Mining Robot Odometry

Tejal Shanbhag
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NewYork -14623
ts8583@cs.rit.edu

***Abstract*—This project implements to understand the types of errors inherent in the robots odometry and ways to mitigate these errors. In this project, the robot odometry data is collected. Various data cleaning techniques would be applied on the collected data in order to prepare it for analysis. After the analysis of the collected data, features are selected for error modelling in order to classify the errors in the robot odometry. Based on the analysis of the odometry data and classification, ways in which these errors can be mitigated is reported in order to calibrate the robot and improve its accuracy.**

***Index Terms*—Robot Odometry; Data Analysis; Ground Truth; Caliberate, Self-Learning**

I. Introduction

Robot odometry is a method to calculate the robot's position based on the motion of its wheels. However, there are some drawbacks in this technique as odometry computes the position of the robot without considering factors like sensor error, wheel slip, type of surface, friction of the surface which can lead to inaccuracies in the computed position compared to the actual real-world location.

Mining helps to understand patterns,anomalies, and trends in the dataset which helps to extract meaningful information about the dataset. Mining the robot odometry data will help to determine the kinds of errors in the data. Determining these errors will help to dynamically calibrate the robot which will improve its accuracy.

One of the most famous robots to exist is the Curiosity Rover which was sent to Mars by NASA to explore the Gale Carter. Since this rover is encountering surfaces and terrains which are completely unknown to humans. Here, there needs to be a constant dynamic calibration for the rover. As without accurate odometry calibration any discovery made by the rover on Mars will differ from the actual position.

A. Structure of the Report

The report begins with a detailed literature review of the types of odometry error in section II. Section III provides the details on how the data was collected from the robots and the description of all attributes of the collected data used for this project. Section IV presents the data cleaning and pre-processing methods applied on the sensor data which is collected from the robots. Section V-A discusses the various observations obtained by visualizing the data. Section V-B presents the error modelling done on the data to know what factors cause the errors in the robot odometry. Section VI concludes the project and identifies direction for future work.

B. Hypothesis

In this project, the robot odometry data would be first collected. The collected data will hold the information about the robot position, orientation, velocity and surface type. Various data cleaning techniques would be applied on the data in order to prepare it for analysis. First the data would be visualised to know about the patterns in them and understand the types of errors inherent in our robot's odometry.Then various models of regression will be applied to know about how the errors are in the data. After this, the features will be selected using Principal Component Analysis and applied to the K Nearest Neighbour algorithm to classify the errors in Odometry.The project analyses this data and provides details on what factors are contributing to the error and how their calibration will improve its accuracy.

II. Literature Review

Robot odometry is a method to calculate the robot's position based on the motion of its wheels.The wheel revolutions are counted by the encoders on the drive motor of the robot. The position of the robot after some movement relative to the starting point can be calculated using simple geometric equations.Odometry only considers displacement of the robot by the wheel revolutions and relative to the surface the robot moving on.However different errors also need to be considered while calculating the odometry.[1]

One usecase is a situation when there is a wheel slippage. If any one of the wheels of the robot were to slip due to oil spill, the encoder still records the wheel revolution and the ground truth position of the robot will not match with the estimated position. Apart from this case, many other situations where inaccurate position would be recorded as error is not taken into account. Therefore its important to compare the values of position and log the error for them accordingly.[2]

These errors in odometry can be broadly divided into two categories[1][2]:

1) **Systematic Errors**
   Systematic errors are those errors which are independent of the environment in which the robot is moving.[2] has categorized the different systematic errors as follows:
   a) **Unequal wheel diameters**

One of the contributing parameters for the robot odometry is the number of revolutions of the robot wheel. Therefore, having unequal diameters for the drive wheels of the robot will affect the speed and displacement of the robot.Also most robot have rubber wheels. Due to asymmetric load distribution these wheels compress differently.[1] This error can be represented as

$$E_d = D_R/D_L$$

where D_R and D_L is actual wheel diameters of right and left respectively and E_d is the error.[1][2]

b) **Average of both wheel diameters differs from nominal diameter**
If the average of both wheel diameters differs from the nominal diameter, then the encoder will incorrectly convert the number of revolutions of the wheel to linear distance,which will give incorrect displacement of the robot from the start position.[1][2]

c) **Misalignment of wheels**
The misalignment of the wheels will result in the robot moving to one side without the robot knowing it.Therefore going in a straight path with such an error will be difficult.[1][2]

d) **Uncertainty about the effective wheelbase (due to non-point wheel contact with the floor)**
Wheelbase is the distance between the contact point of drive wheels and the shaft in between the wheels.Uncertainty in the effective wheelbase error is caused when the robot wheels contact the floor not in one point, but rather in a contact area and the encoder incorrectly converting this to number of revolutions.[1][2]

e) **Limited encoder resolution**
The limited encoder resolution is the number of samples the encoder records during wheel revolution. [2]A good estimate of exact distance covered by the robot can be achieved by taking more number of samples per wheel revolution.[1][2]

f) **Limited encoder sampling rate**
The limited encoder sampling rate refers to how often the encoder takes the a sample.[2] If this sampling rate is slower than what the resolution requires, the encoder will miss some values and it will be translated to error.[1][2]

2) **Non-Systematic Errors**
Non-systematic errors are those errors which are dependent on the environment in which the robot is running.[2] These errors are caused due to the interaction between the robot and the unpredictable environment it is running in. Non-systematic errors are relatively difficult to predict, until the robot runs in that surrounding.[1]

a) **Travel over uneven floors** Irregularities over the surface such as cracks, bumps where the robot moves will affect the motion of the wheels. The robot moves over the terrain which it does not have knowledge about.[1][2]

b) **Travel over unexpected objects on the floor** Whenever a robot comes across an unexpected object, in order to move over it, the robot wheels turn more to go over the object than to go the same distance over an unobstructed floor.Therefore in such cases odometry calculates the distance more as it is based on the wheel revolutions.[1][2]

c) **Wheel-slippage due to slippery floors**
Another cause of error is wheel slippage. This is caused due to multiple scenarios such as slippery floors, over-acceleration, skidding in fast turns, etc.[1][2]This leads to incorrect calculation of distance as odometry depends on number of wheel revolutions and the displacement will be incorrectly recorded.

III. Data Collection

The data collection step began with first developing scripts for the robot to run and gather the data from the log of the robots. Two different kinds of robots were used to collect this data. Figure 1 shows the iRobot Roomba Robot and Figure 2 shows the Mobile Robots.

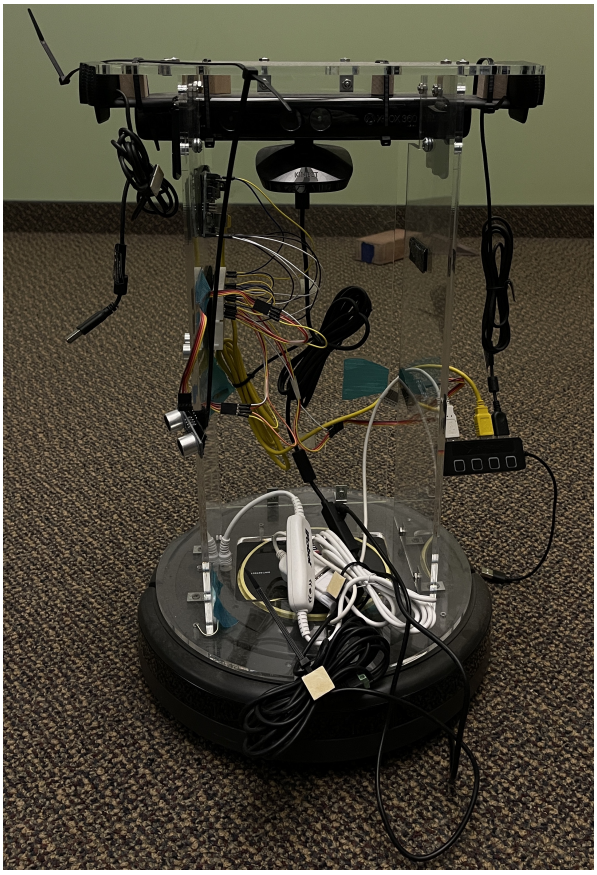

Fig. 1. iRobot Roomba Robots

Fig. 2. Mobile Robots

All the logs were collected by running the robot in the Robotics lab and the GCCIS 3rd-floor hallway. A total of 17392 raw samples were collected in the form of 62 log files from the robots labelled as Mobile Robot Lewis, Mobile Robot Clark, iRobot Roomba E and iRobot Roomba F. A total of 82 different runs with average of 24 data points per second was conducted in order to collect the data.

The odometry information of the robot taken by subscribing to the robot's /odom topic.The groundtruth information i.e the actual position of the robot at a given point is calculated by using the laser scans of the robot.The laser scans of the robot give the distance between the robot and a reference wall from -28 degrees to 28 degrees.These values are obtained by subscribing to the topic /laserscan.This data was collected by keeping the wall as a reference point and getting the coordinates of where the wall is located initially, and then calculating where the robot is currently located using the laser scans and comparing with the actual reading of the odometry reported by the robot sensor.

Following information was logged while running the robot.

- **Timestamp:**Time when the log was recorded
- **X_ODOM:**: Odometer reading of the x-co-ordinate position
- **Y_ODOM:** Odometer reading of the y-co-ordinate position
- **Velocity:** Velocity of the robot
- **Change in angle(Θ):** Angle of the robot at a given time
- **Odometry_Distance_Travelled:** Odometer reading of the distance covered by the robot using the X_ODOM and Y_ODOM and the start position of the robot.
- **X_GR:** Ground truth x-co-ordinate position or the actual x-co-ordinate position of the robot calculated using laser scans.
- **Y_GR:** Ground truth y-co-ordinate position or the actual y-co-ordinate position of the robot calculated using laser scans.
- **Ground_Truth_Distance_Travelled:** Ground truth distance covered by the robot using the X_GR and Y_GR and the start ground truth position of the robot.
- **Surface_Type:** The type of surface the robot is running on.

Using the variables directly collected using the sensors of the robot some other variables were also generated. Following are those variables:

- **X_error:**Difference in odometer recorded x-co-ordinate value and actual ground truth x-co-ordinate value for the robot.
- **Y_error:**Difference in odometer recorded y-co-ordinate value and actual ground truth y-co-ordinate value for the robot.
- **Distance_travelled_error** Difference in odometer recorded distance travelled and ground truth distance travelled value for the robot.

IV. Data Pre-Processing

Data preprocessing is one of the essential steps of data mining. This stage helps to transform the raw information into an usable information.Initially the data collected by running the robots were in log formats, in the first step the logs were converted into CSV format.

An additional column to separate each run of the robot was added. This was added in as an index value in order to separate all the runs of the robot. This column is named as SeriesID.

As this data was all recorded from the sensors of the robot,the had some missing values for certain fields.These missing values were mainly were when the reference wall was beyond a reachable point from the robot laser,therefore calculating the actual location of the robot was difficult as the laser returned values as 'inf'.

V. Analysis

A. Data Exploration

In order to understand the behaviour of the robots and understand the pattern of the data ,we first visualised the data points. Figure 3 shows the plot of change in the angle of the robot in the iRobot Roomba dataset for one run of the robot.It can be observed that there is a variable change in orientation of the data through one particular run. It appears that most change in the angle is highest at the start of the run.

Figure 4 shows the plot of change in the angle of the robot in the iRobot Roomba dataset for multiple velocity profiles.For all the runs it can be observed that the highest deflection in the

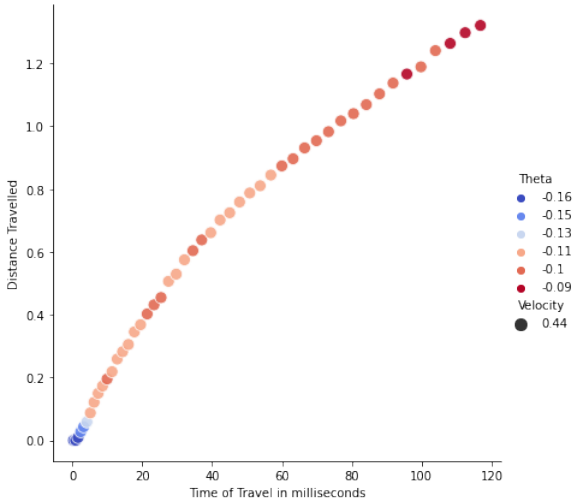

Fig. 3. Variable change in orientation of iRobot Roomba for one single run

angle is at the beginning of each run.This behaviour is visible in all velocity profile.

Each and every run of the motion of the iRobot Roomba robot in the the forward direction had variable change in the angle.However this behaviour was different in the backward direction.In the backward direction, there was variable change in the angle of the robot in some runs only.

Figure 5 shows how for one velocity profile for different runs there is variable change in angle. Figure 6 shows variable change of orientation of the robot had range of -0.02 to +0.02 for velocities between 0 to -0.25. This range increased to -0.2 to +0.2 for velocities lesser -0.25.

Figure 7 shows the change in the angle of the robot as a constant value for the mobile robots.This behaviour of constant change was shown for all the velocity profiles of the mobile robot.But the robot behaves differently and different runs.The value of the change in orientation is ranges between -2.44 to +0.6 when run on the tile flooring and -1.22 to 0.66. on the

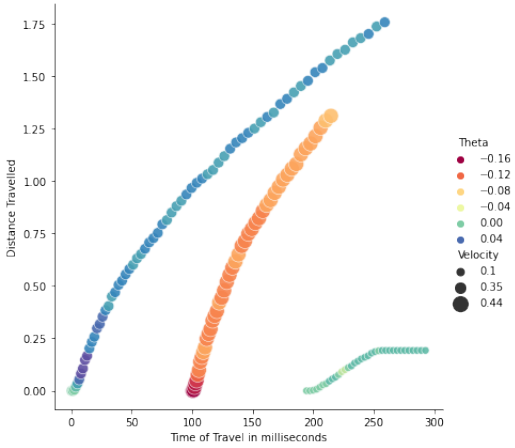

Fig. 4. Variable change in orientation of iRobot Roomba for one single run for multiple velocity profiles

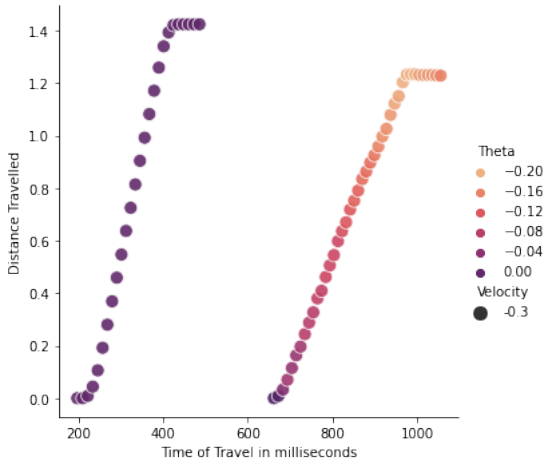

Fig. 5. Variable change in orientation of iRobot Roomba for different runs of same velocity profiles

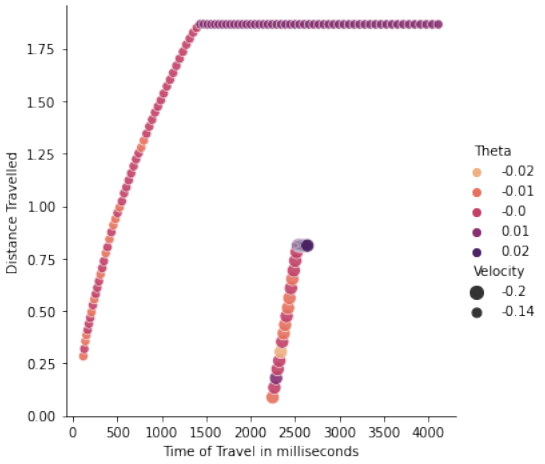

Fig. 6. Constant range of change in orientation of iRobot Roomba across multiple velocity profiles

Carpet flooring.

Robot Type	Flooring	Range of Orientation	Velocity Range
Mobile Robot	Tile	[-2.44,0.6]	[-0.4,0.4]
Mobile Robot	Carpet	[-1.22,0.66]	[-0.4,0.4]

TABLE I
VELOCITY-ORIENTATION RANGES FOR MOBILE ROBOTS

B. Error Modelling

1) ***Correlation Analysis****:* After visualising the data for both the robots, correlation values were checked for each of them. Figure 3 is the correlation heat map of the dataset created for the iRobot Roomba E and F. Figure 4 is the correlation heat map of the dataset created for the Mobile Robots Clark and Lewis.

Figure 8 shows that there is a strong negative correlation between the Theta and the Y_error.Since the Cumulative Theta is aggregation of the theta values that too has a strong negative correlation between the Cumulative Theta and Y_error.There

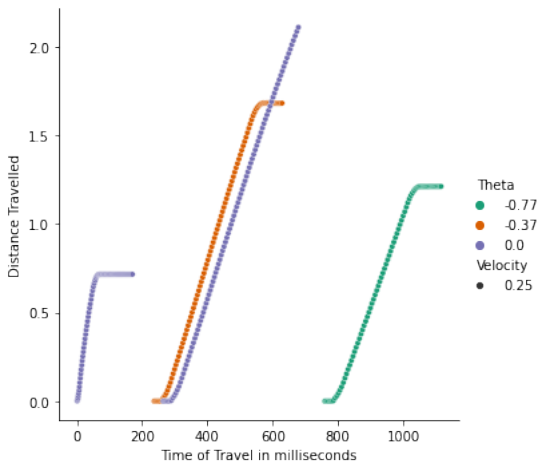

Fig. 7. Constant change in orientation throught the runs of Mobile Robot

is a slight positive correlation between the X_error and Theta i.e the orientation of the robot.Also there is a positive correlation between the X_error and time of motion of the robot.

Figure 9 shows that, there is a negative correlation between the Velocity and Error in the X_error, Y_error and Distance_covered_error. There is a slight positive correlation between the X_error and Theta i.e the orientation of the robot. Also there is a positive correlation between the X_error,Y_error and Distance_travelled_error and time of motion of the robot.

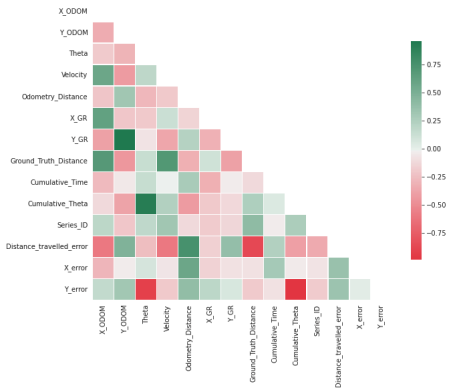

Fig. 8. Correlation Heatmap for iRobot Roomba Dataset

*2) **Regression Analysis**:* Regression helps to understand different patterns in the data as it defines the connection between the target variables and independent variables.I was used to understand if the attributes taken indiviadually are dependent on th target variable. For the different regression applied below on the mobile robot data and the iRobot Roomba data the target variable was considered as the Distance_travelled_error. [3][4]

1) **Linear Regression**

Linear regression analysis of the irobot roomba dataset with the features Velocity,Cumulative Time,Cumulative Theta,X_error and Y_error gave the features Y_error and X_error as highly contributing factor to the distance traveled error of the robot.Velocity also had negative value similar to that of correlation analysis.

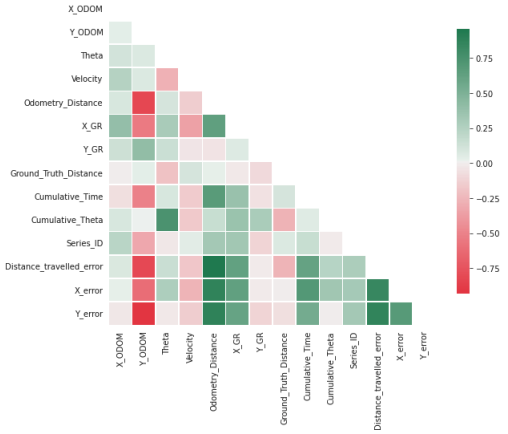

Fig. 9. Correlation Heatmap for Mobile Robot Dataset

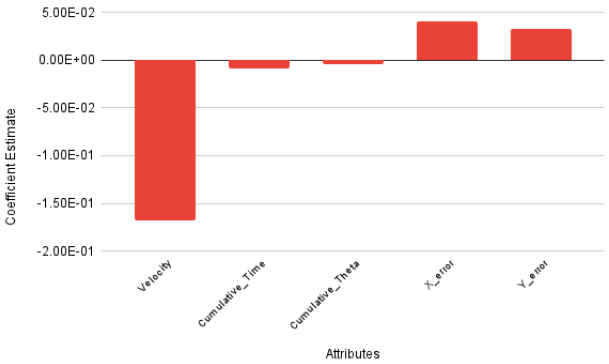

Fig. 10. Coefficient Estimate for irobot roomba dataset for linear regression

Linear regression analysis of the mobile robot dataset with the features Velocity ,Cumulative Time, Cumulative Theta, X_error and Y_error gave the features Velocity and X_error as highly contributing factor to the distance traveled error of the robot.Y_error had negative impact which was contradicting to that of correlation analysis.

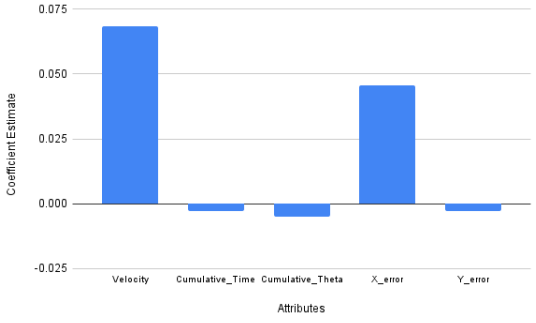

Fig. 11. Coefficient Estimate for mobile robot dataset for linear regression

2) **Lasso Regression:**
Lasso regression analysis of the irobot roomba dataset with the features Velocity, Cumulative Time, Cumulative Theta, X_error and Y_error gave the features Cumulative_Time as a contributing factor and Cumulative_Theta as highly non contributing factor to the distance traveled error of the robot. Lasso regression analysis of the mobile robot dataset with the features Velocity ,Cumulative Time, Cumulative Theta, X_error and Y_error gave the features Cumulative Time, as highly contributing factor to the distance traveled error of the robot.Cumulative Theta had negative impact which was contradicting to that of correlation analysis.

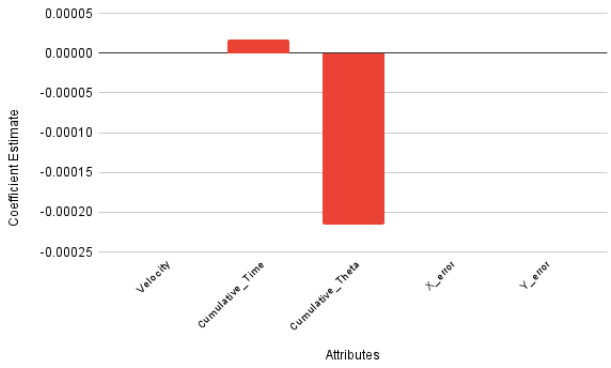

Fig. 12. Coefficient Estimate for irobot roomba dataset for lasso regression

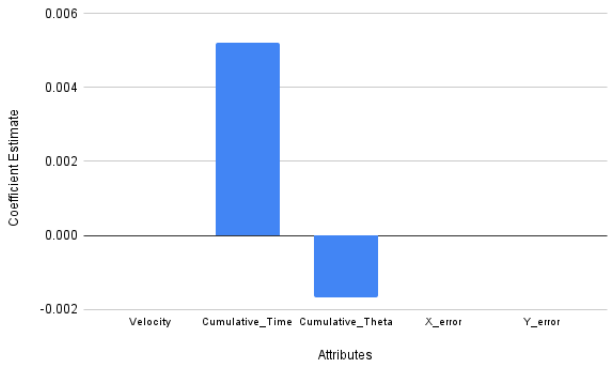

Fig. 13. Coefficient Estimate for mobile robot dataset for lasso regression

Performing this on both datasets gave the Cumulative Time as a highly contributing factor.This helped in eliminating other paramters as the coefficient score value is zero for them.

3) **Ridge Regression:**
Ridge regression analysis of the irobot roomba dataset with the features Velocity,Cumulative Time,Cumulative Theta,X_error and Y_error gave the features X_error and Y_error as a contributing factor to the Distance Error and Velocity to be the non contributing factor for the same.

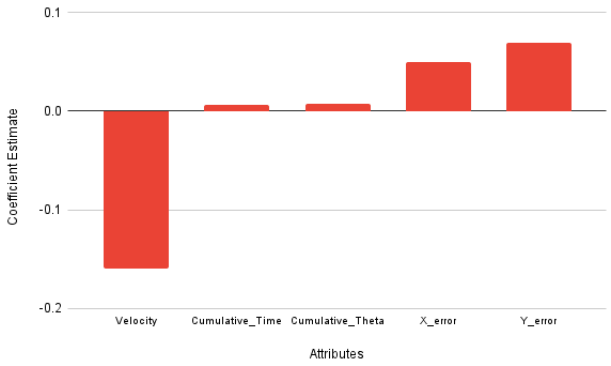

Fig. 14. Coefficient Estimate for irobot roomba dataset for ridge regression

Ridge regression analysis of the mobile robot dataset with the features Velocity ,Cumulative Time, Cumulative Theta, X_error and Y_error gave the features Velocity, and X_error as highly contributing factor to the distance traveled error of the robot.Y_error had negative impact which was contradicting to that of correlation analysis.

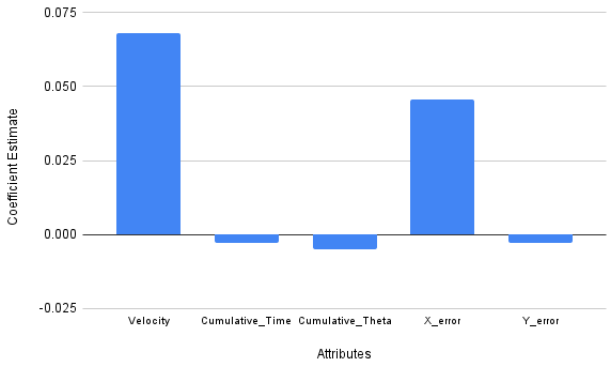

Fig. 15. Coefficient Estimate for mobile robot dataset for ridge regression

4) **Elastic Net**
Elastic Net analysis of the irobot roomba dataset with the features Velocity,Cumulative Time,Cumulative Theta, X_error and Y_error gave the features Cumulative Time as a contributing factor to the Distance Error and Cumulative Theta to be the highly non contributing factor for the same.
Elastic Net analysis of the mobile robot dataset with the features Velocity ,Cumulative Time, Cumulative Theta, X_error and Y_error gave the features Cumulative Time as highly contributing factor to the distance traveled error of the robot.Cumulative Theta had negative impact which was similar to that of correlation analysis.

*3) **Feature Selection**:* After principal component analysis of the Mobile Robots dataset the first eigenvector had the maximum variance in the direction of Y_error. The second eigenvector had the maximum variance in the direction of X_error and Cumulative time. The third eigenvector had the

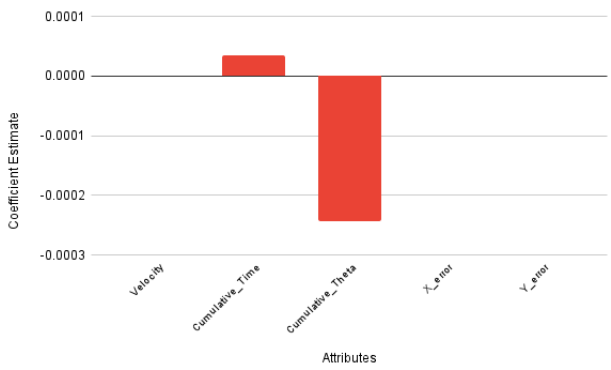

Fig. 16. Coefficient Estimate for irobot roomba dataset for elastic net regression

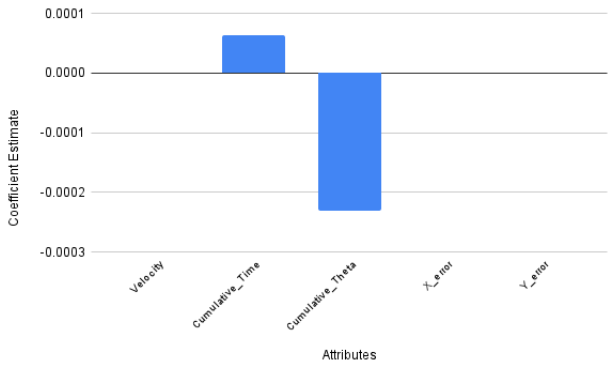

Fig. 17. Coefficient Estimate for mobile robot dataset for elastic net regression

maximum variance in the direction of Velocity of the robot.The first three eigenvectors held more than 95% of the variance for the error therefore, for further KNN modeling , the features Y_error, X_error, Cumulative_time and Velocity are selected.The results for the principal component analysis was matching with what Correlation values were saying about the dataset.

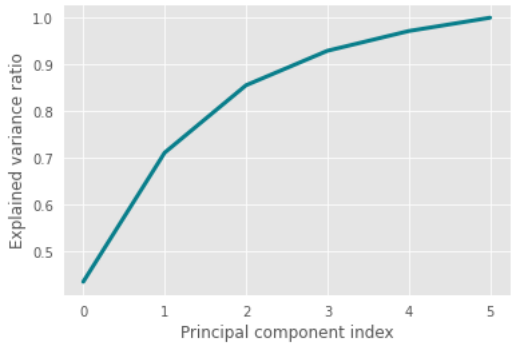

Fig. 18. Cumulative explained variance by number of principal components for Mobile Robot Dataset

After principal component analysis of the iRobots dataset, the first eigenvector had the maximum variance in the direction

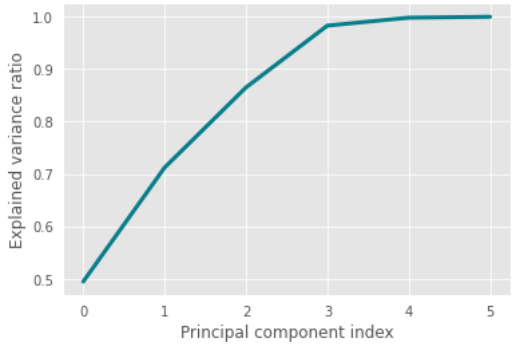

Fig. 19. Cumulative explained variance by number of principal components for iRobot Roomba Dataset

of X_error. The second eigenvector had the maximum variance in the direction of Y_error and Cumulative time. The third eigenvector had the maximum variance in the direction of Cumulative Time of run of the robot.The fourth eigenvector had the maximum variance in the direction of orientation(Theta) of the robot.The fifth eigen vector had the maximum variance in the direction of X_error of the robot.

The first 5 eigenvectors held more than 95% of the variance for the error therefore, for further KNN modeling , the features Y_error, X_error, Cumulative_time and Theta were selected.

Fig. 20. Attributes selected based on PCA

*4) **K-Nearest Neighbours**:* Using the features selected for the dataset, they were supplied to the KNN classifier. The classes were selected if there exist error or not if the Distance travelled error was grater than 0.001.Applying the KNN Classifier on the irobot roomba dataset, classified that most of the errors start to begin after 2.7 seconds the robot is in motion.The error in both X-direction and Y-direction start to appear after 2.5 seconds as seen from Figure 17 and Figure 18.

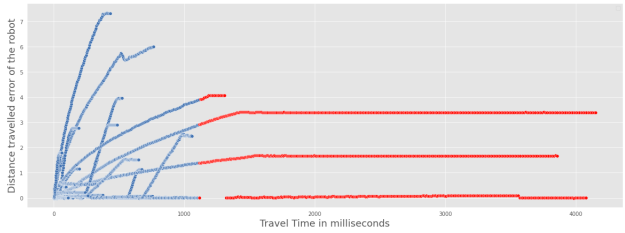

Fig. 21. KNN classifier classified error in distance travelled for irobot roomba dataset

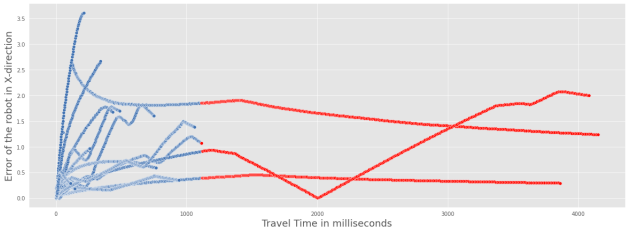

Fig. 22. KNN classifier classified error in X direction for irobot roomba dataset

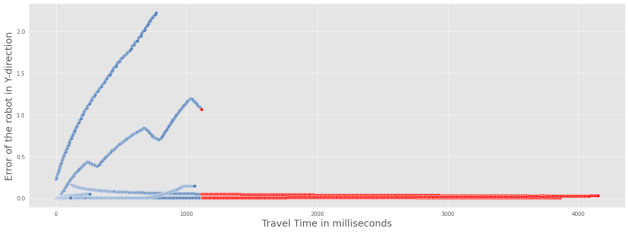

Fig. 23. KNN classifier classified error in Y direction for irobot roomba dataset

From Figure 24, it can be observed that the classifier classifies most of the errors occur when the velocity is between -0.2 and 0.2 which is midway of the speed range of -0.5 to 0.5.These robots seem not to be much affected by high velocity profiles such as high speed of 0.5 both in forward and backward direction.

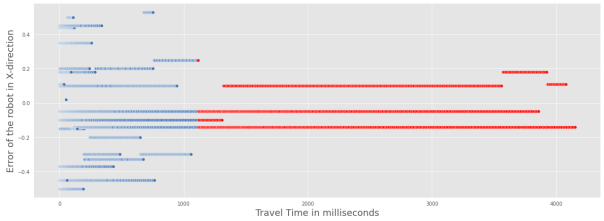

Fig. 24. KNN classifier classified error for different velocities for irobot roomba dataset

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.This robot is more susceptible to error as within a second of movement there are errors in the odometry of the robot.

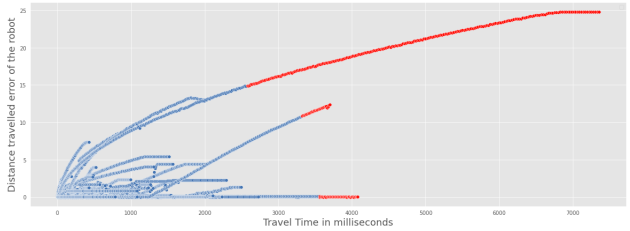

Fig. 25. KNN classifier classified error in distance travelled for mobile robot dataset

However, the errors in X and Y directions individually appear after 2.5 seconds.

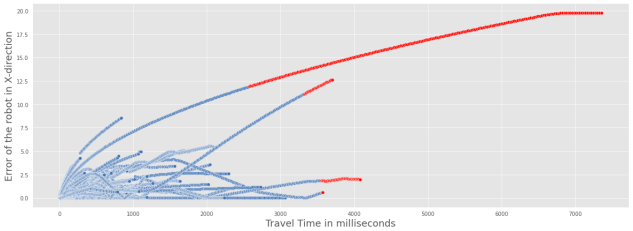

Fig. 26. KNN classifier classified error in X-direction for mobile robot dataset

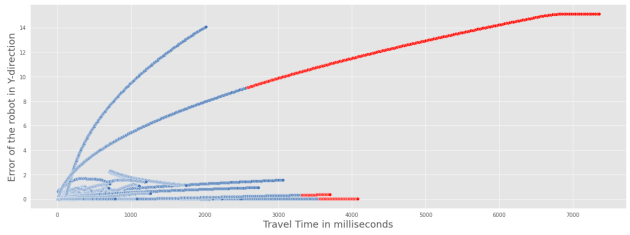

Fig. 27. KNN classifier classified error in Y-direction for mobile robot dataset

From Figure 27 , it can be observed that the classifier classifies most of the errors that happen when the velocity is in 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.

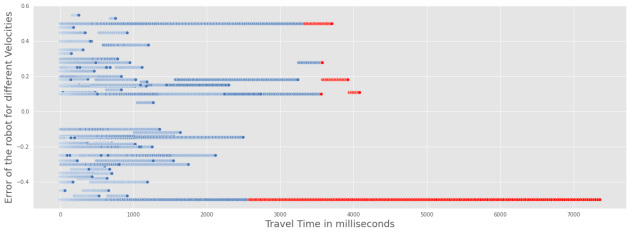

Fig. 28. KNN classifier classified error for different velocities for mobile robot dataset

C. Result Summary

From all the above analysis of both the the robots, change in the orientation of the robot was one of the contributing factors for the error in the robots odometry. Visualizing these orientation values showed how the values were variable when the roomba was in motion and a constant value when the mobile robots were in motion. Error in the X direction was highly correlated with the time of motion for both the robots. The iRobots roomba had high error tendency when the velocities were in medium range and not very fast. This was something unusual but the correlation of the theta and the error were validated due to that. The mobile robots were susceptible to error on high velocity profiles.

After regression analysis of the datasets Cumulative time and Theta were the contributing factor for Mobile robots error and Velocity and Cumulative time as contributing factor for the error in the iRobots roomba robots. Of all the regressions, Lasso regress gave the correct analysis as it gave the highest accuracy for both the datasets.

Further on feature selection, the attributes Velocity,Time, X error and Y error were selected for the iRobot Roomba and Change in orientation,Time, X error and Y error for mobile robots were selected.

After applying these attributes as parameters for the KNN Classifier, velocity range of -0.2 to +0.2 were more susceptible to error in the roomba. Velocities was one of the factors affecting the errors in roomba dataset, and the most errors were classifies in the max range window of velocities i.e 0.2 to -0.2. For the the mobile robots constant angle change was a contributing factor in the robots error and this robot was causing it to move sideways.

Therefore based on the analysis above, using the constant value change in angle for the mobile robots and average of the variable change in angle of the robots in realtime would help to calliberate it can reduce the error overtime.

VI. Conclusion

From the Data Visualisation and the Analysis it can be concluded that high velocities of motion aren't a reason for high errors in the robot odometry. Infact, for the iRobots roomba robot the slowest speeds caused the highest amount of errors in the robot. The change of Theta i.e variable change of theta throughout the motion of the roomba was one of the factors contributing to the error, and if this could be calibrated , then the errors for the roombas could be significantly reduced. For the mobile robots, most of the errors occurred when the it was moving at highest speeds in backward motion and for mid to high speeds in the forward motion. These robots were more susceptible to error due to velocity.

VII. Future Work

Each of the robots showed different behaviours when they were in motion. Using this existing data, we could implement a self learning calibration method which will help to reduce the error of odometry. For the iRobot Roombas using the change in angles for the model to learn and calibrate it can help to reduce the errors drastically as Theta was one of the features contributing to the error in these robots. For the mobile robots, using the constant change in theta for calibration could help to reduce them. Implementing a approach where the robot learns calibration estimates when its moving automatically in real time will help to mitigate the odometry errors.

Acknowledgment

I would like to express my gratitude to Prof. Zachary Butler for being an excellent mentor and guiding me throughout my capstone. There were some instances where I was stuck for some period when Prof. Butler gave me some directions which helped me solve those roadblocks easily. I would also like to thank Prof. Leon Reznik for helping with the milestone presentations and the valuable feedback for the poster.

References

[1] J. Borenstein and L. Feng, "Gyrodometry: a new method for combining data from gyros and odometry in mobile robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, 1996, pp. 423–428 vol.1.

[2] ——, "Umbmark: a benchmark test for measuring odometry errors in mobile robots," in *Other Conferences*, 1995.

[3] S. Weisberg, *Regression Analysis*. John Wiley Sons, Ltd, 2007. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470050118.ecse349

[4] D. Kumar. A complete understanding of lasso regression. [Online]. Available: https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/