Introduction

Odometry means measuring wheel rotation using the Encoders.The data from motion sensors is used to estimate change in position over time. It is used in robotics by robots to estimate their position relative to a starting location.
However, Odometry is not always as accurate as one would like.
Some drawbacks in this technique are that odometry computes the position of the robot without considering factors like sensor error, wheel slippage, type of surface, friction of the surface which can lead to inaccuracies in the computed position compared to the actual real-world location of the robot.This project aims to analyze this robot odometry errors and know what factors were causing it.

Data Collection

In data collection, the robot odometry data and the ground truth data for comparison was collected.
The data collection step began developing scripts for the robot to run and gather both these data in the form of logs.
The odometry data of the robot was collected using robot's /odom topic.
The groundtruth information of the robot is calculated by using the laser scans of the robot by subscribing to topic /laserscan and using a wall as a reference point to calculate the robots location.
Various preprocessing steps were applied on this data such as checking for missing value,duplicate information etc
There were some values where the data for same point was recorded more than once so this redundant data was removed

Analysis

This points were first visualised to understand the inherent patterns in them.
One of the observations while visualising them was that the iRobot roomba robots showed variable change in angle of the robot.This value of the change in orientation ranged between -2.44 to +0.6 for the iRobot Roomba.
However for the Mobile Robots, change in angle is constant throughout its motion.It was observed that this value is different at each run and it was in the range of -2.44 to +0.6 while running on the tile flooring and -1.22 to 0.66 on the carpet flooring.
After correlation analysis,iRobot roomba dataset showed a strong negative correlation between the Theta and the Y_error. There is a slight positive correlation between the X_error and Theta i.e the orientation of the robot and there is a positive correlation between the X_error and time of motion of the robot.
In the mobile robot dataset there is a negative correlation between the Velocity X_error, Y_error and Distance_covered_error. There is a positive correlation between the X_error,Y_error, Distance_travelled_error and time of motion of the robot.

After this feature selection was applied on these datasets.
After principal component analysis of the iRobots dataset attributes Velocity, Cumulative_Time, X_error and Y_error had maximum variance for causing the error.

After principal component analysis of the Mobile Robots dataset attributes Theta, Cumulative_Time, X_error and Y_error had maximum variance for causing the error.

Next, these attributes were applied to the KNN classifier to classify the distance travelled error.

The classifier classified that most of the errors start to begin after 3 second when the roomba is in motion.The error in both X-direction and Y-direction start to appear after 2.5 seconds.Most of these errors occur when the velocity is between
-0.2 and 0.2.This is the medium range of the velocity.

Applying the KNN Classifier on the mobile robot data, classified that most of the errors start to begin after 1 second the robot is in motion.They were the most susceptible to errors.The velocities where most errors appear is the positive range of 0 to 0.2 i.e it is moving forward and at the high speeds of forward and backward motion i.e-0.5 and +0.5

Conclusion & Future Work

Now for the conclusion,As the time in motion increases, both the robots were highly susceptible to error.
The roomba had a variable change in angle and the mobile robots had constant change in angle. Errors in the orientation were one of the contributing factors to the error of the robots.

And as the time of motion for these robots increased the errors were rising.

Therefore,implementing a self calibration tool in which the robots learn the orientation from its current data and then calibrate it will help to reduce the odometry errors in real time and the errors will decrease as the time increases.