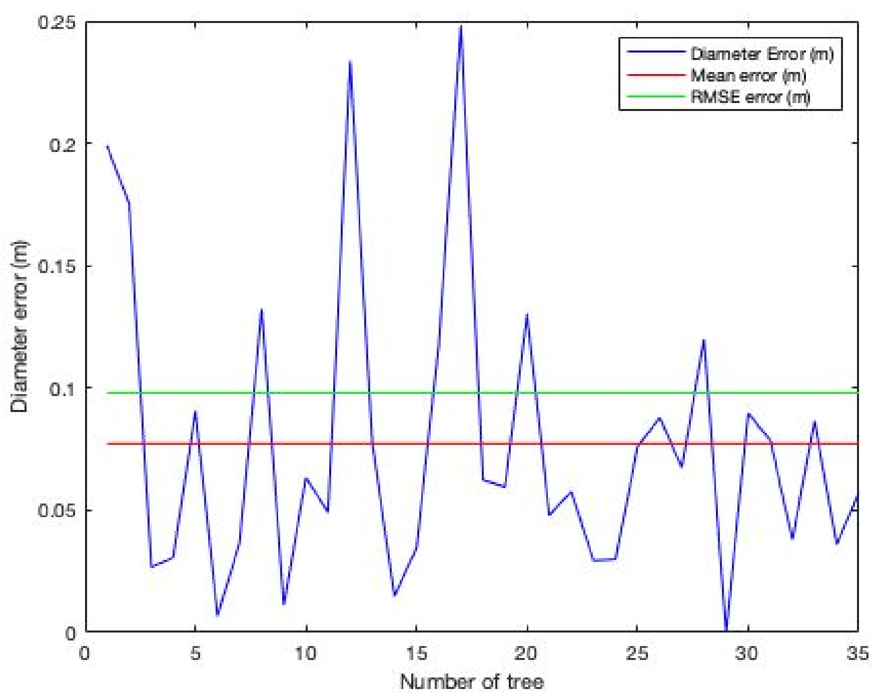




It was observed that in general the position estimation of the trees was relatively good, most estimations presented an error between 0.00 and 0.15 m, with a mean error of 0.13 m. In addition, the maximum error observed was of 0.306 m, and the 95th percentile was of 0.291 m. Considering the row spacing of 3 m and the plant spacing of 2 m, we can conclude that the error values are satisfying.

II) Diameter estimation





It was observed that in general the diameter estimation of the trees was mediocre, since most estimations presented an error smaller than 5 cm, and the mean error of 0.07 m. However, the maximum error observed of 0.248 m is practically unacceptable given that the diameters vary between 0.2 and 0.5 m. Moreover, the 95th percentile error observed was of 0.225 m. More advanced and refined shape recognition techniques could be applied in order to achieve better results regarding estimations of diameter size.

Final Plot

After retrieving information from the image processing algorithm we can visualize both the robot plot and a schematization of the trees in the field.