Mining Robot Odometry

Tejal Shanbhag (ts8583@rit.edu)
Advisor: Dr. Zachary Butler
Computer Science Department-Rochester Institute of Technology, NY

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 slippage, type of surface, friction of the surface which can lead to inaccuracies in the computed position compared to the actual real-world location.

In this project, the robot odometry data would be collected. The collected data will hold the information about the robot position, orientation, linear velocity and angular velocity.

Various data cleaning techniques would be applied on the time series data in order to prepare it for analysis and understand the types of errors inherent in the robot's odometry.

Dataset Details

1. **Data Collection:** Scripts were written for the robot to run and gather the data from the log of the robots. The odometry information of the robot taken by subscribing to the robot's **/odom** topic.The ground truth information of the robot is calculated by using the laser scans of the robot by subscribing to **/laserscan** topic.The ground truth data collected by keeping the wall as a reference point, the start position of the robot using the trigonometry and distance formula.
2. **Data Cleaning:**
   Missing information/Null Values: There were some values where the laser was not able to record the actual position and had returned null values.These values were handled by removing them.
   Duplicate information: There were some values where the data for same point was recorded more than once so this redundant data was removed

Data Analysis

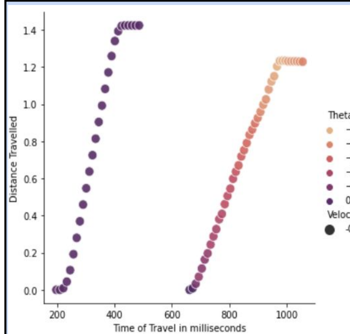

Fig 1: Variable change in orientation of iRobot Roomba for backward motion

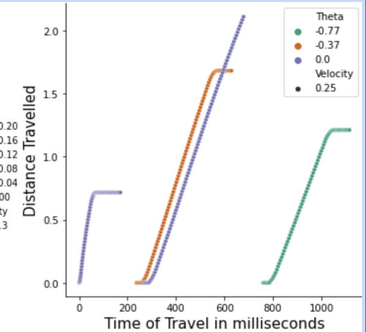

Fig 2:Constant Change in orientation of Mobile Robot.

Change in Angle of the Robot

The iRobot roomba robots showed variable change in orientation when they were in motion.The value of the change in orientation is ranges between -2.44 to+0.6.However for the Mobile Robots, change in orientation when in motion is constant.This value is different at each run and theranges between -2.44 to+0.6 when run on the tile flooring and -1.22 to 0.66 on carpet flooring.

Attribute Correlation

In iRobot roomba dataset there is 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 the 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.

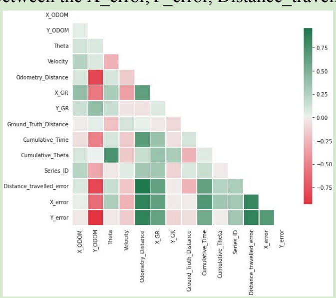

Fig 3:Correlation Heatmap of attributes for iRobot roomba

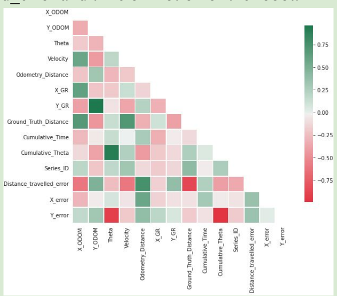

Fig 4:Correlation Heatmap of attributes for Mobile Robots

Modelling

Feature Selection

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.

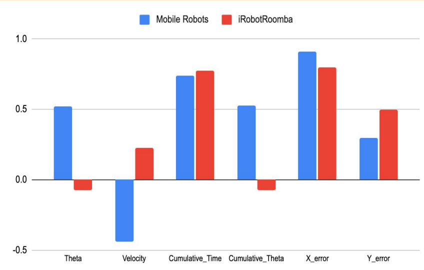

Fig:5 PCA feature importances for both datasets

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

K- Nearest Neighbours

Applying the KNN Classifier on the irobot roomba dataset, classified that most of the errors start to begin after 3 second the robot is in motion.

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.

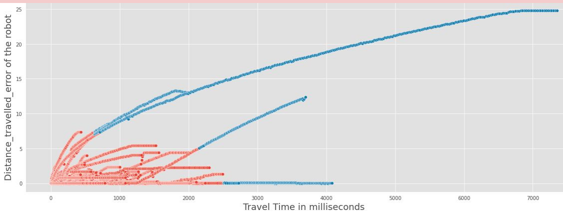

Fig 6:KNN classified errors on iRobot's Roomba Dataset after 3 second

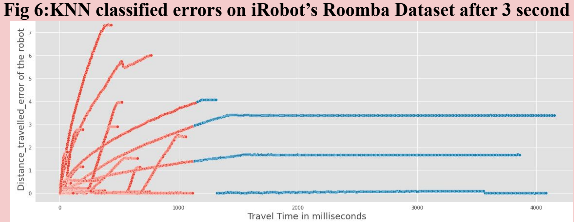

Fig 7:KNN classified errors on Mobile Robots Dataset after 1 second

Conclusion & Future Work

As the time in motion increases, both the robots were highly susceptible to error and specifically the mobile robots which were prone to error after 1 second of movement.

The robot had a variable change in angle in forward motion and constant change in angle for backward motion. Errors in the orientation of robots for forward motion was much greater than that in backward motion.

Implement a self calibration tool in which the robot corrects the odometry errors in real time while moving.

R·I·T | B. THOMAS GOLISANO College of COMPUTING AND INFORMATION SCIENCES

REFERENCES

[1] J. Borenstein and L. Feng. Umbmark: a benchmark test for measuring odometry errors in mobile robots. In Other Conferences, 1995.
[2] 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