

Indoor Robot Odometry Modelling and Calibration Incorporating Floor Surfaces and Acceleration

by

Yanming Pei



Thesis

Submitted by Yanming Pei

for fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Primary Supervisor: Associate Professor Lindsay Kleeman

**Department of Electrical and Computer Systems Engineering
Monash University, Australia**

May, 2018

© Copyright
by
Yanming Pei
2018

Except as provided in the Copyright Act 1968, this thesis may not be reproduced in any form without the written permission of the author. I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

Dedicated to my beloved wife, Shuang ZHANG.

Contents

| | |
|--|-------------|
| List of Tables | viii |
| List of Figures | ix |
| Abstract | xix |
| Acknowledgements | xxii |
| 1 Introduction | 1 |
| 1.1 Contributions | 4 |
| 1.2 Publications | 7 |
| 1.3 Structure of the Thesis | 8 |
| 2 Indoor Floor Classification with Common Sensors | 9 |
| 2.1 Introduction | 9 |
| 2.2 Robot Sensing Hardware System | 12 |
| 2.2.1 Robot Platform | 12 |
| 2.2.2 Colour Sensing | 12 |
| 2.2.3 Motor Current Sensing | 13 |
| 2.3 Robot Operating System (ROS) | 16 |
| 2.4 Floor Classification with Colour Sensing | 18 |
| 2.4.1 Support Vector Machine | 18 |
| 2.4.2 Experiments and Results | 19 |
| 2.5 Floor Classification with Motor Current | 21 |
| 2.5.1 Motivating Experiments | 21 |
| 2.5.2 Floor Inclination Angle Compensation | 24 |
| 2.5.3 Motor Current Compensation | 27 |
| 2.5.4 Experiments and Results | 31 |

| | | |
|----------|--|-----------|
| 2.6 | Summary and Conclusions | 36 |
| 3 | Calibration of Odometry and Sensor Extrinsic Parameters on Different Floors | 37 |
| 3.1 | Introduction | 37 |
| 3.2 | Conventional Differential Drive Robot Odometry Model | 39 |
| 3.3 | Real-time Odometry Calibration on Multiple Floors | 41 |
| 3.3.1 | Calibration on One Floor with EKF | 41 |
| 3.3.2 | Calibration on Multiple Floor Types | 44 |
| 3.3.3 | Experiments and Results | 47 |
| 3.4 | Offline Calibration of Odometry and Sensor Poses on Multiple Floors | 55 |
| 3.4.1 | Quality Metric for Occupancy Grid Maps | 56 |
| 3.4.2 | Fusion of Two Laser Rangefinders Measurements | 57 |
| 3.4.3 | Particle Swarm Optimisation | 59 |
| 3.4.4 | Calibration Method | 60 |
| 3.4.5 | Experiments and Results | 63 |
| 3.4.6 | Convergence Basin Verification | 70 |
| 3.5 | Summary and Conclusions | 73 |
| 4 | Odometry Model Incorporating Linear Acceleration | 74 |
| 4.1 | Introduction | 74 |
| 4.2 | Tyre Road Interaction Theory | 78 |
| 4.2.1 | Longitudinal Slip | 78 |
| 4.3 | Time Synchronisation of Lasers and Encoders | 80 |
| 4.4 | Laser Distance Measurement Calibration | 82 |
| 4.5 | Experiments and Results | 84 |
| 4.5.1 | Constant Linear Acceleration Tests | 84 |
| 4.5.2 | Calibration Incorporating Linear Acceleration | 89 |
| 4.6 | Summary and Conclusions | 92 |
| 5 | Odometry Model for Curved Paths | 94 |
| 5.1 | Introduction | 94 |
| 5.2 | Literature Review of Odometry with Slip | 95 |
| 5.3 | Theoretical Investigation | 97 |

| | | |
|----------|---|------------|
| 5.3.1 | Radius of Curvature | 97 |
| 5.3.2 | Effective B due to Tyre Vertical Elastic Displacement . . . | 99 |
| 5.4 | Linear Dependency of B with the Square of Robot Speed | 103 |
| 5.5 | Model of Wheel Radius and Wheel Angular Velocity | 107 |
| 5.5.1 | Nonlinear Regression with Saturation Growth Model | 107 |
| 5.5.2 | Model Verification with Independent Datasets | 112 |
| 5.6 | Calibration of Wheel Separation on Circular Paths | 115 |
| 5.7 | Experiments on a Carpet | 118 |
| 5.8 | Five Parameter Differential Drive Mobile Robot Odometry Model | 123 |
| 5.8.1 | Verification Experiments | 124 |
| 5.9 | Summary and Conclusion | 133 |
| 6 | Summary | 134 |
| 6.1 | Summary | 134 |
| 6.2 | Future Work | 136 |
| 6.3 | Concluding Remarks | 140 |
| | Appendices | 142 |
| A | Derivation of Jacobian with a Rank 2 on Constant Curvature | |
| | Turning in Subsection 3.3.1 | 143 |
| B | Investigation of Robot Slip Angle Effects | 150 |
| B.1 | Theoretical Analysis | 150 |
| B.2 | Robot Slip Angle Measurements with Circular Paths | 151 |
| | References | 157 |

List of Tables

| | | |
|-----|---|-----|
| 2.1 | Confusion Matrix of the Validation Test. Reused from [1] ©2015 IEEE. | 21 |
| 2.2 | Confusion Matrix of Floor Classification on Compensated Motor Current with a Median Filter on a Data Length of 0.2 m. Reused from [2] ©2016 IEEE. | 33 |
| 3.1 | Ground Truth Odometry Parameters in Simulation. Reused from [1] ©2015 IEEE. | 49 |
| 3.2 | Numbers With Relative Errors Less Than 0.5% in 10,000 Simulation Odometry Calibration Tests. Reused from [1] ©2015 IEEE. | 49 |
| 3.3 | Odometry Calibration Results. Reused from [1] ©2015 IEEE. | 55 |
| 3.4 | Confinement of Each Calibration Parameter in the PSO. Adapted from [3] ©2015 IEEE. | 64 |
| 3.5 | Calibration Results Using Dataset 1 and Dataset 2. Adapted from [3] ©2015 IEEE. | 69 |
| 4.1 | Odometry Calibration Results Using Different Models on the Same Real Environment Dataset. Reused from [4] ©2017 IEEE. | 91 |
| 5.1 | Relationship between Radius of Curvature of Robot Paths and Wheel Velocities without Slippage. | 98 |
| 5.2 | Curve Fitting Results (RMSD and MAE) of Wheel Radii and Angular Velocities. | 111 |
| 5.3 | Optimised Parameters of the Complete Odometry Model with a PSO in the Experiments in Figure 5.24. | 127 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | (a) Map obtained by the raw uncalibrated odometry of a robot with unevenly inflated tires traveling along a corridor. The result of applying a scan-matching algorithm with a large search space to account for the uncalibrated odometry leads to the shortened map shown in (b). A restriction of the search space is not able to fully correct the errors as visualized in (c). However, applying the correct calibration together with a small search space leads to an accurate estimate depicted in (d). Reproduced from [5] ©2011 IEEE. | 2 |
| 2.1 | ActivMedia Pioneer2 DX H-8 differential drive robot and sensors. Reused from [2] ©2016 IEEE. | 12 |
| 2.2 | Hardware schematic of the floor surface classification system. Adapted from [3] ©2015 IEEE. | 13 |
| 2.3 | Floors where the training and testing data for classification were collected. The camera was set about 1 metre to the ground. (a) Carpet 1 (b) Lino (c) Blue Rubber (d) Carpet 2. Adapted from [2] ©2016 IEEE. | 14 |
| 2.4 | Motor current and inclination angle detection system hardware structure. Reused from [2] ©2016 IEEE. | 16 |
| 2.5 | ROS Computational Graph Level [6]. | 17 |
| 2.6 | System schematic with variable definitions. Reused from [1] ©2015 IEEE. | 18 |

| | | |
|------|--|----|
| 2.7 | Light intensity detection results when the robot travelled from Carpet 1 to Lino (appearance shown as in Figure 2.8). (a) Original results. (b) Results after a median filter (window size is 25) and downsampling process (every 80 samples). Reused from [1] ©2015 IEEE. | 19 |
| 2.8 | Surfaces where the training data were collected. (a) Carpet 1 (b) Lino (c) Carpet 2 (d) Marble (e) Carpet 3 (f) Blue Rubber. Reused from [1] ©2015 IEEE. | 20 |
| 2.9 | The training data of the SVM classifier. Each class has 414 feature vectors, which comprise the intensities of the red, green and blue components in the light from the surfaces (a-f), respectively, in Figure 2.8. Reused from [1] ©2015 IEEE. | 20 |
| 2.10 | Left wheel motor current when the robot was running at a speed of 0.05 m/s in the air. Reused from [2] ©2016 IEEE. | 22 |
| 2.11 | Motor current showing a repeat pattern and anti-symmetrical effect on a return straight path on Carpet 2. Reused from [2] ©2016 IEEE. | 22 |
| 2.12 | Searching for the optimal anti-symmetrical starting point for the first return path in Figure 2.11. Reused from [2] ©2016 IEEE. . . | 23 |
| 2.13 | Effect of the strong negative correlation of the motor current on a return robot path between the forward and backward paths. Reused from [2] ©2016 IEEE. | 24 |
| 2.14 | Schematic of the usage of the accelerometer for inclination angle detection. Reused from [2] ©2016 IEEE. | 25 |
| 2.15 | (a) Inclination angles measured in degrees on a return straight path on Carpet 2. (b) Motor current in the same experiment as (a). Reused from [2] ©2016 IEEE. | 25 |
| 2.16 | Pearsons product-moment correlation coefficients between the median filtered average motor current and the inclination angles on four floors. Reused from [2] ©2016 IEEE. | 26 |
| 2.17 | Motor current from which the training data were extracted. (a) Carpet 1 (b) Lino (c) Blue rubber (d) Carpet 2. Reused from [2] ©2016 IEEE. | 27 |

| | | |
|------|--|----|
| 2.18 | Linear relationships between the motor current and the accelerometer measured inclination angles when the robot was driving straight forward. (a) Carpet 1 (b) Lino (c) Blue Rubber (d) Carpet 2. Note that the slopes are approximately the same and consistent with expectation that the change in motor with inclination is independent of the surface. Reused from [2] ©2016 IEEE. | 28 |
| 2.19 | Flow chart of one loop for floor classification with compensated motor currents. Reused from [2] ©2016 IEEE. | 29 |
| 2.20 | Median filtered average motor current. (a) Training data on Carpet 1 (b) Training data on Lino (c) Training data on Blue Rubber (d) Training data on Carpet 2 (e) Testing data on Carpet 1 (f) Testing data on Lino (g) Testing data on Blue Rubber (h) Testing data on Carpet 2. Reused from [2] ©2016 IEEE. | 30 |
| 2.21 | Standard deviation of the median filtered average motor current used for floor classification. (a) Training data (b) Testing data. Reused from [2] ©2016 IEEE. | 31 |
| 2.22 | Effect of data segment length on classification accuracy. Reused from [2] ©2016 IEEE. | 32 |
| 2.23 | Features for floor classification. (a) Mean of the motor current (b) Variance of the motor current. Reused from [2] ©2016 IEEE. | 34 |
| 2.24 | Floor classification experiment with the same bed sheet. (a) Bed sheet on Lino (b) Bed sheet on Carpet 1. Reused from [2] ©2016 IEEE. | 35 |
| 2.25 | Mean motor current values of the training and testing data for floor classification on a bed sheet. Reused from [2] ©2016 IEEE. | 35 |
| 3.1 | Coordinate frame of conventional mobile robot odometry system in Equation (3.3-3.5). | 40 |
| 3.2 | The sketch map of the continuous differential drive mobile robot odometry calibration on Carpet 1 and Blue Rubber, on which the robot trajectories are marked with red and blue lines, respectively. Reused from [1] ©2015 IEEE. | 41 |

| | | |
|------|---|----|
| 3.3 | System structure of the EKF based odometry calibration with floor surface classification using SVM. Reused from [1] ©2015 IEEE. | 46 |
| 3.4 | Robot trajectory in a simulated calibration with the proposed method. The red, green and blue lines show the robot trajectories on different surfaces. Reused from [1] ©2015 IEEE. | 48 |
| 3.5 | Simulated calibration results of \hat{R}_l , \hat{R}_r and \hat{l} on three floors. The blue curves show two standard deviation boundary around the estimated variables. The robot travelled on Floor 1, Floor 2 and Floor 3 successively. Reused from [1] ©2015 IEEE. | 48 |
| 3.6 | Histogram and Gaussian fit of the relative errors of \hat{R}_l on Floor 1 after 10,000 repeated simulated odometry calibrations. Reused from [1] ©2015 IEEE. | 50 |
| 3.7 | Evolutions of the standard deviations of odometry parameters during the robot experiment. Reused from [1] ©2015 IEEE. | 51 |
| 3.8 | Calibration results on Carpet 1 and Blue Rubber. Blue curves in top three plots show ± 2 standard deviations from estimated variables in red. Reused from [1] ©2015 IEEE. | 52 |
| 3.9 | Reference graph map for odometry correction cost comparison. The robot started performing SLAM from the bottom right corner of the map on Carpet 1. Then it was steered clock-wise on Carpet 1 and Blue Rubber, on which the trajectories were shown as red and blue lines, respectively. Green dots represent robot positions and yellow lines represent relative robot poses. Reused from [1] ©2015 IEEE. | 53 |
| 3.10 | The odometry correction cost with and without the proposed floor classification strategy. (a) The odometry correction cost in rotation. (b) The odometry correction cost in translation. Reused from [1] ©2015 IEEE. | 54 |
| 3.11 | Schematic of the laser range finders on the robot. The red and green sectors show the maximum measurement ranges of Laser 1 and Laser 2, respectively. Reused from [3] ©2015 IEEE. | 58 |
| 3.12 | Flow chart of calibrating odometry and sensor extrinsic parameters with PSO on multiple floors. Reused from [3] ©2015 IEEE. | 62 |

| | | |
|------|--|----|
| 3.13 | (a) The environment used to collect the datasets. The size of the environment was approximately 4 m × 5 m. (b) The robot is crossing the boundary of two floors, Carpet 1 and Blue Rubber. Reused from [3] ©2015 IEEE. | 64 |
| 3.14 | The map information evolutions during the simultaneous calibration of odometry and laser pose parameters vs. the number of generations in PSO algorithm when using the proposed floor classification method and a wrong floor classification. Reused from [3] ©2015 IEEE. | 66 |
| 3.15 | The evolution of the mean of the map information values of the 40 particles in each generation in the PSO algorithm. Reused from [3] ©2015 IEEE. | 66 |
| 3.16 | Mapping results to show the influence of floor classification. The trajectories labelled with red and blue correpond to the classified results of floor types, Carpet 1 and Blue Rubber, respectively. Green dots depict positions of the robot for the laser scans used to build the OG maps and yellow lines depict the relative pose between the consecutive laser scans. (a) OG map built with the calibrated parameters based on the proposed floor classification method using Dataset 1. (b) OG map built with the calibrated parameters based on a wrong floor classification as a comparison using Dataset 1. (c) OG map built with the parameters in (a) using Dataset 2. (d) OG map built with the parameters in (b) using Dataset 2. Reused from [3] ©2015 IEEE. | 67 |
| 3.17 | OG map built with the calibrated parameters based on the proposed floor classification method using Dataset 2. Reused from [3] ©2015 IEEE. | 68 |
| 3.18 | Relative calibration differences based on Dataset 1 and Dataset 2 using Equation (3.22). Reused from [3] ©2015 IEEE. | 69 |
| 3.19 | Experiment with an “8” shape trajectory on Lino to verify the convergence basin of the odometry parameters.(a) Surrounding environment. (b) Running. | 71 |

| | | |
|------|--|----|
| 3.20 | Convergence basin for the odometry calibration with the robot following an “8” shape trajectory on Lino. (a) Mesh plot with contour when fixing <i>delta</i> . (b) Another view of (a). (c) Three slices of odometry parameters. | 72 |
| 4.1 | One typical Magic Formula tyre model curve. Reused from [4] ©2017 IEEE. | 79 |
| 4.2 | Front and rear laser registration errors with beam synchronisation adjustment for robot straight line motion. (a) Before. (b) After. Reused from [4] ©2017 IEEE. | 81 |
| 4.3 | Front and rear laser registration errors with beam synchronisation adjustment for robot rotation on the spot motion. (a) Before. (b) After. Reused from [4] ©2017 IEEE. | 82 |
| 4.4 | Experimental setup for calibration of the laser rangefinders using tape measurements. Reused from [4] ©2017 IEEE. | 83 |
| 4.5 | Results of calibration of the scales and offsets of the LRFs using the setup in Figure 4.4. (a) Front LRF. (b) Back LRF. Reused from [4] ©2017 IEEE. | 84 |
| 4.6 | Square environment for experiments of linear acceleration scenarios. Reused from [4] ©2017 IEEE. | 85 |
| 4.7 | Robot moving distances over one straight path using conventional odometry estimation with equal and constant (0.098 m) wheel radius assumption for left and right wheels. Distance measurements using scans are treated as ground truth. (a) Accelerating. (b) Decelerating. | 86 |
| 4.8 | Linear correlation between the distance measurements in Figure 4.7 using odometry estimation and scans. (a) Accelerating. (b) Decelerating. Reused from [4] ©2017 IEEE. | 86 |
| 4.9 | Effective wheel radius. The odd (even) indices traverse the floor in the North (South) direction. (a) Acceleration = 0.3 m/s ² ; (b) Acceleration = 0.45 m/s ² . Reused from [4] ©2017 IEEE. | 87 |

| | | |
|------|---|-----|
| 4.10 | Summary of wall experiments with ± 2 standard deviation error bars confirming the linear variation of effective wheel radius with acceleration. Reused from [4] ©2017 IEEE. | 87 |
| 4.11 | Top right corner (1 m by 1 m) of the occupancy grid map built by attaching 245 scans to robot pose estimation results after more than 300 generations of PSO based odometry calibration. (a) Using optimal odometry parameters of the developed model. (b) Using opposite sign of k_a , i.e., 0.00726184 s ² . Reused from [4] ©2017 IEEE. | 91 |
| 5.1 | Instantaneous centre of curvature (ICC) of a differential drive robot. Adapted from [7]. | 97 |
| 5.2 | Schematic graph of the forces of interest. | 99 |
| 5.3 | Robot and its surrounding environment in the dataset collected on September 27, 2017. | 104 |
| 5.4 | Calibration results of wheel separation distance B for datasets on September 27, 2017. | 104 |
| 5.5 | 2D view of estimated wheel separation distances B for datasets on September 27, 2017. (a) B vs. ω . (b) B vs. v^2 . (c) B vs. Radius of Curvature | 105 |
| 5.6 | Calibration results of effective wheel separation for radius of curvature greater than $B/2$ cases of datasets on September 27, 2017. | 106 |
| 5.7 | Estimate B using a ruler. (a) Left wheel. (b) Right wheel. | 107 |
| 5.8 | Robot and its surrounding environment in Subsection 5.5.1 and Section 5.6. | 108 |
| 5.9 | Robot speeds on straight paths. | 108 |
| 5.10 | Wheel radii variation with respect to wheel angular velocities. (a) Left wheel. (b) Right wheel. | 109 |
| 5.11 | Curve fitting of wheel radii with respect to wheel angular velocities. (a) Left wheel using exponential model fitting. (b) Right wheel using exponential model fitting. (c) Left wheel using saturation growth model fitting. (d) Right wheel using saturation growth model fitting. | 111 |

| | | |
|------|---|-----|
| 5.12 | Robot and its surrounding environment in Subsection 5.5.2. . . . | 113 |
| 5.13 | Robot speeds of each straight paths of datasets in Subsection 5.5.2. | 113 |
| 5.14 | Curve fitting of wheel radii with respect to wheel angular velocities of datasets in Subsection 5.5.2. (a) Left wheel using exponential model fitting. (b) Right wheel using exponential model fitting. (c) Left wheel using saturation growth model fitting. (d) Right wheel using saturation growth model fitting. | 114 |
| 5.15 | Normal fitting results of five repeating experiments in Figure 5.14 with error bars of ± 2 standard deviation in both axes. (a) Left wheel. (b) Right wheel. | 114 |
| 5.16 | Calibration results of wheel separation distances when applying the saturation growth model built in Figure 5.11 (c) and (d). (a) B vs. ω . (b) B vs. v^2 . (c) B vs. Radius of Curvature. | 117 |
| 5.17 | Partial calibration results of B on a lino floor when using saturation growth model for wheel radii and wheel angular velocities under the assumption in Subsection 5.3.2. (a) B vs. ω . (b) B vs. v^2 . . . | 118 |
| 5.18 | Robot speeds during straight running tests. | 119 |
| 5.19 | Robot and its surrounding environment in Section 5.7 (a) Straight running paths. (b) Circular running paths. | 119 |
| 5.20 | Curve fitting of wheel radii with respect to wheel angular velocities for datasets on Carpet 2. Note that the travel direction affects wheel radii due to the directional surface of the carpet or floor inclination. (a) Left wheel using exponential model fitting. (b) Right wheel using exponential model fitting. (c) Left wheel using saturation growth model fitting. (d) Right wheel using saturation growth model fitting. | 120 |
| 5.21 | 3D view of the calibration results of wheel separation on a carpet. | 121 |
| 5.22 | Calibration results of wheel separation distance B when using saturation growth model for wheel radii and wheel angular velocities on a carpet. (a) B vs. ω . (b) B vs. v^2 . (c) B vs. Radius of Curvature. | 122 |

| | | |
|------|---|-----|
| 5.23 | Partial calibration results of B on a carpet when using saturation growth model for radii of curvature greater than $B/2$ cases on a carpet. (a) B vs. ω . (b) B vs. v^2 | 123 |
| 5.24 | Testing environment for a robot running with different speeds and angular velocities. | 125 |
| 5.25 | Robot speeds and accelerations in Subsection 5.8.1. (a) Robot speeds. (b) Robot linear accelerations. | 126 |
| 5.26 | Mean Map Information over 500 generations of PSO in Subsection 5.8.1. | 127 |
| 5.27 | Maximum Map Information over 500 generations of PSO in Subsection 5.8.1. | 128 |
| 5.28 | Evolutions of five independent variables in the PSO process. (a) R_0 . (b) $R_{\infty l}$. (c) $R_{\infty r}$. (d) τ . (e) B | 129 |
| 5.29 | Grid searching of one of five independent variables in the complete odometry model by fixing the other four parameters to be the optimised result in Table 5.3. (a) R_0 . (b) $R_{\infty l}$. (c) $R_{\infty r}$. (d) τ . (e) B | 130 |
| 5.30 | Zoom-in view of two objects in the testing environment. | 131 |
| 5.31 | OG map generated using optimised result in Table 5.3. There are 283 different laser scans by sampling every 0.3 m travelling distance and 15 degree rotation of the robot. (a) OG map. (b) Zoom-in view. Using the distance measurement tool in the ROS visualisation tool, <i>rviz</i> , the author measures a distance of 0.717 m in the grid map between the left edge of the wood block and the right edge of the white foam block. This distance is very close to the physical measurement of 0.636 m (i.e., summing the widths of all three segments) in Figure 5.30 considering there are 283 scans over a relatively long period and running distances. | 132 |
| B.1 | Schematic of robot slip angle on circular paths. Solid black dots represent robot positions. O represents the centre of a fitted circle of robot positions. \overrightarrow{PH} and \overrightarrow{PT} represent the vehicle heading and the path tangent, respectively. $\overrightarrow{PT} \perp \overrightarrow{OP}$ | 152 |

| | | |
|-----|--|-----|
| B.2 | The testing environment for measuring the robot slip angles on circular paths on Lino. (a) Robot is being charged. (b) Robot is automatically driving in a circle. | 153 |
| B.3 | Scan matching results using NDT algorithm in MATLAB when measuring robot slip angles on circular paths on Lino. (a) Example scan matching. (b) Robot poses. | 154 |
| B.4 | (a) OG map generated using the scan matching results of robot poses and its related scans during the circular movement on Lino. The OG map is built by attaching 2,864 laser scans to robot poses. (b) Zoom in view of Figure B.4 (a). | 154 |
| B.5 | Robot slip angles during the circular movement on Lino. | 155 |
| B.6 | Robot slip angle estimations of different datasets. (a) β vs. radius of trajectory. (b) β vs. robot speed. (c) β vs. robot angular velocity. (d) β vs. robot centrifugal acceleration. | 156 |

Indoor Robot Odometry Modelling and Calibration Incorporating Floor Surfaces and Acceleration

Yanming Pei

Monash University, Australia, 2018

Supervisor: Associate Professor Lindsay Kleeman

1

Abstract

One of the fundamental problems in mobile robotics is simultaneous localisation and mapping (SLAM), where one or more robots build a map and localise themselves or other objects within the map at the same time. SLAM often uses robot odometry as a reliable short term pose estimation. Robot odometry estimates a robot's position and orientation from measuring wheel angle changes. Calibration of odometry involves estimating wheel radii and separation distance. This thesis aims to devise novel odometry calibration methods that can detect the difference of multiple floors in the real time, incorporate robot linear accelerations and robot path curvature to significantly improve the localisation and map accuracy.

Floor sensors allow a robot to segment the environment into useful regions with properties associated with floors. Two independent floor sensing systems are developed with colour intensity and motor current sensors. Sensor data is processed with a Support Vector Machine (SVM) and experimentally justifies the accuracy of these floor classification systems.

Floor dependency is often mentioned in the literature of odometry calibration, but little effort has been dedicated to investigate odometry calibration on heterogeneous floors. Two independent odometry calibration methods are developed. The first one uses an Extended Kalman Filter (EKF) to provide online odometry calibration. An odometry correction cost function is derived from graph SLAM to show experimentally that the calibration with multiple classified regions is superior to calibration without floor classification. The second one uses a Particle Swarm Optimisation (PSO) algorithm to optimise the Occupancy Grid (OG) map generated with raw encoder data and laser range finders (LRFs) data and achieves offline odometry calibration. These two systems provide experimental results confirming that odometry calibration parameters depend on floor surface conditions.

The odometry model is extended with a fourth parameter that accounts for wheel slip proportional to linear acceleration of a differential drive robot. To the author's knowledge, this is the first robot odometry model that accounts for linear acceleration. Experimental results show an error range of 9% in the estimation of the wheel radii when a Pioneer2 DX H-8 indoor mobile robot accelerates and decelerates on straight paths at an acceleration value of $\pm 0.45 \text{ m/s}^2$.

The odometry model is further extended to model curved paths, where the wheel radii depend on the velocity and centrifugal accelerations. Research findings are justified on a hard lino floor and a soft carpet floor. At the end of this thesis, a complete model is proposed for a differential drive mobile robot equipped with pneumatic tyres and is verified experimentally.

Indoor Robot Odometry Modelling and Calibration Incorporating Floor Surfaces and Acceleration

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Yanming Pei
May 12, 2018

Acknowledgements

First and foremost, I would like to sincerely thank my main supervisor, A/Prof. Lindsay Kleeman for his extraordinary guidance and kindest help during my PhD candidature. I really appreciate his insightful research ideas, constant encouragement and infinite amount of patience. He is also very generous with his time to have deep and fruitful discussions with me in a meeting no matter how disappointing my experimental results are. These unforgettable experiences can always inspire me to keep calm and carry on with stronger motivation to solve novel research problems. The completion of this thesis would not be possible without Lindsay. He also contributed most of the content in Appendix A of this thesis. I would also like to thank my associate supervisor, A/Prof. Andy Russell for his advice.

Thanks also go to my friends and colleagues in the Department, especially those in Robotics group. Generously, they gave me kind help, warm regards and enjoyable board games experiences. They are Vincent Lui, Andrew Spek, Graeme Best, Dinesh Gamage, Horace Josh, Max Xiaoqin Wang, Benjamin Harwood, Yan Zuo, Anindya Harchowdhury, Titus Tang, Benjamin Meyer, Thanuja Dharmasiri, Timo Stoffregen, Nalika Dona, Winston Yii, Gil Avraham, Josh Weberruss and the rest. Special thanks to Vincent and Andrew, who have kindly helped me in terms of coding and using Ubuntu in the early period of my PhD candidature. In addition, I thank Fredy Tungadi for setting up the framework of laser range finders on top of the robot, which was used in my experiments.

I also acknowledge the suggestions of Prof. Tom Drummond, A/Prof. Mehmet Yuce, Dr. Wai Ho Li, Dr. Damien Browne, Dr. David Boland, who formed the committee of three milestone reviews of my PhD candidature.

I thank Monash University for providing me Monash International Postgraduate Research Scholarship and Monash Graduate Scholarship. Meantime, I am also grateful to Department of Electrical and Computer Systems Engineering and Faculty of Engineering for partially supporting my travel costs and conference registration costs to five different international academic conferences.

Finally, I would like to thank my parents, who have not only brought me up but also taught me to develop kind personality. I also thank my parents in law and my brother in law, who have strongly supported and encouraged me through the journey. Last but not the least, I thank my wife, Shuang Zhang, for her continuous love, support, encouragement, patience and understanding during the past 11 years. Her support is the spiritual energy keeping me forward and I appreciate it very much.

Yanming Pei

Monash University, Australia
May 2018

Introduction

Wheeled mobile robots are arguably the most common types of robots around the world. They can be seen in various scenarios, such as rescue missions [8, 9], domestic environment [10], on streets [11], for education [12] and in farms [13, 14]. These wheeled robots often rely on their inbuilt wheel encoders, gyroscopes and accelerometers to perform mapping and localisation tasks [5]. These proprioceptive sensors can provide direct measurements of robot poses albeit with errors that drift with time.

Odometry calibration is important not only because the simple and inexpensive wheel encoders may be the only localisation sensors for a robot but also wrong odometry calibration may lead to a poor mapping results. In particular, a wheeled robot commonly uses odometry pose estimation for mapping and localisation. Kümmerle et al. [5] shows in Figure 1.1 that well calibrated proprioceptive sensors, such as odometry, help improve not only robot navigation but also map quality even after having taken advantage of laser scan matching. In [15], Kelly also pointed out that a proper odometry calibration can help to reduce the difficulty of performing loop closure, which can significantly shrink the errors of mapping and localisation tasks. Similarly, Burgard et al. [16] claim that more accurate odometry estimation can reduce robot pose correction in a loop closure situation and scan matching during map construction, thereby reducing errors. The highly cited article by Cox [17] mentions that map-matching combined with dead-reckoning allows fast robust matching, which leads to accurate knowledge of position.

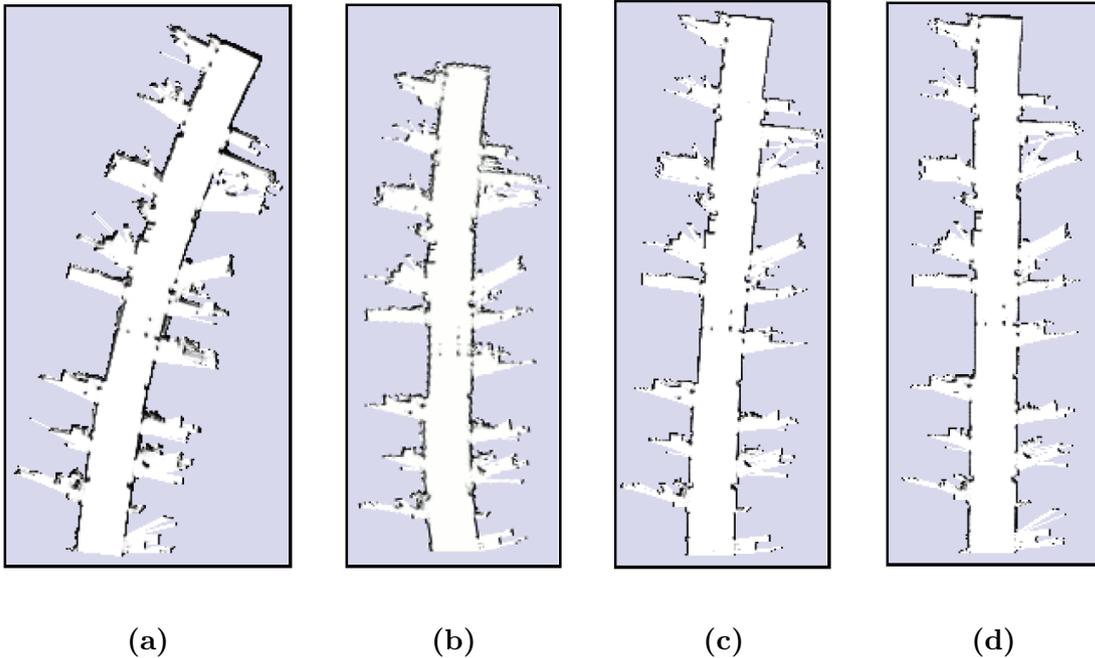


Figure 1.1: (a) Map obtained by the raw uncalibrated odometry of a robot with unevenly inflated tires traveling along a corridor. The result of applying a scan-matching algorithm with a large search space to account for the uncalibrated odometry leads to the shortened map shown in (b). A restriction of the search space is not able to fully correct the errors as visualized in (c). However, applying the correct calibration together with a small search space leads to an accurate estimate depicted in (d). Reproduced from [5] ©2011 IEEE.

Like other measurement methods, there are systematic odometry errors and non-systematic odometry errors in the odometry system as mentioned in [15, 18, 19, 20, 21]. The commonly known systematic errors are caused by kinematic imperfections of the robot [21]. They include unequal wheel radii, uncertainty about the wheel separation distance, limited encoder resolution and so on [21]. Non-systematic errors include travelling over uneven floors, travelling over unexpected objects and wheel-slippage (due to slippery floors, over-acceleration and fast turning)[21]. Some research papers [18, 19] focus on analysing and modelling the non-systematic errors and the odometry covariance matrix for special trajectories. Some other papers [21, 22] target calibrating systematic errors. Kelly [15] develops a general technique to be used for the extraction of unknown parameters of either systematic odometry models or non-systematic error models.

In contrast to all the previously mentioned research, the author regards the following three factors as systematic error sources. Novel odometry models incorporating them are developed and calibrated in this thesis.

- Surface of floors where mobile robots travel on. Different surfaces may have different friction coefficients, which will affect odometry estimation in a systematic way. Details are explained in Chapter 2 and Chapter 3.
- Constant linear acceleration or deceleration of the robot. They can cause a relatively constant wheel slip ratio of pneumatic tyres on a certain uniform surface. Hence the effective wheel radii should be calibrated accordingly. Details are explained in Chapter 4.
- Robot slip angle and cornering force on pneumatic tyres under a constant centrifugal acceleration. Details are explained in Chapter 5

1.1 Contributions

- **Online Odometry Calibration on Multiple Floors**

Mobile robot odometry parameters depend on tyre pressure, wear, robot load and surface type. The surface dependency is often quoted in the literature [5, 23, 24], but no direct experiments have been reported on different surfaces. These experiments must be conducted within a short time duration of each other so as to eliminate any calibration time drift.

A system to perform online mobile robot odometry calibration on multiple floors is developed. Experimental results are shown to prove the floor classification method with a colour intensity sensor is effective using a Support Vector Machine (SVM) [25]. This can improve SLAM map quality by segmenting the environment into distinct regions with different odometry parameters. This system achieves greater than 98% precision and recall values for a testing dataset consisting of six different floors. The Extended Kalman Filter (EKF) and correlative laser scan matching [26] are used to calibrate the odometry parameters. Using an odometry correction cost function derived from graph SLAM in [16] to show experimentally that the calibration with multiple classified regions is superior to calibration without floor classification. This confirms that odometry parameters do depend on the floor surface, a result which has not been published by other researchers. In addition, the system can detect a transition to a new floor region and then start a new odometry calibration process. It can also detect a transition back to a previously visited region and then continue the last suspended calibration with the calibration results on that region. This is all done in real time without stopping the robot when changing floors. Details about this contribution are described in Chapter 2 and Chapter 3

- **Simultaneous Calibration of Odometry and Sensor Extrinsic Parameters on Multiple Floors**

This research also develops a system which can calibrate mobile robot odometry and sensor extrinsic parameters simultaneously on multiple floors. This can significantly improve the accuracy of robot mapping and localisation. The calibration is achieved by optimising Map Information (MI) quality for Occupancy Grid (OG) maps generated with odometry estimation and laser rangefinder (LRF) measurements using the standard Particle Swarm Optimisation (PSO) algorithm. The benefit of the proposed floor classification system is demonstrated by real world experiments. Furthermore, the consistency of the calibration method is also investigated with different datasets. To the author's knowledge, this is also the first research that applies PSO in odometry and sensor poses calibration problems. This is the first simultaneous sensor pose and odometry calibration approach to take direct account of floor dependency. Note that the map quality does not require any ground truth map. This is a data driven method described in Chapter 2 and Chapter 3.

- **Floor Classification with Motor Current**

This research develops a novel floor classification system for mobile robots based on motor current measurements compensated linearly by floor inclination angle in the real time. The motor current is proportional to the rolling resistance of a flat floor when a robot travels at constant velocity. Small deviations in the inclination of indoor floors less than one degree are shown to be sufficient to corrupt this measurement. Compensation for the inclination angle with a low cost accelerometer is successfully implemented. Using a SVM classifier, this system achieves an accuracy of 95% with 0.2 m travelling distance to classify 4 indoor floors, including two similar carpets. Experimental results show that this method significantly improves the previous floor classification system based on the colour sensor if floor colours are similar or same. This contribution is described in Chapter 2.

- **Odometry Model Incorporating Linear Acceleration**

Most, if not all, previous publications on indoor mobile robot odometry calibration methods assume no wheel slip as clearly stated in [15]. Minor wheel slip can affect the effective wheel radii estimation and result in errors in the odometry estimation system if not included in the calibration process.

A novel differential drive robot odometry model incorporating robot linear acceleration is developed in this thesis. A new fourth parameter is introduced that accounts for wheel slip proportional to linear acceleration of the robot. The model requires little extra computation. Accelerations can be obtained from robot motion commands. The new odometry model is validated via experiments on a hard lino floor. Experimental results show an error range of 9% in the estimation of the wheel radii when a Pioneer2 DX H-8 indoor mobile robot accelerates and decelerates on straight paths at an acceleration value of $\pm 0.45 \text{ m/s}^2$. The author uses another independent experiment to show a 0.3% improvement of the quality of the OG map [27] built using the three parameters model. The maps were generated by attaching laser scans to robot poses based only on odometry estimation. For fairness in the number of parameters, the proposed model is compared with another four parameter model. In addition, two different methods to calibrate the new model consisting of four parameters are introduced. This contribution is described in Chapter 4.

- **Novel Odometry Model on Curved Paths**

Chapter 5 investigates and develops a linear model between the wheel separation and the square of robot speeds. This model improves the estimation accuracy of wheel separation by 0.39%. In addition, a saturation growth model between wheel radius and wheel rotation speed is discovered. It is shown that the saturation growth model improves the estimation accuracy of wheel radius under normal working condition by 9.2%. These researching findings are justified on a hard lino floor and a soft carpet floor. At the end of this chapter, a five parameter model for differential drive mobile

robot equipped with pneumatic tyres is developed and justified with real experiments.

1.2 Publications

Published

1. Yanming Pei and Lindsay Kleeman, “A Novel Odometry Model for Wheeled Mobile Robots Incorporating Linear Acceleration”, in *Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA)*, 2017. **[Winner of Best Student Paper Award]**
2. Yanming Pei and Lindsay Kleeman, “Mobile Robot Floor Classification using Motor Current and Accelerometer Measurements”, in *Proceedings of IEEE International Workshop on Advanced Motion Control (AMC)*, 2016.
3. Yanming Pei and Lindsay Kleeman, “Robot Calibration of Sensor Poses and Region Based Odometry Using Offline Optimisation of Map Information”, in *Proceedings of IEEE International Conference on Information and Automation (ICIA)*, 2015. **[Best Paper Finalist Award]**
4. Yanming Pei and Lindsay Kleeman, “Online Robot Odometry Calibration over Multiple Regions Classified by Floor Colour”, in *Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA)*, 2015.

To be submitted

1. Yanming Pei and Lindsay Kleeman, “Robot Odometry Models and Calibration for Accelerating and Curved Paths”, to be submitted to *Journal of Field Robotics*.

1.3 Structure of the Thesis

Chapter 2 gives an introduction to two independent floor classification systems. One consists of a colour intensity sensor. The other uses a commonly available motor current sensor, where current sensing data are compensated by a floor inclination angle detection system. It details the software platform for the robot experiment development, i.e., Robot Operating System (ROS) and SVM algorithm for the floor classification purpose. The conventional differential drive mobile robot kinematic model is also introduced. Chapter 3 describes the developed online mobile robot odometry calibration on multiple floors based on an EKF. Chapter 3 also describes the devised method to simultaneously calibrate LRF extrinsic parameters and robot odometry parameters using PSO and Map Information for occupancy grid maps. Chapter 4 details the acceleration dependency of wheeled mobile robot odometry supported by theories in tyre mechanics. Chapter 5 details the odometry model for robots driving on curved paths by incorporating the nonlinear relationship between wheel radii and wheel rotation speeds. In the last chapter, Chapter 6 summarises this thesis and lists the future work.

Indoor Floor Classification with Sensors

2.1 Introduction

In typical scenarios, robots may drive on multiple types of floors. Floor sensors allow a mobile robot to segment the environment into useful regions with properties associated with the floor, such as odometry calibration, cleaning requirements and semantic map labelling. This chapter describes two different floor classification methods that work in real time with colour intensity sensors and motor current sensors. Their effectiveness is demonstrated through experiments. A SVM [25] is adopted as the classifier.

Robot Terrain Classification (RTC) is an open problem with many applications in robotics and computer vision. By measuring accelerations at the tip of a mechanical tactile probe, Giguere and Dudek [28] successfully classified 10 indoor and outdoor floors within a few seconds using an artificial neural network. Haldane [29] developed a heterogeneous team of legged robots to detect the slippery spots with an accuracy of 92%. This allows the main robot in the team to avoid possibly hazardous terrains. More recently, Walas [30] relied on the intensity values of the laser beams reflected from different terrains to classify the floors with a statistical approach called the texton approach, a Fourier Transform approach and an elevation map approach. All approaches in [30] are based on an SVM classifier. Brooks and Iagnemma [31] developed a self-supervised terrain classification framework consisting of a vibration-based classification approach and a traction-based classification approach. With a compact camera, Filitchkin and Byl [32] used a bag of visual words to classify different outdoor terrains with an SVM

classifier. Although they achieved high classification rates that were robust to illumination changes, the system could not operate in dark environments. Ojeda et al. [33] proposed a floor classification system with multiple sensors, such as gyros, accelerometers, motor current and voltage sensors, ultrasonic and microphones. Neural networks are adopted to differentiate the frequency components of the signals from sensors. This work does not include compensation for floor inclination angle variation that this thesis demonstrates is important. Best et al. [34] developed an SVM terrain classification approach with a Hexapod robot and its proprioceptive position sensors within leg servos aiming to improve the robot motion on different terrains. This thesis uses a similar training approach for supervised classification of floor colours. Posada et al. [35] developed a pure vision based floor segmentation algorithm with a robust performance over several hours of autonomous operation. However, this vision system can suffer from low illumination and the inability to classify the floor immediately below the robot due to the camera position on top of the robot.

In the literature, the colour of an object has been widely used as a feature in fire-like object detection systems [36] and vehicle identification systems [37]. Khan et al. [38] explored a variety of colour and texture descriptors and proposed a compact-texture description to classify textures. A comprehensive review about image category recognition was provided by Van et al. [39]. They evaluated a variety of colour descriptors regarding their invariance properties and distinctiveness with a recommendation of combined colour descriptors when no prior knowledge was available. Unlike these research papers focusing on the pattern recognition about objects or images, this thesis does not aim to develop a better algorithm or a descriptor. Instead, it intends to use a simple colour intensity based classification system to provide a floor classification result for the region based odometry calibration algorithm as in Chapter 3. Therefore this thesis introduces a colour sensor mounted underneath the robot that illuminates the floor from four directions. The colour sensor is sufficiently accurate to classify floor surfaces and provide almost immediate notification of changes in floor surface classification. Other work [5] developed an online graph based odometry calibration method by extending the graph-based formulation of SLAM. They could

find a floor transition or load change only after odometry parameters converge to significant values, which is not as effective nor precise for the timely classification of floor surfaces.

The author also develops a system to perform floor classification tasks where colour sensors fail to work effectively. For example, classification systems based on colour sensors will not distinguish floors with the same colour but different materials, or multiple floors with various colours on each floor. The system based on motor current sensors can compensate the motor current measurements using the floor inclination angle detection based on a low-cost accelerometer. The first five statistical moments of motor current sensing data are selected to be the features for training and testing of a SVM. The proposed motor current floor classification approach applies to indoor environments and is not dependent on lighting conditions. The motor current is proportional to the rolling resistance on a flat floor when the robot travels at a constant velocity. This chapter shows that commonly occurring small deviations of less than one degree in the inclination of indoor floors significantly affect motor current measurements. A classification accuracy of 95% for a 0.2 m travelling distance on four indoor surfaces including similar dark carpets is achieved. Experimental results show that floor inclination has a significant effect on the motor current for common indoor floors which are normally assumed to be flat.

In this chapter, Section 2.2 describes the robot sensing hardware system, including a robot, two laser rangefinders (LRFs), wheel encoders, a colour intensity sensor, a motor current sensing system and a floor inclination angle sensing system. A brief introduction of the Robot Operating System (ROS) is presented in Section 2.3. The colour sensing system designed to differentiate different floors is detailed in Section 2.4, where the author gives a short introduction to SVM followed by experimental results. In Section 2.5, the above mentioned floor classification system is developed in detail both in theoretic analysis and experimental tests. The last section, Section 4.6 summarises the chapter.

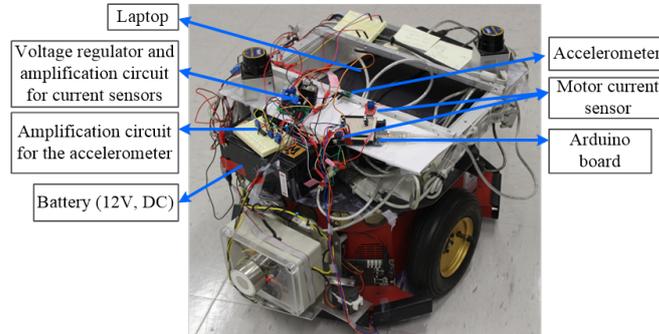


Figure 2.1: ActivMedia Pioneer2 DX H-8 differential drive robot and sensors. Reused from [2] ©2016 IEEE.

2.2 Robot Sensing Hardware System

2.2.1 Robot Platform

The robot used in this thesis is an ActivMedia Pioneer2 DX H-8 differential drive robot equipped with two Hokuyo URG-04LX LRFs and two driving wheels with encoders as shown in Figure 2.1. Each range finder has a field of view (FOV) of 240 degrees and valid range measurements from 0.02 m to 4 m. This setup provides the ability to fuse the measurements of two LRFs and generates a virtual LRF with a FOV of 360 degrees with benefits in the scan matching process [40].

2.2.2 Colour Sensing

Most indoor floor surfaces have unique colours or blends of colours. A detection system consisting of a colour light-to-frequency converter and Arduino UNO microcontroller has been designed.

This research uses a TAOS TCS230 colour sensor that measures the intensity of four channels corresponding to 16 phototransistors: 4 red, 4 blue, 4 green and 4 unfiltered. The floor is illuminated by 4 white LEDs from each side of a square

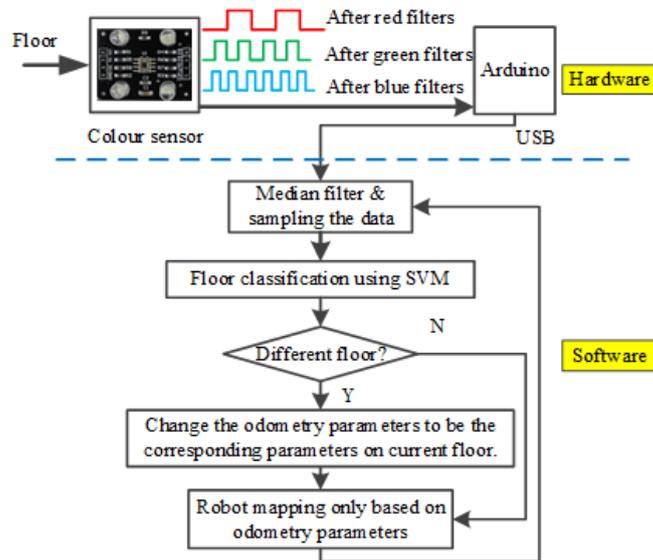


Figure 2.2: Hardware schematic of the floor surface classification system. Adapted from [3] ©2015 IEEE.

as in Figure 2.2. The TCS230 is interfaced to an Arduino Uno microcontroller that converts the frequency encoded intensity values from the TCS320 to integer format outputted onto a USB connection to a laptop. As shown in Figure 2.1, the colour sensor module is fixed under the robot 10 mm above the ground and the distance from the sensor to the center point of the two wheels is 200 mm. Mounting the colour sensor beneath the robot can reduce interference from other light sources or unwanted reflected light.

2.2.3 Motor Current Sensing

Torque and Current of a DC Motor

The robot wheel rolling resistance is a physical property of floor surfaces. It is defined as the force required to maintain a constant speed, not subject to aerodynamic forces, over a flat horizontal surface. The rolling resistance of a surface is assumed to be related to robot odometry properties on that surface,

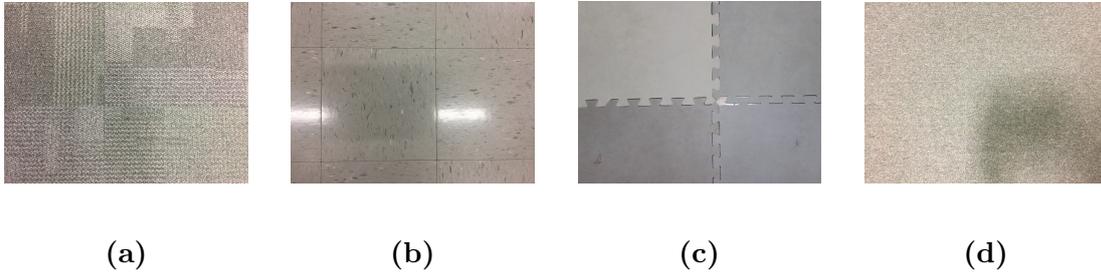


Figure 2.3: Floors where the training and testing data for classification were collected. The camera was set about 1 metre to the ground. (a) Carpet 1 (b) Lino (c) Blue Rubber (d) Carpet 2. Adapted from [2] ©2016 IEEE.

such as the effective wheel radii and separation distance. When running at a constant speed a DC motor provides torque proportional to armature current. This research focuses on differential drive robots powered by DC motors. As stated in [41], the total torque developed when an armature current I is passing through the coils of a DC motor is given by

$$T = K_T \Phi I \quad (2.1)$$

where Φ is the total flux produced by the magnetic field, and K_T is constant for a given motor. The flux is constant for most DC motors, so the motor torque T is directly proportional to the armature current I . As a result, the armature current is proportional to the rolling resistance when the robot is travelling on a flat horizontal surface. This section describes a motor current detection system and its performance on four indoor floors shown in Figure 2.3. Figure 2.4 shows the system hardware structure including the inclination angle detection part.

Hardware for Motor Current Sensing

With a standard laptop and other on-board electronics, the current of the DC motors never exceeded 1.4 Amps when the robot was driving at a speed of 0.05 m/s in a straight line on different indoor floors. Two hall-effect-based linear current sensors ACS712 were used to measure the motor current of each driving

wheel. This sensor was configured to work at the maximum sensitivity of 185 mV/A.

For a robot travelling at a constant velocity, measuring the robot tilt angle with an accelerometer is possible since gravity is the only acceleration stimulus. The measurement uses the projection of the gravity vector on one of the axes of an accelerometer [42].

To measure the floor inclination angle, the low power, low profile capacitive micromachined accelerometer, MMA7361L, was chosen. It has a high sensitivity of 800 mV/g when working at a range of $[-1.5, 1.5]$ g. Since the floor inclination angle was limited within $[-5, 5]$ degrees, the system was designed to operate within this range. The accelerometer was attached on top of the flat frame of the robot shown in Figure 2.14. The Y axis pointed to the front and it was used to compensate for the tilt angle. Note that in the classifier in Section 2.5 this system did not require a direct measurement of the tilt angle in degrees but only the calibrated tilt angle effect on the motor current. The current detection system was limited to measure only the positive current when the robot moved forward using a 10 bit Analogue-to-Digital Converter (ADC) on an Arduino Uno board. The motor current had a sampling rate of 152 Hz. If multiple robots are deployed around the same area, we can take advantage of their coordinating capabilities. For example, these robots can share the location based floor type information to each other using any internet of things technique, such as ZigBee or bluetooth. This can speed up the floor classification process and build a more robust classification system.

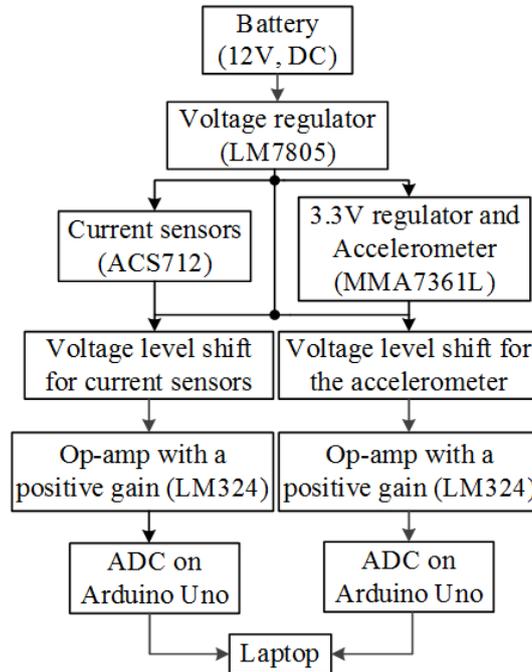


Figure 2.4: Motor current and inclination angle detection system hardware structure. Reused from [2] ©2016 IEEE.

2.3 Robot Operating System (ROS)

ROS [43] provides libraries, tools, hardware abstraction, device drivers, visualizers, message-passing, package management, state-of-the-art algorithms and more for robot developers worldwide. It is all open source and fully supported under Linux. A brief introduction to ROS is provided in its official website, <http://www.ros.org/>. The robot control algorithms is implemented based on *roscaria* package, which is an interface of using Advanced Robot Interface for Applications (ARIA). ARIA is a C++ library (software development toolkit or SDK) for all MobileRobots/ActivMedia platforms.

A robot operating system often runs a number of executables in parallel that need to exchange data synchronously or asynchronously. This objective is reached in the ROS Computation Graph [6] in Figure 2.5.

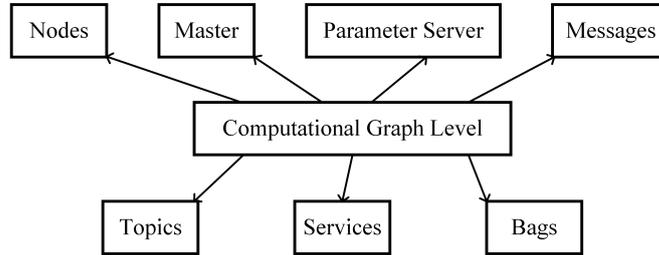


Figure 2.5: ROS Computational Graph Level [6].

- Nodes: They are processes that perform computation. A ROS system often contains many nodes to execute different tasks, e.g. controlling the wheels of a robot, performing path planning etc..
- Master: It provides name registration and lookup to the rest of the Computation Graph.
- Parameter Server: It allows data to be stored by key in a central location.
- Messages: It is a way to communicate between multiple nodes.
- Topics: It is a name that is used to identify the content of a message. Multiple nodes can publish and subscribe messages with the same topic name.
- Service: It provides synchronous communication between two nodes by request/reply interactions.
- Bags: It is a format for saving and playing back ROS message data. All the datasets in this thesis are stored in the format of *.bag*.

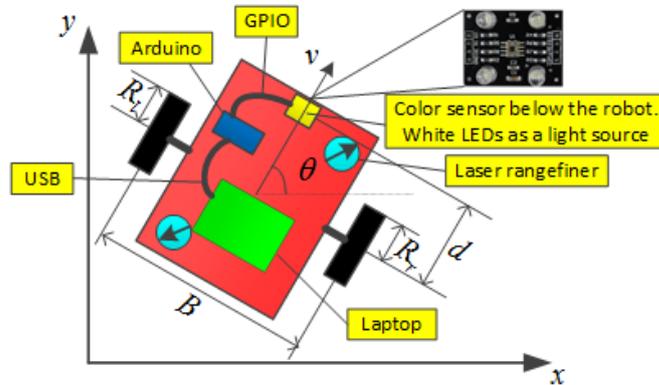


Figure 2.6: System schematic with variable definitions. Reused from [1] ©2015 IEEE.

2.4 Floor Classification with Colour Sensing

2.4.1 Support Vector Machine

SVM was first introduced in [44] in 1992. As a popular supervised learning algorithm, SVM was originally a binary classification model. Considering the research focus of this thesis is not proposing a new classification algorithm, the author refers readers to the other materials about SVM. In terms of implementation of SVM, the open-source LIBSVM library [45] was utilised in both offline MATLAB implementation and online C++ implementation with ROS *ml_classifier* package. The suggested Gaussian Radial Basis (RBF) kernel function [45] is adopted, because the linear kernel function and the sigmoid kernel function behave like RBF for certain kernel parameters and polynomial kernels have more numerical difficulties compared to RBF. One-vs-one multi-class decision algorithm [46] was utilised. This system uses the grid based search for the shape parameter and the soft margin.

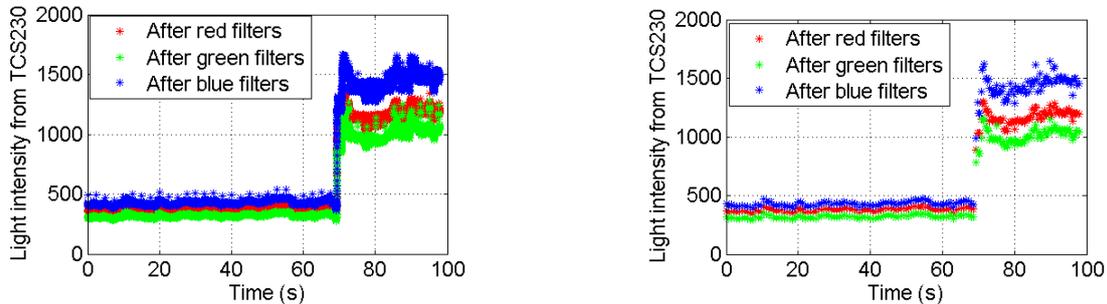


Figure 2.7: Light intensity detection results when the robot travelled from Carpet 1 to Lino (appearance shown as in Figure 2.8). (a) Original results. (b) Results after a median filter (window size is 25) and downsampling process (every 80 samples). Reused from [1] ©2015 IEEE.

2.4.2 Experiments and Results

The intensities of the red, green and blue components of light from the floor formed the feature vectors of SVM. Before training and classification, the original intensity data were processed in two steps. The first one was a median filter with a window size of 25. The second step was to down-sample the filtered data to a frequency of approximately 2 Hz. This step was to get a clear separation between the data from two adjacent floors. These pre-processes would reduce the confusion for classification work of SVM as shown in Figure 2.7.

To show the effect of the multi-class SVM classification, six different floors were selected as shown in Figure 2.8. To test this system, the robot was steered on completely different trajectories and the testing dataset was recorded. Each training class contained 414 three dimension feature vectors. The training data on the soft blue rubber (8 mm thick) tiles were noisy due to the uneven colouring and gaps between tiles. Each test class had 396 feature vectors. The results from the MATLAB implementation are shown in the form of confusion matrix in Table 2.1.

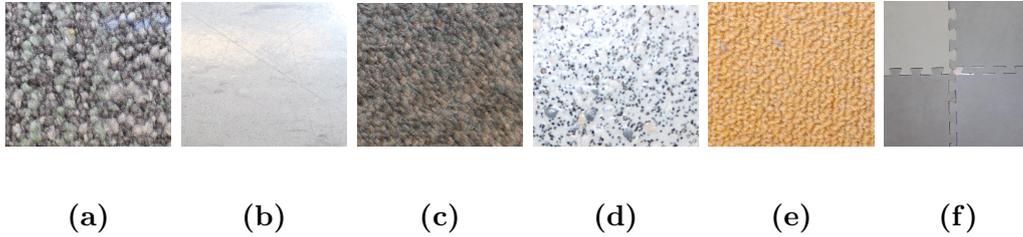


Figure 2.8: Surfaces where the training data were collected. (a) Carpet 1 (b) Lino (c) Carpet 2 (d) Marble (e) Carpet 3 (f) Blue Rubber. Reused from [1] ©2015 IEEE.

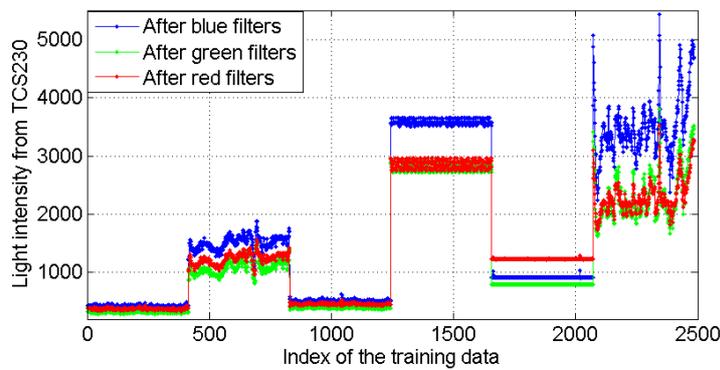


Figure 2.9: The training data of the SVM classifier. Each class has 414 feature vectors, which comprise the intensities of the red, green and blue components in the light from the surfaces (a-f), respectively, in Figure 2.8. Reused from [1] ©2015 IEEE.

From the results, we can see precision and recall are above 98% for all classes. Carpet 1 is wrongly classified as Carpet 2 six times, because these two floors were both dark carpets with little light reflection. This similarity is also clearly shown in Figure 2.9. Nevertheless, this validation test has shown a promising result for the proposed surface classification. A classification validation test is done in the ROS implementation with the same training and testing dataset as well. It also shows similar high precision and recall values.

Table 2.1: Confusion Matrix of the Validation Test. Reused from [1] ©2015 IEEE.

| Floors | (a) | (b) | (c) | (d) | (e) | (f) | Precision (%) |
|------------|-----|-----|------|-----|-----|-----|---------------|
| (a) | 390 | 0 | 6 | 0 | 0 | 0 | 98.5 |
| (b) | 0 | 396 | 0 | 0 | 0 | 0 | 100 |
| (c) | 0 | 0 | 396 | 0 | 0 | 0 | 100 |
| (d) | 0 | 0 | 0 | 395 | 0 | 1 | 99.7 |
| (e) | 0 | 0 | 0 | 0 | 396 | 0 | 100 |
| (f) | 0 | 0 | 0 | 0 | 0 | 396 | 100 |
| Recall (%) | 100 | 100 | 98.5 | 100 | 100 | 100 | |

(a) Carpet 1 (b) Lino (c) Carpet 2 (d) Marble (e) Carpet 3 (f) Blue Rubber.

2.5 Floor Classification with Motor Current

In Section 2.5.1, experiments were performed to show that the floor inclination, even on apparently flat indoor surfaces, was a significant factor in the motor current measurement. The investigation started by measuring the motor current with the robot propped up so that the wheels do not contact the floor. This eliminated possible effects of gearbox and bearing variations in motor current that were correlated with wheel position. Then the author examined the motor current on a path outward and then returning to the start. This showed variations in motor current that were opposite in sign for the same position between outward and returning paths. This suggests that the floor inclination variations are the cause of the current variations. Note that in all these experiments the robot was controlled to travel at constant speeds.

2.5.1 Motivating Experiments

An initial measurement was made with the robot wheels propped up without floor contact. The robot velocity was set at a constant speed of 0.05 m/s. A median

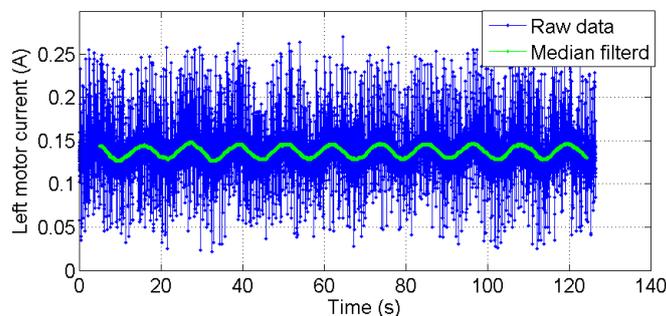


Figure 2.10: Left wheel motor current when the robot was running at a speed of 0.05 m/s in the air. Reused from [2] ©2016 IEEE.

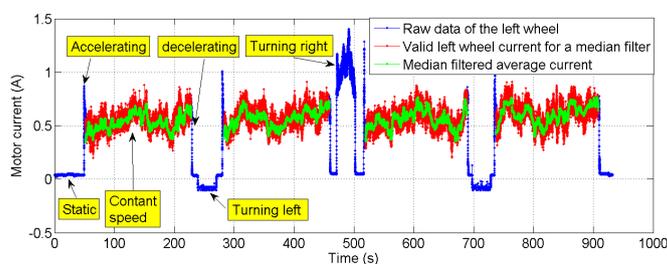


Figure 2.11: Motor current showing a repeat pattern and anti-symmetrical effect on a return straight path on Carpet 2. Reused from [2] ©2016 IEEE.

filter of 1 second window size was applied to the motor current. The result is shown in Figure 2.10. The periodic current variations can be explained by systematic variations in the friction of the gear box and wheel bearings since this has the same period as the wheel angles. This period is the wheel circumference divided by 0.05 m/s that is approximately 11.5 seconds.

In the next experiment, the robot performed a return trip on the same straight path twice on Carpet 2. Figure 2.11 shows the motor current in this test. Note that the current varies systematically with position along the path and with a matching negative variation in the return path. These anti-symmetrical variations in median current are much higher than those seen in the no load test of Figure 2.10 and hence cannot be attributed to gear box or bearing friction.

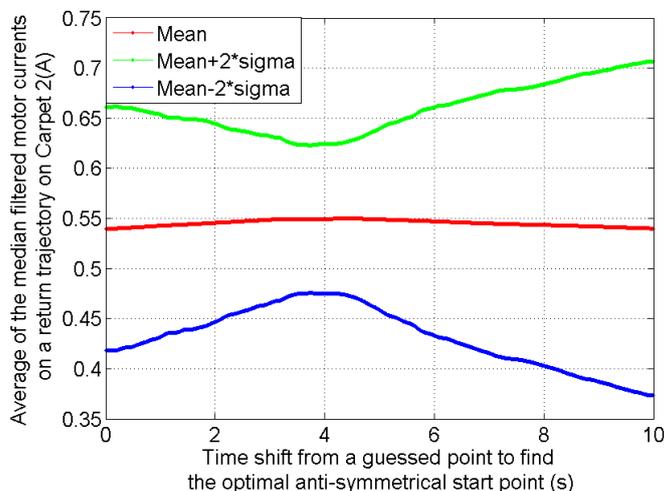


Figure 2.12: Searching for the optimal anti-symmetrical starting point for the first return path in Figure 2.11. Reused from [2] ©2016 IEEE.

The anti-symmetrical motor current on a return trip inspires the following data analysis. There should be an optimal starting point, P_{op} , in the second green segment in Figure 2.11, where the first green segment would be most anti-symmetrical that is current average between the forward and backward paths at the corresponding positions would show least variation. If we average the corresponding motor current in the first two green segments in Figure 2.11 starting from P_{op} , the statistical variance should be the smallest among all the possible starting point around P_{op} . Figure 2.12 shows the mean and two standard deviations of the current summed between corresponding points in the opposing direction paths against the offset in the designated start time. Note that the four seconds offset minimises the standard deviation and corresponds to aligning the end of the forward path with the start of the backward path.

The scatter plot between motor current on corresponding points on the paths is shown in Figure 2.13. The X axis is the forward path current and the Y axis is the backward path current for the same point on the robot path. The Pearson's product-moment correlation coefficient [47] between the two axes is -0.65, which represents a strong negative correlation. The small floor inclination variations are the only feasible cause. For example if the robot is travelling uphill in the

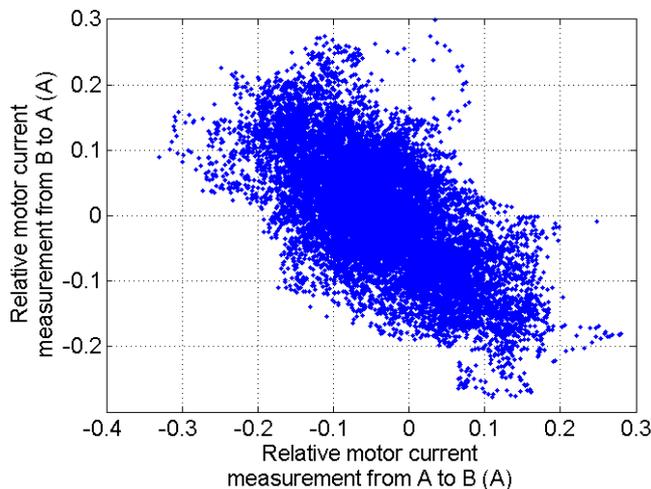


Figure 2.13: Effect of the strong negative correlation of the motor current on a return robot path between the forward and backward paths. Reused from [2] ©2016 IEEE.

forward path it will travel downhill in the backward path at the same point on the floor.

2.5.2 Floor Inclination Angle Compensation

First it is necessary to calibrate the offset in the accelerometer when the robot was travelling on a horizontal surface. A joystick was used to control the robot to follow a closed path several times at a constant speed. Since the trajectory of the robot started and finished at the same position, the mean robot inclination angle should be horizontal. Therefore the mean of the ADC output was the offset the system should be set.

An experiment was performed on Carpet 2 with two return trips. The motor current measured using the 10 bit ADC was $\theta_{0Carpet2} = 466.9$. Since the accelerometer voltage shift and amplification circuit is designed to measure the inclination angle range of $[-5^\circ, 5^\circ]$, the measured inclination angle in this experiment can be approximated by the following equation:

$$\alpha_{Carpet2}(t) = (\theta_{Carpet2}(t) - \theta_{0Carpet2}(t)) \times 10^\circ / 1024 \quad (2.2)$$

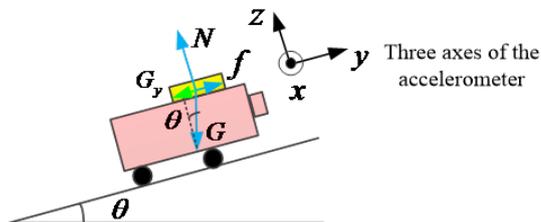


Figure 2.14: Schematic of the usage of the accelerometer for inclination angle detection. Reused from [2] ©2016 IEEE.

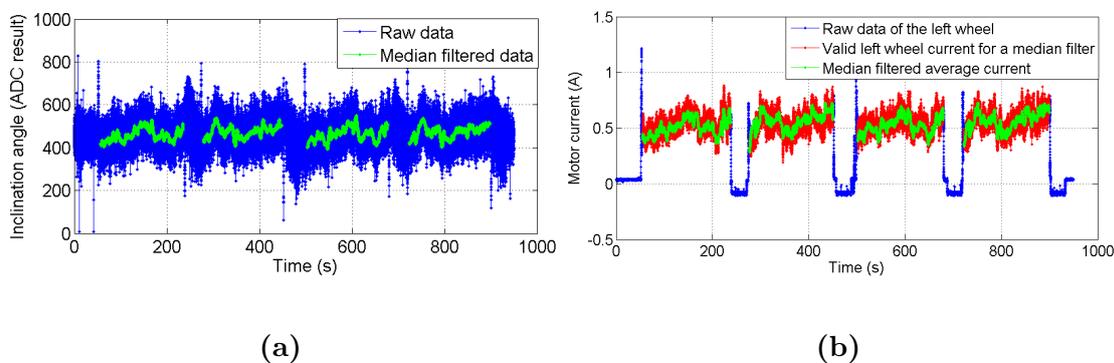


Figure 2.15: (a) Inclination angles measured in degrees on a return straight path on Carpet 2. (b) Motor current in the same experiment as (a). Reused from [2] ©2016 IEEE.

where $\alpha_{Carpet2}$ and $\theta_{Carpet2}(t)$ are the inclination angles measured in degrees and in ADC values respectively. $\theta_{Carpet2}(t)$ is shown in Figure 2.15 (a) and the corresponding motor current measurements are shown in Figure 2.15 (b). The strong correlation between inclination angle and the motor current provides further evidence for the motor current dependency on inclination angle. Pearson's product-moment correlation coefficients for current and inclination on four floors are shown in Figure 2.16.

Given a set of motor current and floor inclination measurements from the robot, the classifier aims to determine the floor type accurately and quickly. The training and testing datasets were collected separately on different robot paths on four floors shown in Figure 2.3. For the training step, a 10-fold cross validation was performed and the final accuracy was the cross-validation accuracy.

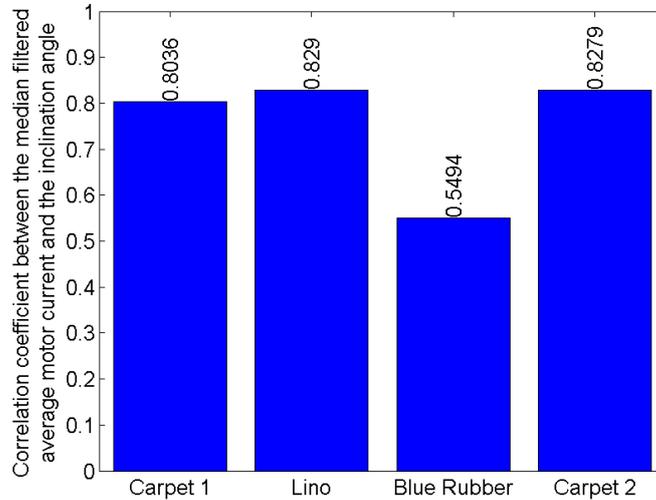


Figure 2.16: Pearsons product-moment correlation coefficients between the median filtered average motor current and the inclination angles on four floors. Reused from [2] ©2016 IEEE.

A set of features need to be extracted from the data and then be provided to the classifier to be trained and tested. These features should describe the corresponding motor current signals well enough to differentiate the currents on various floors. In [48], the authors report that the first five statistical moments (mean, variance, skewness, kurtosis, fifth moment) are effective and generalizable features for robot-terrain interactions. These moments were chosen as features in this work. The k^{th} statistical moment, μ_k^x , of a time-series of n observations of x is defined as below [29]:

$$\mu_k^x = \frac{1}{n} \sum_{i=1}^n x_i^k \quad (2.3)$$

The floor classification problem is divided into two main steps: training and testing. The training and testing data are recorded using the *rosbag* tool in ROS. Each feature was normalised to the range $[0, 1]$ before training and testing as suggested in [45] for the best performance of SVMs.

2.5. FLOOR CLASSIFICATION WITH MOTOR CURRENT

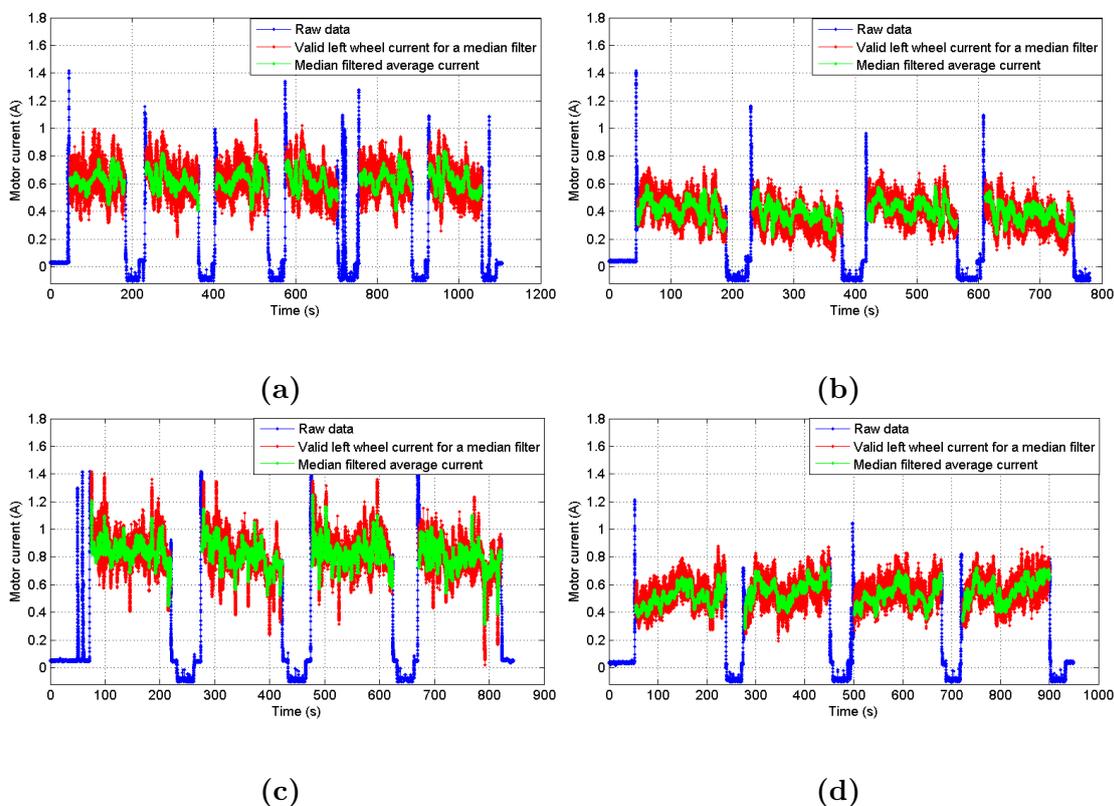


Figure 2.17: Motor current from which the training data were extracted. (a) Carpet 1 (b) Lino (c) Blue rubber (d) Carpet 2. Reused from [2] ©2016 IEEE.

2.5.3 Motor Current Compensation

The robot was controlled to run forward in a straight line at 0.05 m/s, followed by a stop, a 180 degree turn and another straight line back to the original point. Figure 2.17 depicts the motor current where the training data were extracted.

The motor current and the inclination angles in the training data are correlated on all four floors. Their linear relationships are fitted with the least square method as the red lines in Figure 2.18.

Let us denote the compensated motor current as $\bar{i}(t)$, the median filtered average motor current at time t as $i(t)$, the median filtered inclination angle measured by

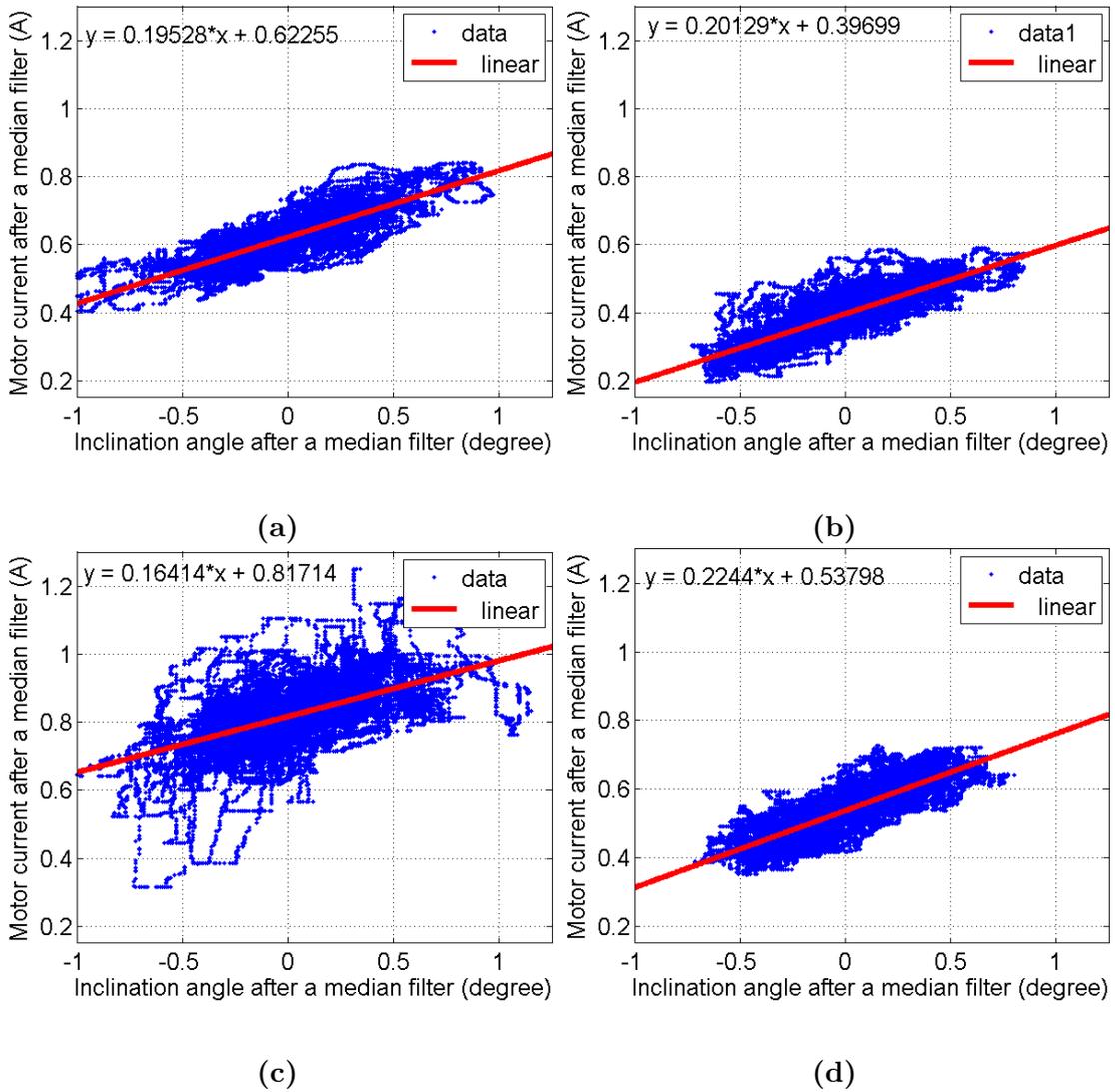


Figure 2.18: Linear relationships between the motor current and the accelerometer measured inclination angles when the robot was driving straight forward. (a) Carpet 1 (b) Lino (c) Blue Rubber (d) Carpet 2. Note that the slopes are approximately the same and consistent with expectation that the change in motor with inclination is independent of the surface. Reused from [2] ©2016 IEEE.

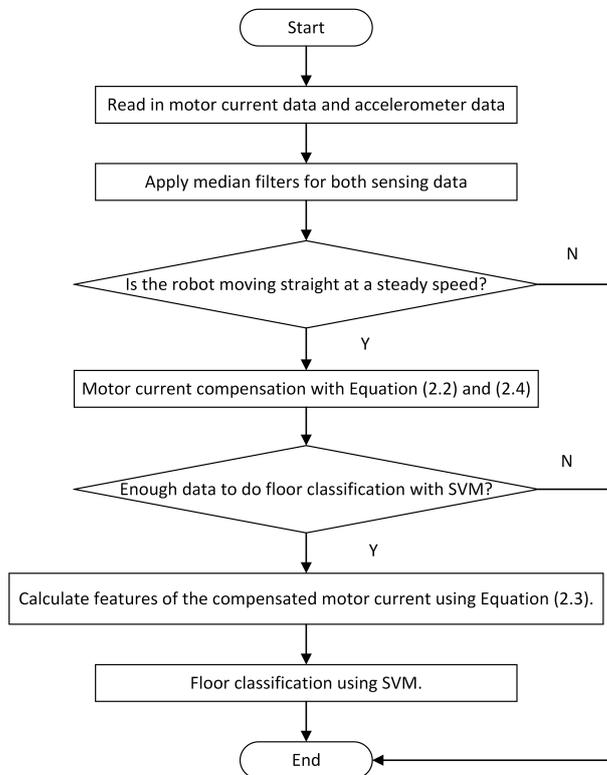


Figure 2.19: Flow chart of one loop for floor classification with compensated motor currents. Reused from [2] ©2016 IEEE.

the accelerometer at time t as $\alpha(t)$ and the slope representing the linear relationship between $i(t)$ and $\alpha(t)$ as k .

$$\bar{i}(t) = i(t) - k\alpha(t) \quad (2.4)$$

In the experiments the author chose $k = 0.207$, the mean of the four slopes in Figure 2.18, since the contribution to the motor current due to the inclination is not expected to depend on the floor surface. Note that Equation (2.4) provides direct compensation for motor current using the accelerometer input without needing to calibrate for the actual inclination angle. Now five features for the floor classification system are listed as $\mu_1^{\bar{i}(t)}$, $\mu_2^{\bar{i}(t)}$, $\mu_3^{\bar{i}(t)}$, $\mu_4^{\bar{i}(t)}$, $\mu_5^{\bar{i}(t)}$. Figure 2.19 is a flow chart to explain one classification loop in the algorithm. The compensated motor current for both training and testing data is shown in Figure 2.20.

2.5. FLOOR CLASSIFICATION WITH MOTOR CURRENT

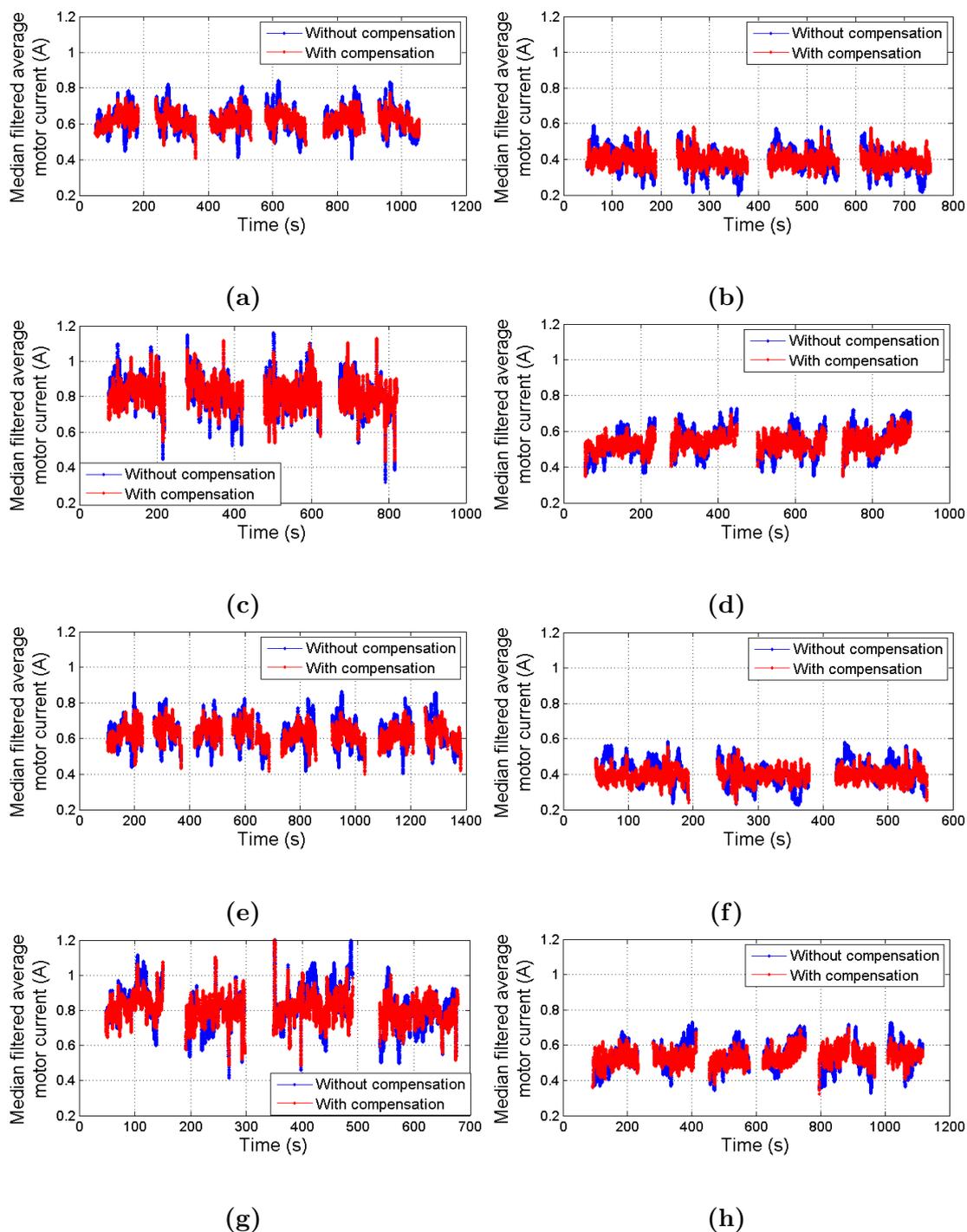


Figure 2.20: Median filtered average motor current. (a) Training data on Carpet 1 (b) Training data on Lino (c) Training data on Blue Rubber (d) Training data on Carpet 2 (e) Testing data on Carpet 1 (f) Testing data on Lino (g) Testing data on Blue Rubber (h) Testing data on Carpet 2. Reused from [2] ©2016 IEEE.

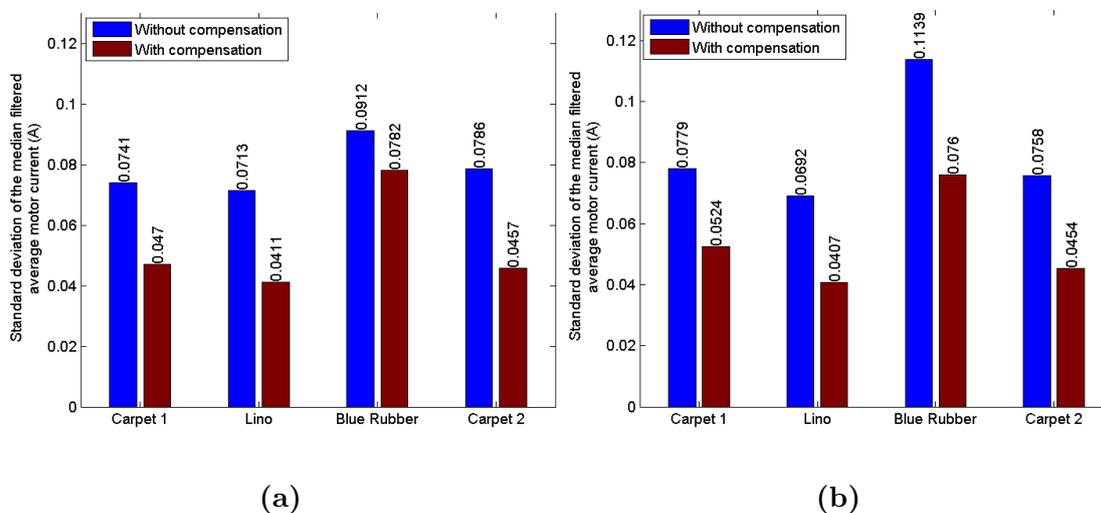


Figure 2.21: Standard deviation of the median filtered average motor current used for floor classification. (a) Training data (b) Testing data. Reused from [2] ©2016 IEEE.

Figure 2.21 shows the reduction of the standard deviation of the detected motor current due to inclination compensation for this experiment.

2.5.4 Experiments and Results

Classification of Four Floors

As pointed out in [48], there is a trade-off between data segment length and classification accuracy. The author varied the data segment length measured in the travelling distance from 0.025 m to 3 m. For every dataset length, a grid search for the best feature combinations among all the combinations of the five statistical moments was performed. In this case, the total number of feature combinations is calculated as below.

$$f_{num} = \binom{5}{1} + \binom{5}{2} + \binom{5}{3} + \binom{5}{4} + \binom{5}{5} \quad (2.5)$$

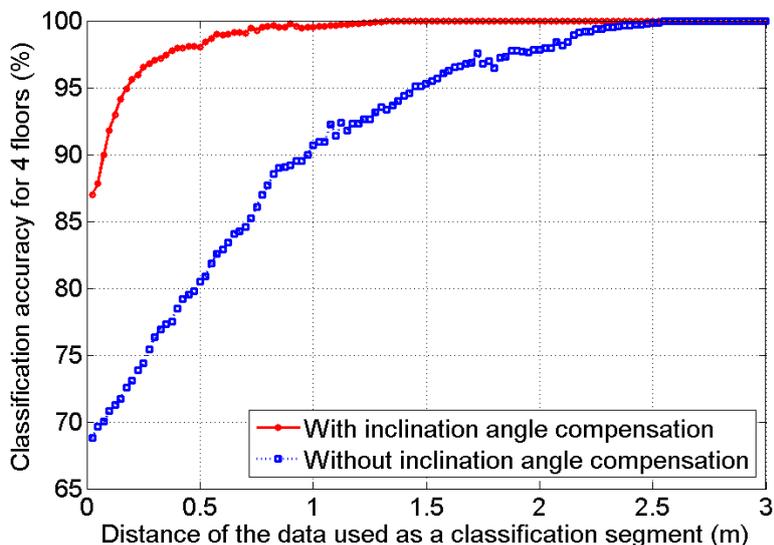


Figure 2.22: Effect of data segment length on classification accuracy. Reused from [2] ©2016 IEEE.

Eventually the best classification accuracy for that length of data segment among all the feature combinations was selected. It is clear that compensating the motor current with the corresponding inclination detection is analogously successful as shown in Figure 2.22.

Compensating the motor current with inclination angles brought significant improvement in the classification accuracy for every data length except those longer than 2.5 m. With a travelling distance shorter than 2.5 m, the classification accuracy with inclination angle compensation is higher than that without compensation. For lengths longer than 2.5 m both cases have an accuracy of 100%. This can be attributed to the average floor inclination angle being close to zero over a 2.5 m distance, allowing the classifier to differentiate based on floor type alone. Although the two carpets were similar and challenging to differentiate, the system still performed well. Among these different data lengths, the author chooses a data length of 0.2 m to show the means and variances of the compensated motor current for the training and testing datasets in Figure 2.23. Among

all the 31 possible combinations of the features, the set of mean, skewness, kurtosis and fifth moment generates the highest classification accuracy of 95.40%. The confusion matrix is shown in Table 2.2.

Table 2.2: Confusion Matrix of Floor Classification on Compensated Motor Current with a Median Filter on a Data Length of 0.2 m. Reused from [2] ©2016 IEEE.

| Floors | (a) | (b) | (c) | (d) | Precision (%) |
|------------|-------|-------|------|-------|---------------|
| (a) | 762 | 0 | 11 | 69 | 90.5 |
| (b) | 0 | 812 | 0 | 3 | 99.63 |
| (c) | 6 | 0 | 908 | 0 | 99.34 |
| (d) | 58 | 11 | 0 | 802 | 92.08 |
| Recall (%) | 92.25 | 98.66 | 98.8 | 91.76 | |

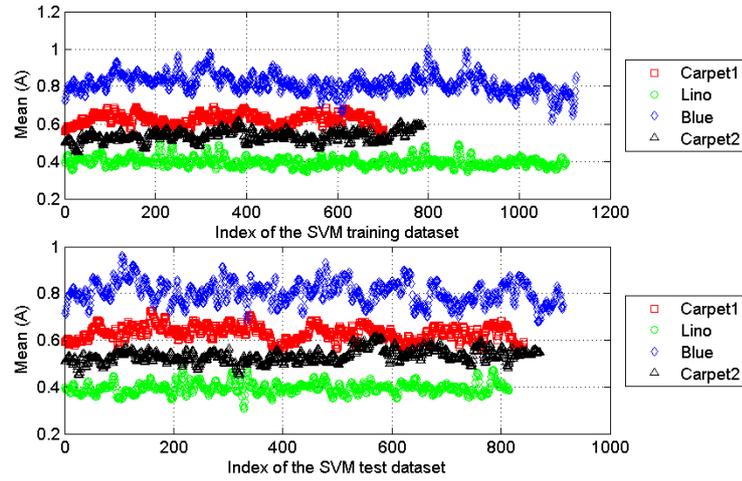
(a) Carpet 1 (b) Lino (c) Blue Rubber (d) Carpet 2.

Classification of Two Floors with the Same Colour and Pattern

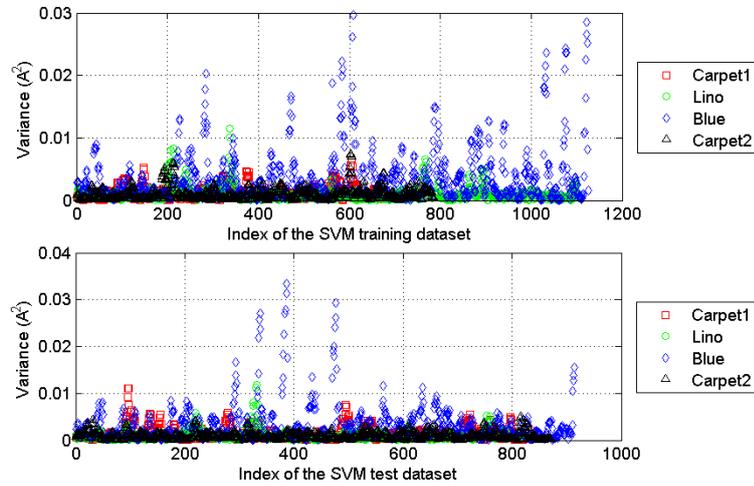
This experiment shows that floor classification system with motor current sensors can overcome the disadvantage of the previous floor classification system based on a colour sensor alone [1].

The robot was manually controlled to collect some training and testing data for floor classification on two floors, a bed sheet on Lino and a bed sheet on Carpet 1 as shown in Figure 2.24. For a data length of 0.05 m, the system reached a classification accuracy of 99%. For a data length of 0.15 m, the accuracy was 100%. The mean values of the training and testing datasets for a data length of 0.15 m are plotted in Figure 2.25. However, the classification using only the colour sensor data gave an accuracy of around 70%.

2.5. FLOOR CLASSIFICATION WITH MOTOR CURRENT



(a)



(b)

Figure 2.23: Features for floor classification. (a) Mean of the motor current (b) Variance of the motor current. Reused from [2] ©2016 IEEE.



Figure 2.24: Floor classification experiment with the same bed sheet. (a) Bed sheet on Lino (b) Bed sheet on Carpet 1. Reused from [2] ©2016 IEEE.

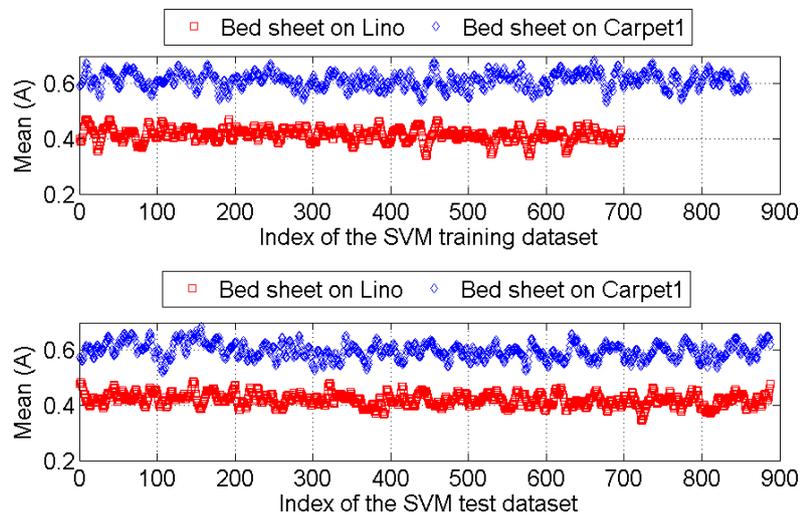


Figure 2.25: Mean motor current values of the training and testing data for floor classification on a bed sheet. Reused from [2] ©2016 IEEE.

2.6 Summary and Conclusions

This chapter has presented two floor surface classification systems with a colour intensity sensor and motor current sensor. The systems can provide immediate notification of floor changes to the robot, allowing faster calibration adaptation in the real time. The SVM algorithms implemented in MATLAB and ROS use data collected from the floor colour sensor and successfully classify six different floor types with precision and recall above 98%. The second classification approach is based on mobile robot motor current with an accelerometer to compensate for small inclination changes. The experiments show significant improvements in classification accuracy when accelerometer data are included. The floor classification accuracy compares well with rough terrain systems [28, 30, 48]. The system can detect a transition to a new floor region almost immediately, which is beneficial not only for the calibration of odometry and sensor extrinsic parameters but also for a mapping process because the system actively adapts the odometry parameters to estimate the robot pose more accurately. At the same time, there are some drawbacks of the proposed floor classifications. It cannot predict the floor type where the robot will move on in a short while, which is doable using a vision floor classification system. The motor current sensing system may not differentiate two flat surfaces with very similar friction coefficients. However, such floors may not affect odometry parameters very differently. Despite these drawbacks, the developed floor classification system is still useful and accurate in most scenarios. Despite the drawbacks, these floor classification methods can be integrated into indoor service robots or rescue robots easily, which can help them to locate more accurately on various floors to improve their working efficiency. For example, a floor cleaning mobile robot can adapt its odometry parameters to fit different floor conditions in a house. By doing that, the robot can save its battery and clean wherever it has not been to in a map accurately. This is essential for such a robot, because robot owners wish the robot to clean every corner of their house more quickly and cover every piece of floors. The calibration of odometry and sensors will be described in the next chapter.

Calibration of Odometry and Sensor Extrinsic Parameters on Different Floors

3.1 Introduction

One of the fundamental problems in mobile robotics is simultaneous localisation and mapping (SLAM), where one or more robots build a map and localise within the map at the same time. It is important to calibrate mobile robot odometry if good localisation and mapping accuracy are required.

Many researchers have contributed to this area. One of the most popular odometry calibration method is UMBmark [21]. This method experimentally calibrates the difference between integrated robot positions and positions in a reference frame by running the robot several times along a square trajectory clockwise and counter-clockwise. This may not be applicable in relatively small spaces. Larsen et al. [49] took advantage of the online estimation property of a Kalman filter and introduce an augmented Kalman filter (AKF) algorithm to fuse the wheel encoder measurements and the external LRF measurements to simultaneously estimate the robot pose and its odometry parameters. More recently, Martinelli et al. [50] introduced an Observable filter (OF) and combined it with an AKF to estimate the systematic and non-systematic errors for both synchronous and differential drive robots. These methods all fall into the online odometry calibration category, which is beneficial, particularly when the odometry parameters change slowly due to tyre pressure, wear, robot load and surface type as pointed out by Maye [51].

When a sensor is mounted on a robot, the position and orientation of the sensor in robot coordinates are examples of extrinsic parameters. Levinson and Thrun [52] addressed extrinsic parameter calibration by defining an energy function on point clouds that penalised points far from the surfaces constructed with other measurements of a 3D laser range finder. They applied a grid search to optimise this cost function. Censi [22] developed closed form offline maximum likelihood based odometry calibration and simultaneously calibrated odometry parameters and the sensor parameters of mobile robots. The algorithms can produce accurate calibration on a uniform surface. Grid search methods are suitable for low dimensional problems, but are inefficient for optimization on many dimensions, such as 12 dimensions in this research. Similar to the approach developed, Sheehan et al. [53] used the Rényi Quadratic Entropy as a cost function to measure the degree of organisation of a point cloud, which is linked to the crispness of the point cloud of a 3D laser scanner. Their aim is to optimise extrinsic parameters. However, the above research does not calibrate the robot odometry parameters simultaneously. This is important since higher odometry estimation accuracy can help improve sensor pose estimation on a robot.

The system developed here by the author allows differences in the odometry calibration on different floor surfaces. In 2013, Maye et al. [51] first developed an online algorithm to calibrate the robot odometry and sensor extrinsic parameters, which detected and locked unobservable parameters in the parameter space and only updated the parameters in the observable directions. Their contribution is to use an information theoretic measure to automatically identify and store incoming sensing data to reduce the computation load by discarding redundant information. They also explore and take advantage of the observability of the calibration parameters and calibrated parameters as they become observable, keeping other unobservable parameters to be the initial guess. However, this method works for a robot with a single sensor observing distinct landmarks and known data associations at each time step. The method developed in this chapter can work with multiple sensors and does not require known data associations, that may not be robust in practice.

The dependency of odometry calibration on floor surface has not been investigated by previous research papers, although it was mentioned in [5, 23, 24]. A limited number of publications have focused on the dependency of terrain types and robot odometry parameters. Reinstein et al. [54] improved the root-mean-square error (RMSE) in position by 68% with respect to a state of the art odometry model by classifying different terrains for outdoor skid-steer robots. Ray [55] analysed the net traction and resistive wheel torques for a robot on rigid or deformable terrains, which directly affected the robot odometry parameters. However her work does not include floor classification and requires a motor torque sensor.

This chapter describes two different odometry calibration methods. The first one is based on an EKF. The other one uses PSO to find the optimal sensor extrinsic parameters and odometry parameters simultaneously. The novelty of the proposed methods in this chapter are calibrating these parameters on multiple floors classified using the systems described in Chapter 2.

3.2 Conventional Differential Drive Robot Odometry Model

The widely used differential drive wheeled robot kinematic model derived by Chong and Kleeman [19] is deployed here with the robot pose at time step k :

$$\Delta L_l(i) = \omega_l(i)R_l\Delta t(i) \quad (3.1)$$

$$\Delta L_r(i) = \omega_r(i)R_r\Delta t(i) \quad (3.2)$$

$$\Delta L(i) = [\Delta L_r(i) + \Delta L_l(i)] / 2 \quad (3.3)$$

$$\Delta\theta(i) = [\Delta L_r(i) - \Delta L_l(i)] / B \quad (3.4)$$

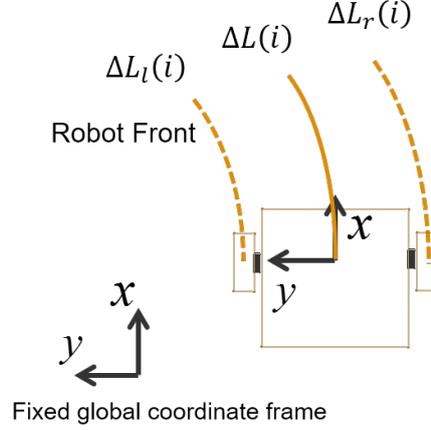


Figure 3.1: Coordinate frame of conventional mobile robot odometry system in Equation (3.3-3.5).

$$\begin{bmatrix} \theta(k) \\ x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} \theta(0) + \sum_{i=1}^k \Delta\theta(i) \\ x(0) + \sum_{i=1}^k \Delta L(i) \cos[\theta(i-1) + \Delta\theta(i)/2] \\ y(0) + \sum_{i=1}^k \Delta L(i) \sin[\theta(i-1) + \Delta\theta(i)/2] \end{bmatrix} \quad (3.5)$$

The definition of odometry coordinate system is in Figure 3.1. The odometry parameters consist of R_l , R_r and B . R_l and R_r are the radii of left and right wheels respectively. B is the separation distance between the two driven wheels. The reciprocal of B is denoted as l . $\omega_r(i)$ and $\omega_l(i)$ are right and left wheel angular velocities over the time interval $\Delta t(i)$. $\Delta L_l(i)$, $\Delta L_r(i)$ and $\Delta L(i)$ are the distances travelled of the left wheel, right wheel and the robot respectively. All variables with a $\hat{\cdot}$ on top of them represent that they are estimated values of the corresponding variables without the $\hat{\cdot}$ symbol. If R_l , R_r and B are known, an accurate estimate of the robot pose can be obtained. However, small errors in the estimated values, \hat{R}_l , \hat{R}_r , \hat{B} and wheel slippage can be accumulated to significant errors in the estimation of the robot pose over time. This odometry model is further improved in Chapter 4 and Chapter 5.

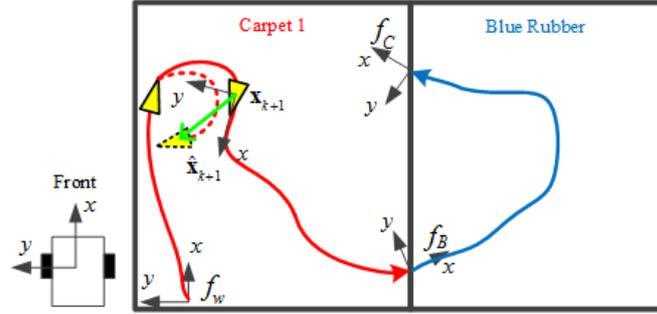


Figure 3.2: The sketch map of the continuous differential drive mobile robot odometry calibration on Carpet 1 and Blue Rubber, on which the robot trajectories are marked with red and blue lines, respectively. Reused from [1] ©2015 IEEE.

3.3 Real-time Odometry Calibration on Multiple Floors

3.3.1 Calibration on One Floor with EKF

A Kalman filter is an optimal estimator for a linear system model with additive independent white noise in both the transition and the measurement systems. It is recursive so that new measurements can be processed as they arrive [56]. However, since the robot motion is not a linear function of the odometry parameters as evident in Equation (3.5), the EKF is employed to linearise the measurement function. In Figure 3.2, suppose the robot is at t_k with a state vector of three odometry parameters, $\mathbf{x}(k) = [R_l(k), R_r(k), l(k)]^T$, where $l(k) = 1/B(k)$. It starts to move in the solid red trajectory until t_{k+1} . This movement can be measured with an exteroceptive sensor, such as a camera or a LRF. Proprioceptive sensors, such as wheel encoders, are used to predict this pose change based on the odometry parameters (i.e. the EKF state $x(k)$). In this case, robot pose change estimation with scan matching is chosen as the *laser pose change measurement*. The pose change based on wheel encoders with estimated odometry parameters forms the *odometry pose change prediction*. The difference between the laser and odometry pose change forms the innovation of the EKF. Then state estimate (i.e. the odometry parameters) is updated using the EKF update equations.

Following the notation in [56, 57], the state transition function is $\mathbf{x}(k+1) = f(k, \mathbf{x}(k)) + \mathbf{v}(k)$, where $\mathbf{v}(k)$ is the additive white noise to the state vector. f and h in Equation 3.7 are symbols of a function with the first parameter to be the time step and the second parameter to be the independent variable. If the robot runs on the *same* surface, it is assumed that the odometry parameters do not vary during the online calibration process. Therefore, the error covariance matrix of $\mathbf{v}(k)$ denoted by $\mathbf{Q}(k)$ is $\mathbf{0}_{3 \times 3}$. According to this assumption, $\mathbf{x}(k+1) = \mathbf{x}(k)$ and the Jacobian matrix:

$$\mathbf{F}(k) = \left. \frac{\partial f(k)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} \quad (3.6)$$

where $\hat{\mathbf{x}}(k|k)$ represents estimate of $\mathbf{x}(k)$ given measurements $\mathbf{z}(k), \mathbf{z}(k-1), \dots$. The measurement vector of the EKF is the laser pose change measurement. The state $\mathbf{x}(k)$ is used to predict this via the odometry update Equations (3.1-3.5) that depend on $\mathbf{x}(k)$. The measurement prediction is

$$\hat{\mathbf{z}}(k+1|k) = h(k+1, \hat{\mathbf{x}}(k+1|k)) = \begin{bmatrix} \Delta \hat{\theta}(k+1) \\ \Delta \hat{x}(k+1) \\ \Delta \hat{y}(k+1) \end{bmatrix} = \begin{bmatrix} \hat{\theta}(k+1) - \hat{\theta}(k) \\ \hat{x}(k+1) - \hat{x}(k) \\ \hat{y}(k+1) - \hat{y}(k) \end{bmatrix} \quad (3.7)$$

where $\hat{\mathbf{x}}(k+1|k) = [\hat{R}_l, \hat{R}_r, \hat{l}]^T$ is the estimated state. Note that Equation (3.7) uses the wheel angular velocity vectors $[\omega_l(i), \omega_r(i)]^T$ that lie between t_k and t_{k+1} .

The relative pose change between two time steps is measured using the correlative scan matching algorithm proposed by Olson [58]. The work by Olson [58] is instrumental for the operation of the EKF since it accurately estimates the scan match error covariance based on the actual laser scan data. For example a scan match in a long corridor will correctly estimate a large error in the direction of the corridor. The measurement function is

$$\mathbf{z}(k+1) = \mathbf{h}_{scan}(k+1) + \mathbf{w}(k+1) \quad (3.8)$$

where $\mathbf{h}_{scan}(k+1)$ is the mean value of the pose change vector from scan matching and $\mathbf{w}(k+1)$ is its corresponding white noise, whose covariance matrix is $\mathbf{R}(k+1)$. The measurement residual is the green arrow shown in Figure 3.2 with the formula as

$$\mathbf{v}(k+1) = \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1|k) \quad (3.9)$$

The state covariance at t_k is denoted as $\mathbf{P}(k|k)$. To translate the odometry parameters estimation errors into the robot pose change measurement errors, it is necessary to calculate the measurement Jacobian as following:

$$\mathbf{H}(k) = \left. \frac{\partial \mathbf{h}(k+1)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k)} = \begin{bmatrix} \frac{\partial \Delta \hat{\theta}(k+1)}{\partial \hat{R}_l(k+1)} & \frac{\partial \Delta \hat{\theta}(k+1)}{\partial \hat{R}_r(k+1)} & \frac{\partial \Delta \hat{\theta}(k+1)}{\partial \hat{l}(k+1)} \\ \frac{\partial \Delta \hat{x}(k+1)}{\partial \hat{R}_l(k+1)} & \frac{\partial \Delta \hat{x}(k+1)}{\partial \hat{R}_r(k+1)} & \frac{\partial \Delta \hat{x}(k+1)}{\partial \hat{l}(k+1)} \\ \frac{\partial \Delta \hat{y}(k+1)}{\partial \hat{R}_l(k+1)} & \frac{\partial \Delta \hat{y}(k+1)}{\partial \hat{R}_r(k+1)} & \frac{\partial \Delta \hat{y}(k+1)}{\partial \hat{l}(k+1)} \end{bmatrix} \quad (3.10)$$

Since the state is assumed to be constant for a calibration process on the *same* surface, the state prediction matrix is

$$\mathbf{P}(k+1|k) = \mathbf{P}(k|k) \quad (3.11)$$

Residual covariance matrix is

$$\mathbf{S}(k+1) = \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}(k+1)^T + \mathbf{R}(k+1) \quad (3.12)$$

The filter gain is

$$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}(k+1)^T\mathbf{S}(k+1)^{-1} \quad (3.13)$$

Then an updated estimate of the state vector based on the latest laser pose change measurement and the odometry (wheel encoders) is calculated as below.

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\mathbf{v}(k+1) \quad (3.14)$$

The corresponding confidence for the above estimate is expressed in a covariance matrix as following.

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}(k+1)^T \quad (3.15)$$

For a good reference of the equations in Subsection 3.3.1, the author suggests to read the reference article [56] written by his supervisor, A/Prof. Kleeman. Note that $\mathbf{H}(k)$ in Equation (3.10) may be rank deficient for some robot motions [51, 59]. For example, a constant curvature path results in a Jacobian rank of 2 and hence its inverse is not defined. This can be seen by assuming $\omega_r(i) = \omega_0$ and $\omega_l(i) = k\omega_0$ for a robot path, where k is a constant. One can apply the odometry Equations (3.1-3.5) to show that the final pose depends only on two parameters: $(R_r + kR_l)$ and $(R_r - kR_l)/B$. That means there are only 2 degrees of freedom in the motion, so the Jacobian has a rank of 2. This is derived in Appendix A.

In Equation (3.12), the measurement covariance \mathbf{R} has full rank in practice, ensuring \mathbf{S} has full rank, so Equation (3.13) is well defined. Moreover, in Equation (3.13) the null space of \mathbf{H} effectively removes the effects of measurements on any unobservable subspace of the state via the Kalman gain null space. For example, a straight path will not update the wheel separation distance B nor the corresponding component of \mathbf{P} in the EKF. This effect has been observed to avoid ill-conditioned updates to the state.

3.3.2 Calibration on Multiple Floor Types

Equations (3.1),(3.2) and (3.4) assume R_l , R_r and B to be time invariant, but this assumption is *not* true when a robot runs on multiple floors [1, 3] or with accelerations [4] in one calibration process. The proposed strategy for odometry calibration on *multiple* floor types is explained in Figure 3.3. The robot monitors the result of the SVM floor classification. When there are four consecutive surfaces changing classification results from the SVM classifier, the system declares a different floor region has been entered. The system will then store the

3.3. REAL-TIME ODOMETRY CALIBRATION ON MULTIPLE FLOORS

current estimated state vector $\mathbf{x}_{terrain(i)}(t_j)$ and its corresponding covariance matrix $\mathbf{P}_{terrain(i)}(t_j)$. If the current surface has not been detected previously, a new odometry calibration is created with an initialised EKF. If the robot detects the floor colour where the odometry parameters have been calibrated previously, it will continue the last suspended calibration using the stored calibration state and state covariance.

3.3. REAL-TIME ODOMETRY CALIBRATION ON MULTIPLE FLOORS

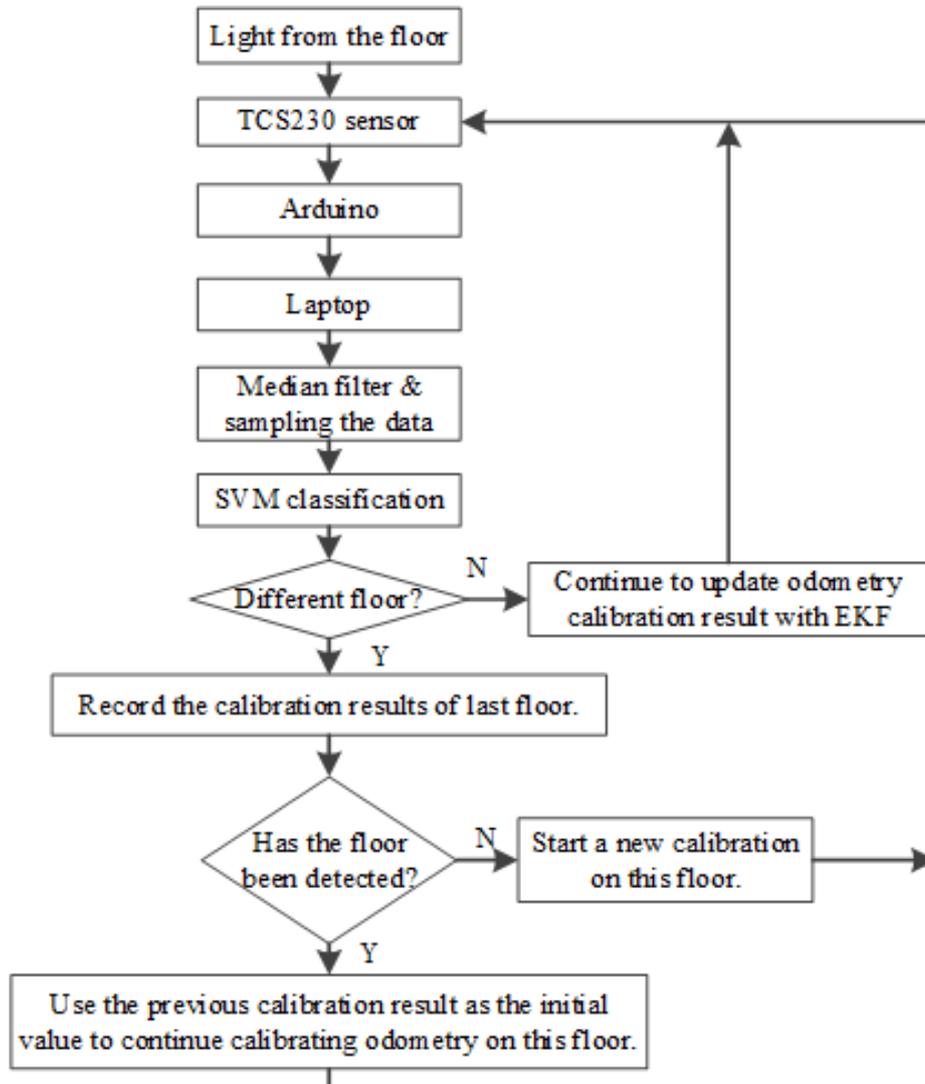


Figure 3.3: System structure of the EKF based odometry calibration with floor surface classification using SVM. Reused from [1] ©2015 IEEE.

3.3.3 Experiments and Results

Simulation on Three Floors

The author simulated a differential drive mobile robot with realistic parameters as those in the real world experiment. The wheel angular velocities were randomly generated within the range $[0 \text{ rad/s}, 1.047 \text{ rad/s}]$. The robot velocity was about 0.1 m/s at maximum. This is the same maximum speed as in the real experiment below. It is a reasonable speed for indoor service robots, which should not be operated at very high speed for safety concerns. Figure 3.4 depicts one of the trajectories. To simulate the scan matching measurement in the real experiments, Gaussian white noise was added to the ground truth pose change. The standard deviation vector for this simulated scan matching measurement was set to $[0.024 \text{ rad}, 0.018 \text{ m}, 0.0075 \text{ m}]^T$ for the heading, x and y dimensions, respectively. These were from the real environment test scan matching standard deviations in the real world experiment. The initial odometry calibration state was set to manufacturer values $\hat{\mathbf{x}}(0) = [0.0955 \text{ (m)}, 0.0955 \text{ (m)}, 2.778 \text{ (m}^{-1})]^T$. The initial state covariance was set to about 20% of the corresponding state values $\mathbf{P}(0|0) = \text{diag}((0.02 \text{ m})^2, (0.02 \text{ m})^2, (0.56 \text{ m}^{-1})^2)$. The ground truth odometry parameters on the three surfaces are shown in Table 3.1. Figure 3.5 depicts calibration results in one simulated odometry calibration using the proposed method on three floors.

10,000 repeated tests were performed in order to justify that the designed calibration method worked well in terms of estimation accuracy. Table 3.2 is an overview of the results from these repeated tests. More than 99% of the simulations have a less than 0.5% relative error of \hat{R}_l and \hat{R}_r on all floors, where relative error is defined as the difference from true value divided by the true value. About 90% of the tests have 0.5% relative error of \hat{l} , which is due to limited orientation changes of the robot pose.

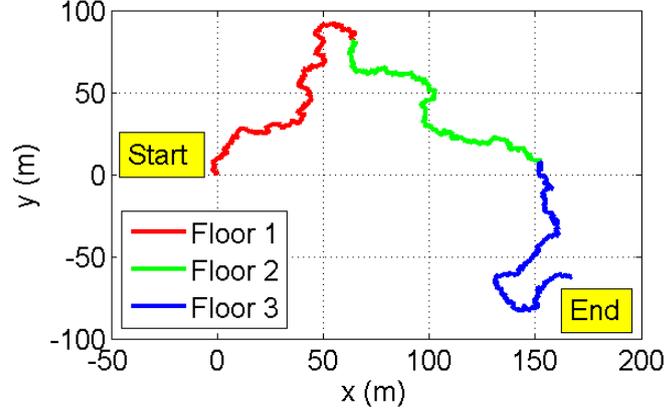


Figure 3.4: Robot trajectory in a simulated calibration with the proposed method. The red, green and blue lines show the robot trajectories on different surfaces. Reused from [1] ©2015 IEEE.

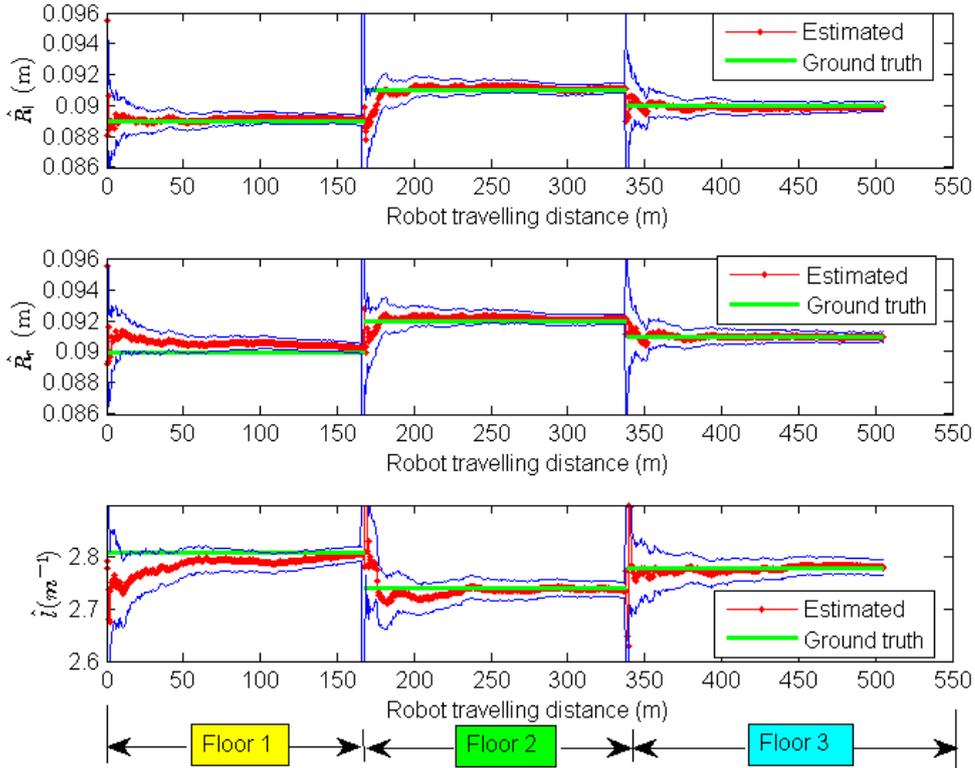


Figure 3.5: Simulated calibration results of \hat{R}_l , \hat{R}_r and \hat{l} on three floors. The blue curves show two standard deviation boundary around the estimated variables. The robot travelled on Floor 1, Floor 2 and Floor 3 successively. Reused from [1] ©2015 IEEE.

Table 3.1: Ground Truth Odometry Parameters in Simulation. Reused from [1] ©2015 IEEE.

| | Floor1 | Floor2 | Floor3 |
|-----------------|--------|--------|--------|
| $R_l(\text{m})$ | 0.089 | 0.091 | 0.09 |
| $R_r(\text{m})$ | 0.09 | 0.092 | 0.091 |
| $B(\text{m})$ | 0.356 | 0.365 | 0.36 |

Table 3.2: Numbers With Relative Errors Less Than 0.5% in 10,000 Simulation Odometry Calibration Tests. Reused from [1] ©2015 IEEE.

| | Floor1 | Floor2 | Floor3 |
|-------------|--------|--------|--------|
| \hat{R}_l | 9924 | 9942 | 9932 |
| \hat{R}_r | 9937 | 9940 | 9937 |
| \hat{l} | 9005 | 8953 | 9027 |

Figure 3.6 is the histogram with a Gaussian fit curve of the relative errors of \hat{R}_l on Floor 1. The Gaussian fit has a negligible mean of -1.4110^{-5} and a standard deviation of 0.00184 in relative errors meaning approximately 95% of the estimations of R_l have a relative error in the range $[-0.368\%, 0.368\%]$. The other two parameters have similar results.

Real-world Experiments on Two Floors

The extrinsic parameters of the two on-board LRFs were previously calibrated [40]. The measurements from the two lasers were merged into one virtual laser located in the middle between the two wheels [40]. The robot was steered in a winding trajectory in a lab environment. The sketch map of the trajectory was shown in Figure 3.2. The robot started odometry calibration on Carpet 1. After about 8 minutes, the robot was driven to the soft blue rubber floor and started a new calibration there. This movement was to test the recalibration

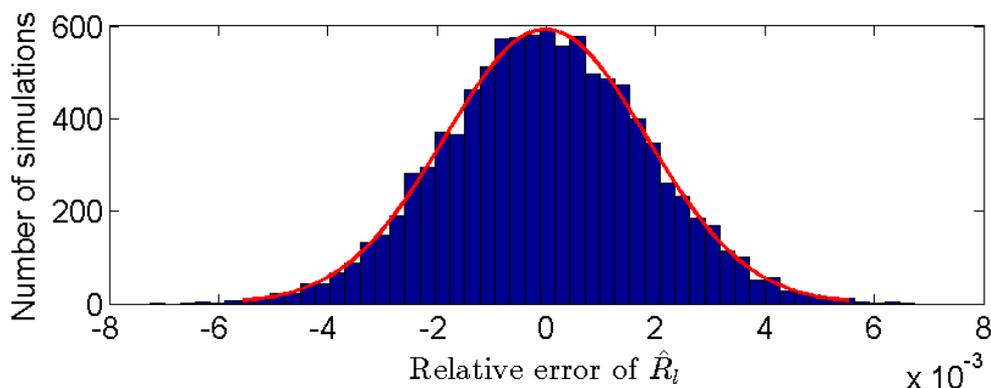


Figure 3.6: Histogram and Gaussian fit of the relative errors of \hat{R}_l on Floor 1 after 10,000 repeated simulated odometry calibrations. Reused from [1] ©2015 IEEE.

ability when the robot returned to Carpet 1 after one hour’s calibration on the blue rubber floor. The maximum speed of the robot was set to 0.1 m/s to reduce non-systematic odometry errors (e.g. slippage) and scan matching was performed whenever the robot pose changed by 0.5 m or 20°. $\hat{x}(0)$ and $\mathbf{P}(0|0)$ are the same as those in the simulation experiments.

The evolution of the standard deviations of the odometry parameters during a calibration on Carpet 1 and Blue Rubber is shown in Figure 3.7. Figure 3.8 depicts the evolutions of the odometry parameters. It shows the proposed odometry calibration method can start a new odometry calibration as soon as the floor classification system detects a change. Table 3.3 lists the calibration results with and without floor classifications. These small differences (about 1.5 mm in wheel radii) affect the map quality as shown below.

Odometry Correction Cost Comparison

State-of-the-art algorithms and metrics have been applied to show a convincing benefit from the proposed calibration strategy on multiple surfaces. These algorithms are listed below and the author refers readers to the corresponding references for more details:

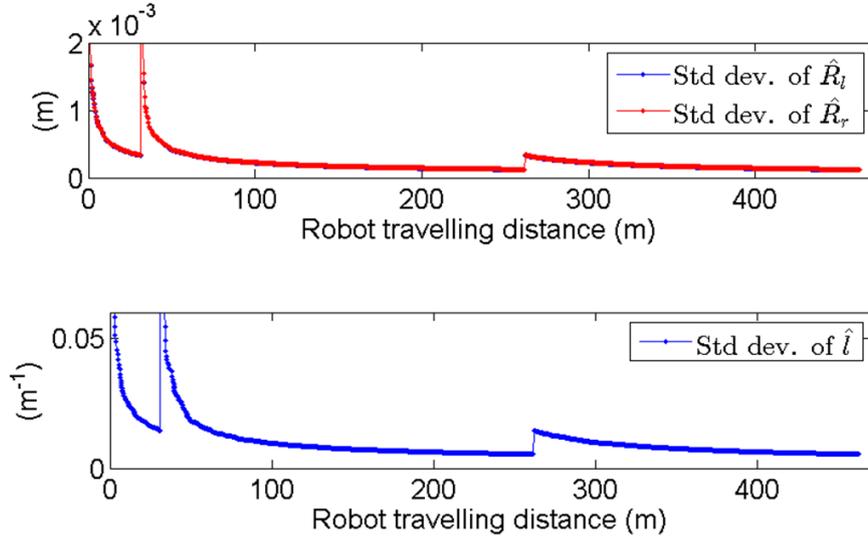


Figure 3.7: Evolutions of the standard deviations of odometry parameters during the robot experiment. Reused from [1] ©2015 IEEE.

1. Sparse Pose Adjustment (SPA) [26]: An efficient method to construct and solve the optimization problems in a graph based SLAM problem. The proposed system in this chapter constructs the graph with an initial estimate of the robot pose from the robot odometry and correct them based on the scan matching estimation.
2. Odometry Correction Cost Metric [16]: A framework for analysing the results of SLAM approaches based on a metric using only relative relations between poses to measure the square error of the corrected trajectory.

Figure 3.9 is the reference graph map built after a loop closure detection and optimisation of the graph with SPA. This optimised graph is a very close estimate of the ground truth, which we can see from the clear grid map built from the graph. The purple dots representing the poses of the robot are treated as the reference poses when applying the metric in [16]. The different colours of the trajectory on different floors are semantic labels. These trajectories were calculated using only the odometry pose of Equations (3.1-3.5) based on the calibration results (with floor classifications) in Table 3.3. It is clear that the robot trajectory almost

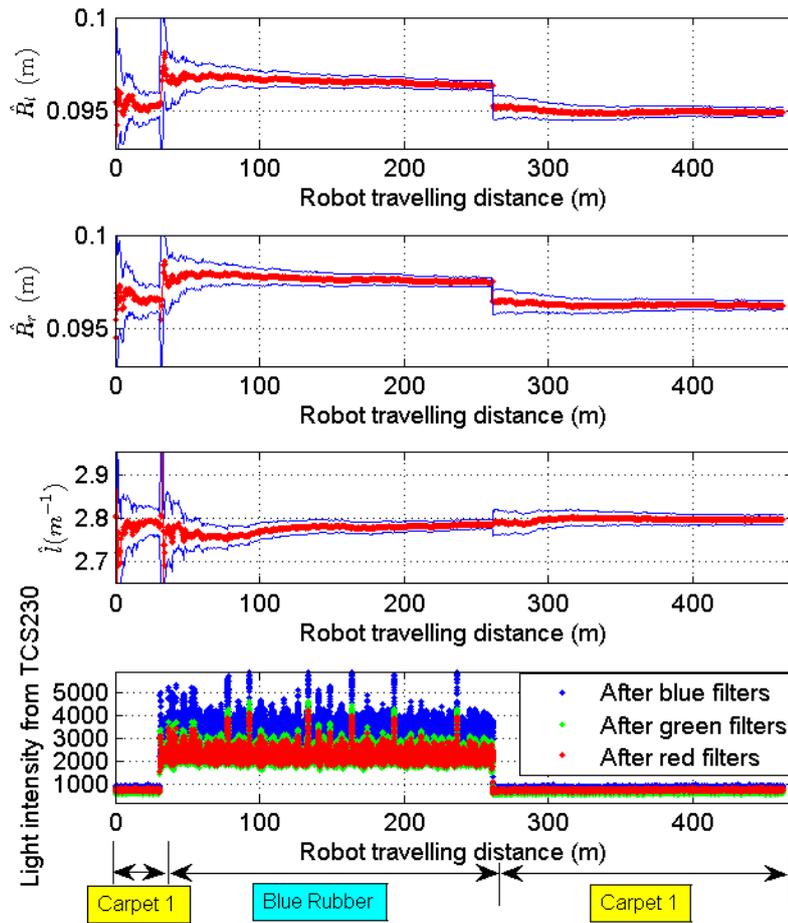


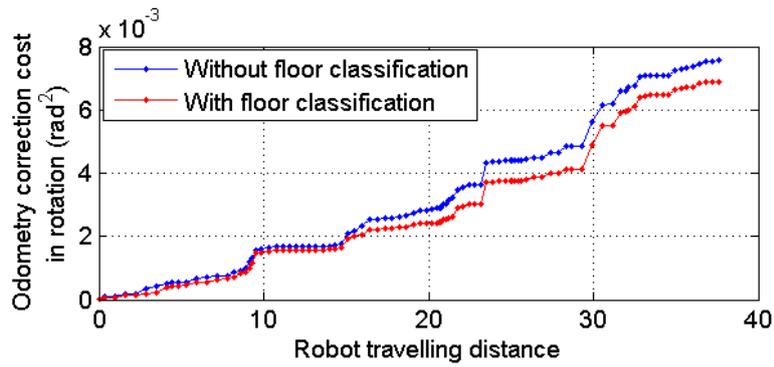
Figure 3.8: Calibration results on Carpet 1 and Blue Rubber. Blue curves in top three plots show ± 2 standard deviations from estimated variables in red. Reused from [1] ©2015 IEEE.



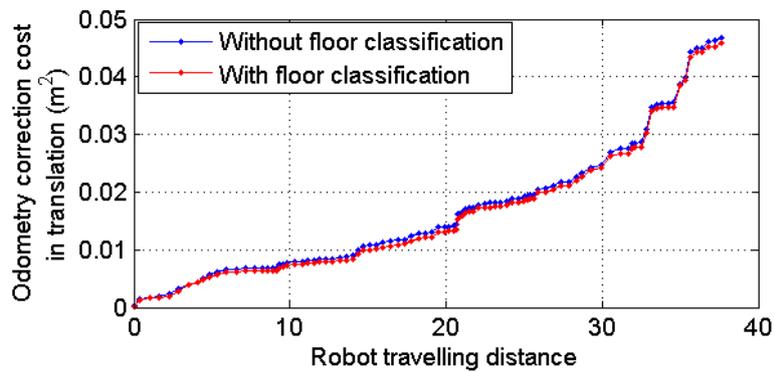
Figure 3.9: Reference graph map for odometry correction cost comparison. The robot started performing SLAM from the bottom right corner of the map on Carpet 1. Then it was steered clock-wise on Carpet 1 and Blue Rubber, on which the trajectories were shown as red and blue lines, respectively. Green dots represent robot positions and yellow lines represent relative robot poses. Reused from [1] ©2015 IEEE.

aligns with the yellow reference graph map and only becomes a little displaced from the graph SLAM map after accumulating the pose estimation errors.

Figure 3.10 shows the odometry correction cost with and without using the developed floor classification method to calibrate the odometry parameters of the robot based on the same recorded dataset.



(a)



(b)

Figure 3.10: The odometry correction cost with and without the proposed floor classification strategy. (a) The odometry correction cost in rotation. (b) The odometry correction cost in translation. Reused from [1] ©2015 IEEE.

3.4. OFFLINE CALIBRATION OF ODOMETRY AND SENSOR POSES ON MULTIPLE FLOORS

Table 3.3: Odometry Calibration Results. Reused from [1] ©2015 IEEE.

| | With floor classification | | Without floor classification |
|--|---------------------------|-------------|------------------------------|
| | Carpet 1 | Blue Rubber | |
| \hat{R}_l (m) | 0.0949433 | 0.0963686 | 0.0955958 |
| \hat{R}_r (m) | 0.096239 | 0.0974883 | 0.0968012 |
| \hat{l} (m ⁻¹) | 2.7971 | 2.78587 | 2.79363 |
| Std dev. of \hat{R}_l (m) | 0.00011818 | 0.00012153 | 0.00008398 |
| Std dev. of \hat{R}_r (m) | 0.00011987 | 0.00012279 | 0.000085003 |
| Std dev. of \hat{l} (m ⁻¹) | 0.0053 | 0.0053 | 0.0038 |

3.4 Offline Calibration of Odometry and Sensor Poses on Multiple Floors

The odometry calibration method in Section 3.3.1 requires known positions and orientations of the onboard LRFs, since the LRFs measurement data provide the reference for EKF to converge to. On the other hand, a better calibrated odometry system can help the estimation of the extrinsic parameters of LRFs if we choose to optimise the quality of a map built by aggregating a series of laser scans and robot poses. This becomes a chicken and egg problem. Therefore, the author proposes a novel approach to calibrate odometry parameters and onboard LRFs extrinsic parameters simultaneously. Every LRF is attached to the robot rigidly and cannot rotate about itself, so we have 3 extrinsic parameters (position and orientation) for each LRF. The author demonstrated the approach by maneuvering the robot on two different floors considering the floor dependence issue. Hence there were 6 odometry parameters in total. This offline approach can provide a more accurate estimation of both odometry parameters and sensor extrinsic parameters, since it is a data driven method and the optimisation algorithms can be repeated many times until all parameters converge to a reasonable

basin with a very low cost value of the cost function. The cost function in this section is map information (MI) that will be detailed in Subsection 3.4.1.

3.4.1 Quality Metric for Occupancy Grid Maps

The Occupancy Grid (OG) map is a common representation of a map built from laser scan measurements and is useful for robot navigation and measuring map quality [27]. The map consists of grid cells with discrete random variables with two outcomes, occupied cells and free cells. The entropy of the information in an OG map cell can be calculated as below

$$H(m_{ij}) = -p(m_{ij}) \log_2(p(m_{ij})) - \bar{p}(m_{ij}) \log_2(\bar{p}(m_{ij})) \quad (3.16)$$

where $p(m_{ij})$ denotes the probability of a cell which has index (i, j) of the map m is occupied, while $\bar{p}(m_{ij})$ represents the probability of that cell which is not occupied. Blanco et al. [60] proposes the map information (MI) metric, $\hat{I}(m)$, and shows it better represents the quality of an OG map. The MI of an OG map m is defined as Equation (3.17)

$$\hat{I}(m) = \begin{cases} \sum_{\forall i,j} (1 - H(m_{ij}))/N, & \text{if } N > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.17)$$

where N is the total number of the cells that have been observed. The MI metric is mostly independent of the grid resolution for the commonly used cell sizes between 1 cm and 10 cm [60]. This metric can be explained intuitively by noting that the better the alignment between observations (laser scans for this research) in the map, the higher the values obtained from MI. For example a cell with probability 0.5 contributes zero to the MI whilst a probability close to 1 or 0 contributes nearly 1. The poses of the on-board sensors for mapping and the odometry parameters of the robot will affect how well the observations align to each other, which is the outcome of MI.

The author develops a similar sensor model to [61] when calculating $p(m_{ij})$, which is initialised to the uniform prior of 0.5 and updated by a ray-casting operation determining its occupancy probability along a beam between the laser origin to the measured point. Readers are referred to [27] and [61] for a detailed explanation about this sensor model and its calculation. Experimentally, the author finds the log likelihood value $l_{occ} = 1.2224$ for an occupied cell and $l_{free} = -0.585$ for a free cell, corresponding to probabilities of 0.7 and 0.4 for occupied and free volumes, work well. This selection is the same as [61], but in order to use bits/cell as the unit for MI this system adopts the base 2 logarithm instead of the natural logarithm.

3.4.2 Fusion of Two Laser Rangefinders Measurements

Sensor extrinsic parameters are very important for the robot to measure accurate absolute value of its perception data, because most sensors, including LRFs, can only measure the relative distance or angles of the target objects. In this case, it is essential to calibrate the relative pose between the coordinate frame of LRFs and the origin of robot odometry coordinate frame. The robot is assumed to work on a horizontal plane and only calibrate three out of six parameters, namely the relative position coordinate (x, y) and relative orientation angle θ .

Using multiple LRFs can reduce the chance of divergence, ambiguity and inaccuracy in the process of scan matching as explained in the previous work [40]. However, most mapping algorithms are designed to work with only one LRF. In this research, the measurements of two Hokuyo URG-04LX LRFs are projected to a virtual one located at the original point of the robot coordinate frame as shown in Figure 3.11. Each range finder has a field of view (FOV) of 240 degrees and valid range measurements from 0.02 m to 4 m. The angular resolution is approximately 0.352 degree and one laser scan contains 681 laser beams. The virtual LRF has a FOV of 360 degrees with 1024 laser beams and the same angular resolution. Assume the two LRFs are placed in the same horizontal plane. Coordinate frames $\{O\}$, $\{R\}$, $\{L_1\}$ and $\{L_2\}$ are the world frame, the robot frame,

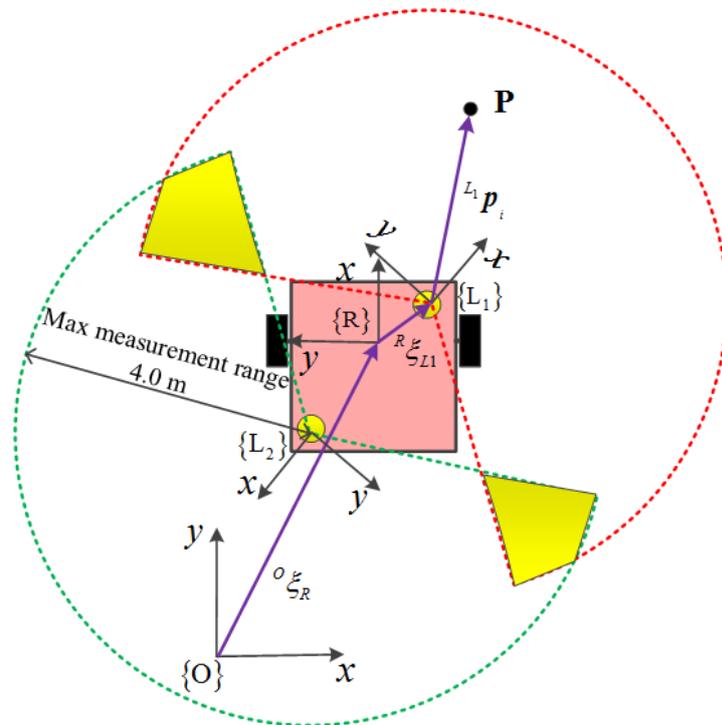


Figure 3.11: Schematic of the laser range finders on the robot. The red and green sectors show the maximum measurement ranges of Laser 1 and Laser 2, respectively. Reused from [3] ©2015 IEEE.

the frame of Laser 1 and the frame of Laser 2, respectively. The relative pose of a frame $\{B\}$ with respect to a frame $\{A\}$ is denoted by the symbol ${}^A\xi_B$. The laser scan of Laser 1 at time step t is described as $S_1(t) = (x_1, y_1, \theta_1, r_i^1(t), \alpha_i^1(t)_{i=1}^n)$, where x_1, y_1, θ_1 are the position and orientation of Laser 1 in $\{R\}$. $r_i^2(t), \alpha_i^2(t)_{i=1}^n$ describes n range measurements $r_i^1(t)$ at bearings $\alpha_i^1(t)$. $n = 1, 2, \dots, N$ (N is the total number of laser beams in one scan). Analogously, the laser scan of Laser 2 is defined as $S_2(t) = (x_2, y_2, \theta_2, r_i^2(t), \alpha_i^2(t)_{i=1}^n)$. Note that the time step label for the relative poses symbols is omitted for the convenience of notation. Let r_i^{Laser1} and α_i^{Laser1} denote the virtual range measurements and bearings seen from the central virtual laser based on the measurements of Laser 1 as in [62].

$$r_i^{Laser1} = \sqrt{(r_i^1 \cos(\theta_1 + \alpha_i^1) + x_1)^2 + (r_i^1 \sin(\theta_1 + \alpha_i^1) + y_1)^2} \quad (3.18)$$

$$\alpha_i^{Laser1} = \text{atan2}(r_i^1 \sin(\theta_1 + \alpha_i^1) + y_1, r_i^1 \cos(\theta_1 + \alpha_i^1) + x_1) \quad (3.19)$$

r_i^{Laser2} and α_i^{Laser2} are calculated in the same way. Before scan projection, two scan processing steps including a median filter and laser segmentation are done. After the scan projection, a linear interpolation is performed and the final range measurements are the measurements as the corresponding laser measurements if there are no common areas between the two laser measurements. In the common area, the shorter range measurement is chosen.

3.4.3 Particle Swarm Optimisation

Inspired by swarm intelligence in a population of simple agents interacting with their neighbours and the whole environment, Kennedy and Eberhart [63] first introduced Particle Swarm Optimisation (PSO) as a computer model for flocking birds in 1995. Later, researchers used this as a metaheuristic to find the optimum solution in a high dimensional problem. It does not require continuous or differentiable fitness function in contrast to gradient descent optimisation methods. A particle contains several properties, namely position, velocity, fitness value, pbest and gbest [64]. The position of the i^{th} particle is a potential solution for the problem and is denoted $\mathbf{X}_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$, where D is the dimension of the

problem. The velocity of the i^{th} particle is the displacement used to calculate the next position and is denoted $\mathbf{V}_i = [v_{i1}, v_{i2}, \dots, v_{iD}]^T$. The fitness value is the function value by applying the position. The pbest $\mathbf{P}_i = [p_{i1}, p_{i2}, \dots, p_{iD}]$ is also called personal best value, which is the best position obtained from the corresponding particle itself. The gbest $\mathbf{G} = [g_1, g_2, \dots, g_D]^T$ is the global best position so far achieved by all the particles in the swarm. The following equations are used to update position and velocity for the i^{th} particle from time step t to $t + 1$ [64]:

$$\mathbf{V}_i^{t+1} = \omega \mathbf{V}_i^t + c_1 \mathbf{R}_{i1}^t \otimes (\mathbf{P}_i^t - \mathbf{X}_i^t) + c_2 \mathbf{R}_{i2}^t \otimes (\mathbf{G} - \mathbf{X}_i^t) \quad (3.20)$$

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + \mathbf{V}_i^{t+1} \quad (3.21)$$

where ω is the inertia coefficient to prevent explosion of the particles. Cognitive acceleration coefficient c_1 and social acceleration coefficient c_2 are used to include the influences of the pbest and gbest for this update. Independent vectors \mathbf{R}_{i1}^t and \mathbf{R}_{i2}^t are uniformly distributed in the range $[0, 1]$, \otimes denotes element-wise vector multiplication. This is the basic PSO algorithm.

3.4.4 Calibration Method

The calibration is divided into three phases: dataset collection, training a SVM and calibration on multiple floors.

Dataset Collection

The trajectory needs to contain sufficient diversity to avoid rank deficiency as stated in [51]. Examples of rank deficiency motions are a pure straight line, or a rotation on the spot or a constant curvature arc motion. A combination of these motions is not rank deficient and allows all parameters to be observable. Furthermore, long datasets are not suitable for offline calibration since the proposed

method requires the dataset to be repeated until the PSO algorithm reaches a steady and large MI value. Experimentally, the author found a dataset of around 350 seconds was a good choice on two testing floors, Carpet 1 and Blue Rubber. An enclosed environment with most parts visible by LRFs was chosen primarily for efficiency reasons. This allowed full use of the range information in building an OG map, whose quality measured in MI was most affected by the calibration parameters of the odometry and the poses of the LRFs.

Training a SVM

Floors should ideally be of even colour for the floor classification method to work well. The classification accuracy will be higher if more data are used to train the SVM, so the robot is driven along a trajectory covering a greater area of the floor than what is used in the calibration phase. In addition, the training data should be based on data after a median filter and down sampling. These pre-processed data were fed into the C++ implementation of SVM for floor classification.

Calibration on Multiple Floors

Calibrated parameters are odometry parameters on different floors and extrinsic parameters of exteroceptive sensors. The fitness value of the PSO algorithm is the MI value of the OG map built using only encoder data and laser extrinsic parameters without scan matching. The suggested swarm size 40 was adopted [64]. Every 100th scan and odometry poses were used to build an OG map. The Standard PSO 2011 (SPSO-2011) was adopted, because it performed well against 28 various functions as reported in [65]. The detail of SPSO-2011 was explained in [64]. A flowchart of the proposed calibration method is in Figure 3.12.

3.4. OFFLINE CALIBRATION OF ODOMETRY AND SENSOR POSES ON MULTIPLE FLOORS

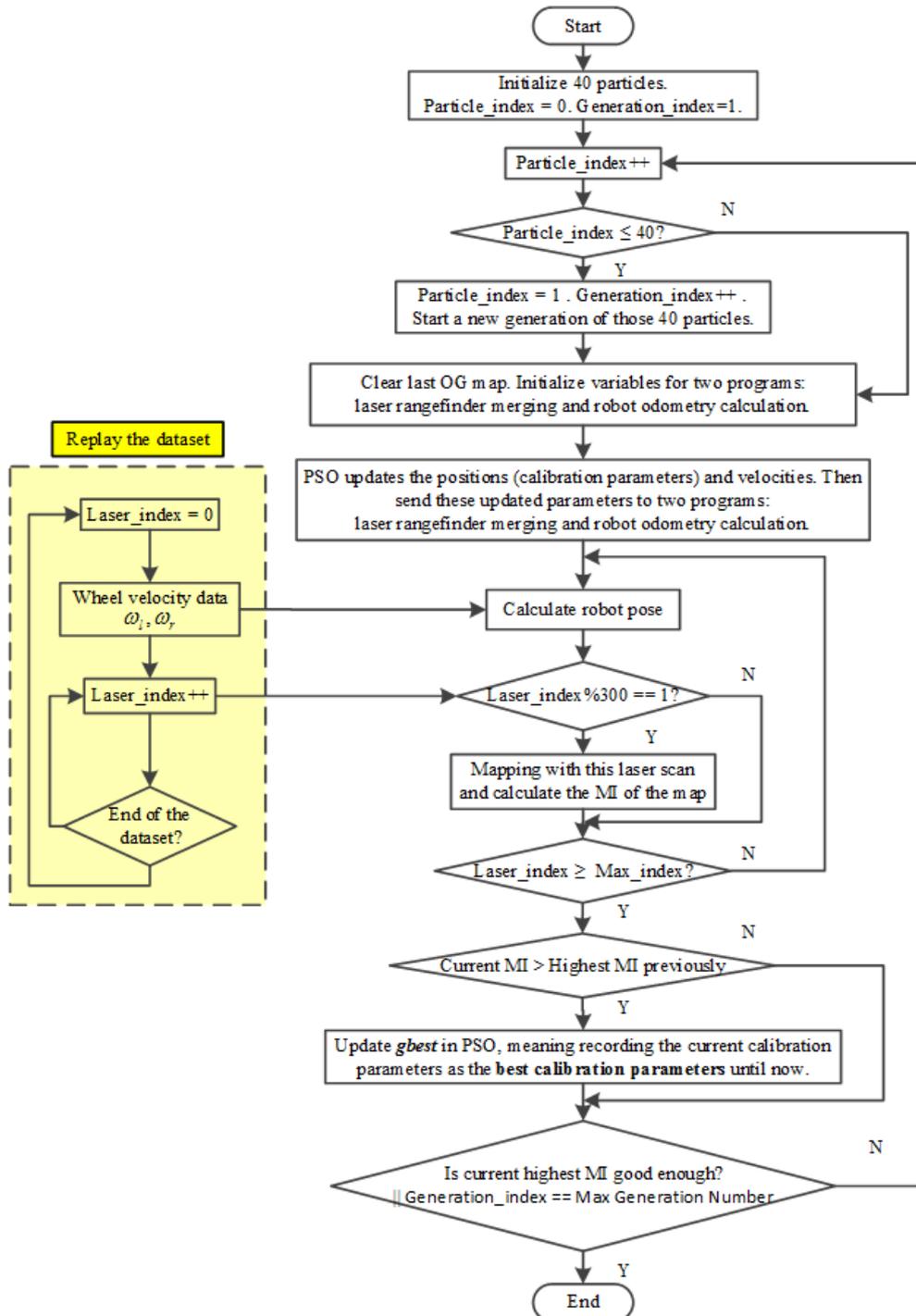


Figure 3.12: Flow chart of calibrating odometry and sensor extrinsic parameters with PSO on multiple floors. Reused from [3] ©2015 IEEE.

3.4.5 Experiments and Results

Experiment Settings

To evaluate the proposed calibration method, two different datasets were recorded, Dataset 1 and Dataset 2, one after the other, on Carpet 1 and Blue Rubber in a closed environment as shown in Figure 3.13(a). Figure 3.13(b) depicts the robot crossing the boundary of two floors. Since every 100th laser scan was used to build the OG map, Dataset 1 and Dataset 2 have 35 and 36 laser scan measurements in their maps, respectively. The designed algorithm replayed the datasets at a rate of twice the original ROS message publish rate via the *rosbag* command-line tool.

Let R_{l1} , R_{r1} and l_1 be the left wheel radius, right wheel radius and the reciprocal of the separation distance B_1 on Carpet 1. R_{l2} , R_{r2} and l_2 are the corresponding variables on Blue Rubber. Let x_1 , y_1 and θ_1 be the extrinsic parameters for Laser 1 and x_2 , y_2 and θ_2 for Laser 2. The above 12 parameters form the calibration parameter vector in this work. According to a preliminary estimation, each calibration parameter is confined as listed in Table 3.4 when performing the PSO algorithm. In terms of the cell size of the OG map, the commonly used cell size, 2.5 cm, was chosen. As suggested by [64], 30 generations of the PSO algorithm in most practical problems are sufficient for a good optimisation result. It was found that there was little improvement after 40 generations of PSO. Therefore, the author chose 40 as the total generations in all the following experiments. On a Lenovo Z580 laptop with an Intel i5 four core CPU working at 2.50 GHz and a RAM of 4 GB, 40 generations of 40 particles in a PSO took approximately 14 hours of computation time. The time was proportional to the number of particles and the number of generations in the PSO.

3.4. OFFLINE CALIBRATION OF ODOMETRY AND SENSOR POSES ON MULTIPLE FLOORS

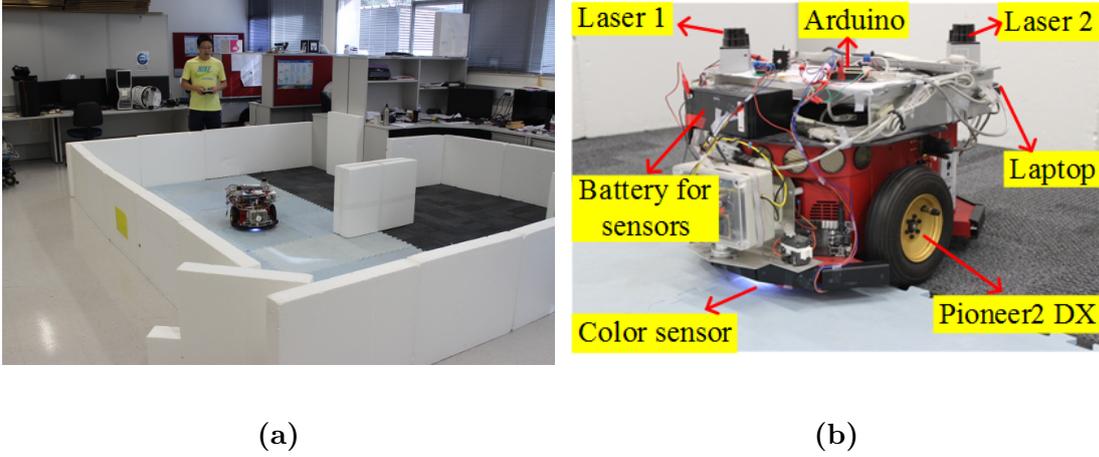


Figure 3.13: (a) The environment used to collect the datasets. The size of the environment was approximately $4\text{ m} \times 5\text{ m}$. (b) The robot is crossing the boundary of two floors, Carpet 1 and Blue Rubber. Reused from [3] ©2015 IEEE.

Table 3.4: Confinement of Each Calibration Parameter in the PSO. Adapted from [3] ©2015 IEEE.

| Parameters | Min | Max |
|------------------------|---------|---------|
| $x_1(\text{m})$ | 0.06 | 0.08 |
| $y_1(\text{m})$ | -0.145 | -0.125 |
| $\theta_1(\text{rad})$ | -0.8154 | -0.7754 |
| $x_2(\text{m})$ | -0.22 | -0.18 |
| $y_2(\text{m})$ | 0.125 | 0.145 |
| $\theta_2(\text{rad})$ | 2.2 | 2.4 |
| $R_{l1}(\text{m})$ | 0.093 | 0.099 |
| $R_{r1}(\text{m})$ | 0.093 | 0.099 |
| $l_1(\text{m}^{-1})$ | 2.765 | 2.84 |
| Continued on next page | | |

Table 3.4 – continued from previous page

| Parameters | Min | Max |
|----------------------|-------|-------|
| $R_{l_2}(\text{m})$ | 0.093 | 0.099 |
| $R_{r_2}(\text{m})$ | 0.093 | 0.099 |
| $l_2(\text{m}^{-1})$ | 2.765 | 2.84 |

Map Quality Improvements with Floor Classification

The effects of the floor classification was investigated by a series of experiments. Firstly, the author calibrated the odometry and laser pose parameters with a correct floor classification using the proposed floor classification system. Then the parameters with a wrong floor classification which separated the robot trajectory into two parts: the first 17 laser scans on Carpet 1 and the other 18 laser scans on Blue Rubber was calibrated. Figure 3.14 shows the differences of the map information evolutions during the two comparison experiments above. MI of the OG map built with the proposed floor classification method is higher. Figure 3.15 shows the evolution of the mean of the MI values of the 40 particles in each generation in the PSO algorithm, which indicates that most particles approached better estimated values. Finally, the calibrated parameters in last two experiments were applied onto Dataset 2 to show the benefits of successful floor classification. Figure 3.16 summarises the experiments. MI values of the OG maps in Figure 3.16 (c) and Figure 3.16 (d) are 0.9234 bits/cell and 0.9207 bits/cell, respectively.

3.4. OFFLINE CALIBRATION OF ODOMETRY AND SENSOR POSES ON MULTIPLE FLOORS

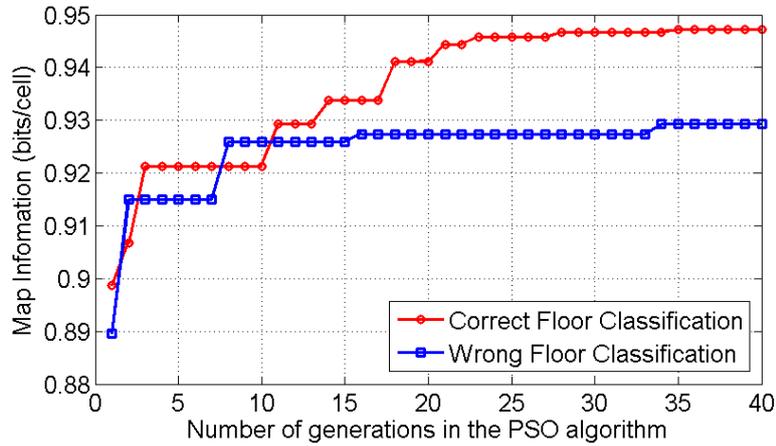


Figure 3.14: The map information evolutions during the simultaneous calibration of odometry and laser pose parameters vs. the number of generations in PSO algorithm when using the proposed floor classification method and a wrong floor classification. Reused from [3] ©2015 IEEE.

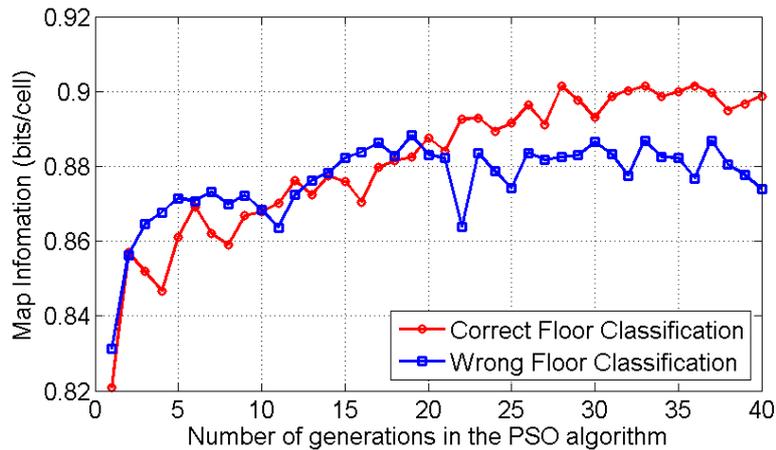


Figure 3.15: The evolution of the mean of the map information values of the 40 particles in each generation in the PSO algorithm. Reused from [3] ©2015 IEEE.

3.4. OFFLINE CALIBRATION OF ODOMETRY AND SENSOR POSES ON MULTIPLE FLOORS

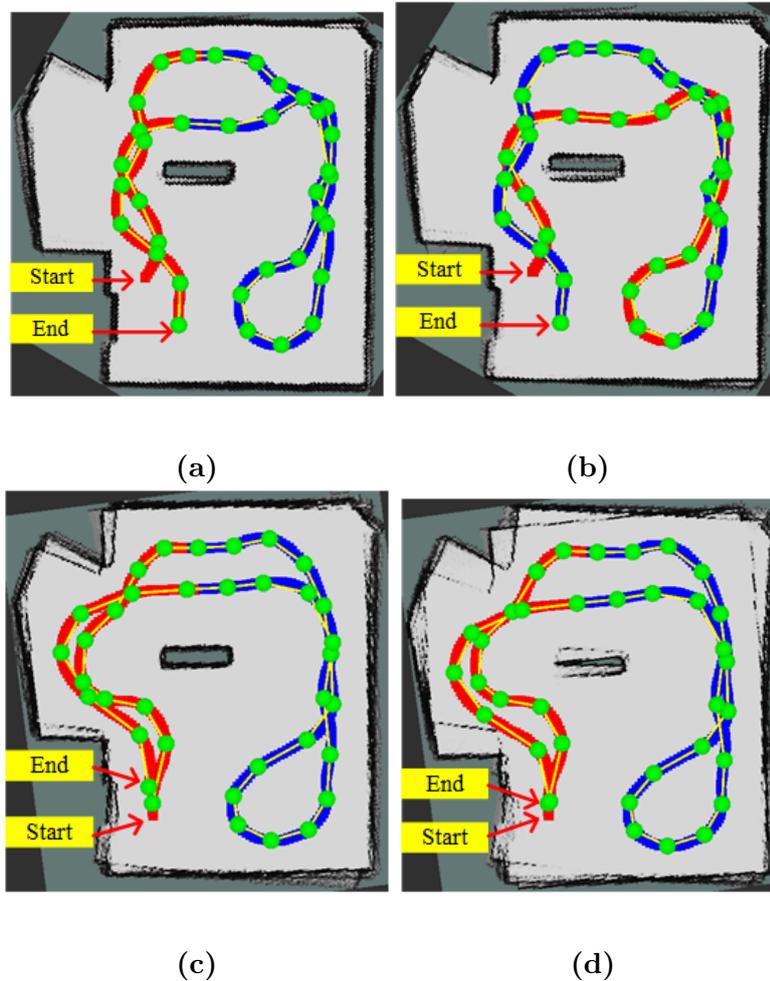


Figure 3.16: Mapping results to show the influence of floor classification. The trajectories labelled with red and blue correspond to the classified results of floor types, Carpet 1 and Blue Rubber, respectively. Green dots depict positions of the robot for the laser scans used to build the OG maps and yellow lines depict the relative pose between the consecutive laser scans. (a) OG map built with the calibrated parameters based on the proposed floor classification method using Dataset 1. (b) OG map built with the calibrated parameters based on a wrong floor classification as a comparison using Dataset 1. (c) OG map built with the parameters in (a) using Dataset 2. (d) OG map built with the parameters in (b) using Dataset 2. Reused from [3] ©2015 IEEE.

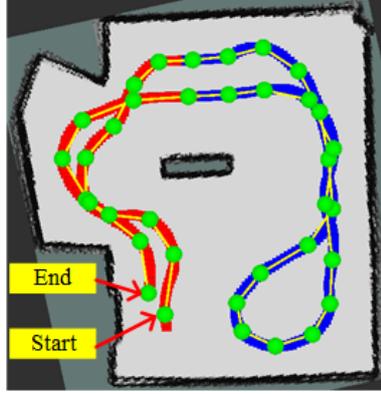


Figure 3.17: OG map built with the calibrated parameters based on the proposed floor classification method using Dataset 2. Reused from [3] ©2015 IEEE.

Consistency Validation on Two Datasets

The consistency of the PSO based offline simultaneous calibration of odometry and sensor extrinsic parameters was justified by calibrating the parameters using different datasets, Dataset 1 and Dataset 2. The map built with the calibrated parameters using Dataset 2 is shown in Figure 3.17. Table 3.5 is the calibration result using two datasets, which shows a consistent result for each parameter. Define $\mathbf{S}(i)$ as Equation (3.22) to test the calibration consistency, where $p_1(i)$ and $p_2(i)$ denote the i^{th} dimension of the calibration parameters using Dataset 1 and Dataset 2, respectively. Figure 3.18 shows the consistency between the calibration results in Table 3.5.

$$\mathbf{S}(i) = \frac{|p_1(i) - p_2(i)|}{|(p_1(i) + p_2(i))/2|} \quad (3.22)$$

3.4. OFFLINE CALIBRATION OF ODOMETRY AND SENSOR POSES ON MULTIPLE FLOORS

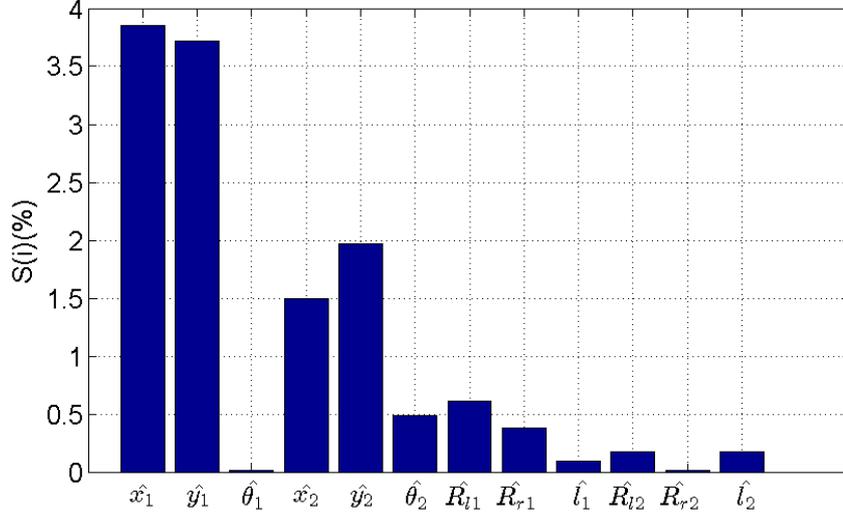


Figure 3.18: Relative calibration differences based on Dataset 1 and Dataset 2 using Equation (3.22). Reused from [3] ©2015 IEEE.

Table 3.5: Calibration Results Using Dataset 1 and Dataset 2. Adapted from [3] ©2015 IEEE.

| | Dataset 1 | Dataset 2 |
|------------------------|-----------|-----------|
| $x_1(\text{m})$ | 0.070456 | 0.0677955 |
| $y_1(\text{m})$ | -0.13935 | -0.134264 |
| $\theta_1(\text{rad})$ | -0.799576 | -0.799708 |
| $x_2(\text{m})$ | -0.203149 | -0.206218 |
| $y_2(\text{m})$ | 0.133731 | 0.136393 |
| $\theta_2(\text{rad})$ | 2.2913 | 2.3024 |
| $R_{l1}(\text{m})$ | 0.0945349 | 0.0951182 |
| $R_{r1}(\text{m})$ | 0.0960021 | 0.0963673 |
| $l_1(\text{m}^{-1})$ | 2.79122 | 2.7887 |
| $R_{l2}(\text{m})$ | 0.0961968 | 0.0963655 |
| $R_{r2}(\text{m})$ | 0.0973374 | 0.0973481 |
| $l_2(\text{m}^{-1})$ | 2.80303 | 2.79807 |

3.4.6 Convergence Basin Verification

PSO, like other metaheuristic optimisation algorithms, cannot guarantee finding the global optima [66]. Therefore the author applied a grid search upon the odometry parameters to show this parameter estimation problem had a reasonable convergence basin.

The testing environment is shown in Figure 3.19, where the robot followed an “8” shape path with a fixed speed at 0.1 m/s and performs 90° rotating at each corner. The MI map quality was applied. The odometry parameters, R , $RonB$ and $delta$ were chosen as [24], which can decouple their effects on the map quality. R , $RonB$ and $delta$ are defined as Equation (3.23), Equation (3.24) and Equation (3.25). Grid search has a resolution of 0.00025 m, 0.00025 and 0.00008 for R , $RonB$ and $delta$, respectively and each dimension of the search has 35 grid elements. Grid search results are shown in Figure 3.20 indicating that there is a reasonable convergence basin.

$$R = (R_r + R_l)/2 \quad (3.23)$$

$$RonB = R/B \quad (3.24)$$

$$delta = \frac{R_r - R_l}{R_r + R_l} \quad (3.25)$$

3.4. OFFLINE CALIBRATION OF ODOMETRY AND SENSOR POSES ON MULTIPLE FLOORS



(a)

(b)

Figure 3.19: Experiment with an “8” shape trajectory on Lino to verify the convergence basin of the odometry parameters.(a) Surrounding environment. (b) Running.

3.4. OFFLINE CALIBRATION OF ODOMETRY AND SENSOR POSES ON MULTIPLE FLOORS

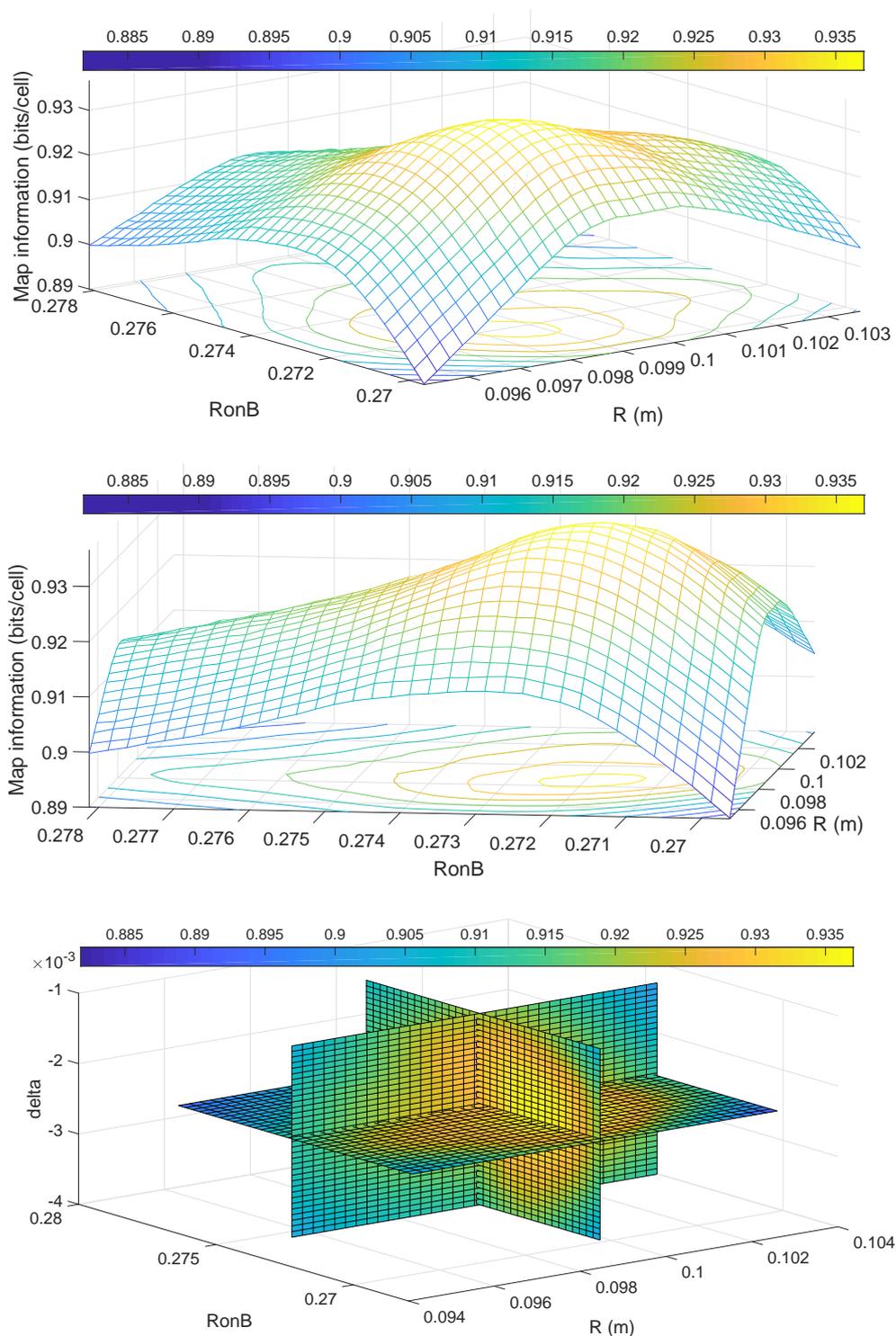


Figure 3.20: Convergence basin for the odometry calibration with the robot following an “8” shape trajectory on Lino. (a) Mesh plot with contour when fixing δ . (b) Another view of (a). (c) Three slices of odometry parameters.

3.5 Summary and Conclusions

In this chapter, online EKF based mobile robot odometry calibration on multiple floor types has been developed. The map quality from the graph based SLAM system in a lab environment has been improved using the proposed method. This research has presented new experimental results that confirm that the odometry calibration of a differential drive robot depends on the floor type. This is often mentioned in robotics papers yet there has been little published justification of this dependency. This research shows experimentally that odometry calibration with floor classification is superior to that without floor classification on multiple floors.

In addition, this chapter develops a novel offline simultaneous calibration of robot odometry and sensor extrinsic parameters using the PSO algorithm on multiple floors. A state-of-the-art map quality metric for OG map, Map Information [60], has been chosen as the fitness value of the PSO algorithm. It has been shown that a calibration with the colour floor classification system is more accurate for each parameter than using an arbitrary classification. Note that 12 parameters are used in both the arbitrary and colour classified cases to achieve a fair comparison, since a comparison with single floor region would involve optimising over just 9 parameters and therefore disadvantage the single region. Furthermore, the calibration method has been tested for consistency between two independent datasets. The PSO based calibration framework could be applied to calibrate odometry and the extrinsic sensor parameters for 3D laser scanners or Kinect sensors by extending the MI metric to 3D occupancy grid maps built with OctoMap [61]. This research takes a novel approach for calibrating odometry and extrinsic sensor parameters by sensing and segmenting the environment into different floor surface types. In subsequent chapters, the effect of robot acceleration is investigated for its impact of odometry accuracy and calculator.

Odometry Model Incorporating Linear Acceleration

4.1 Introduction

Many mobile robots have differential drive wheels with pneumatic tyres. Many of these robots depend on odometry pose estimations when performing map building or localisation tasks. Currently a three parameter odometry estimation model is widely used as explained in Chapter 3. This model works well when robots run at a low acceleration, i.e., a steady or slowly changing velocity. However, Equations (3.1-3.5) assume R_l , R_r and B to be time invariant. This assumption is *not* correct when a robot accelerates or decelerates as shown in this chapter and also published here [4].

This chapter discusses the effects of acceleration on robot odometry calibration. As a robot accelerates, wheel slip increases. Hence the effective wheel radius is reduced. This can affect odometry accuracy and a new odometry model to handle this effect is developed in this chapter. Researchers in the field of tyre-road contact mechanics have known that *slip* or *skid* is the main reason for effective tyre radius differences [67]. Those researchers have come up with a series of theoretical and empirical models, including knowledge of contact mechanics, physics and materials. This theory is applied to the research in this chapter. The assumptions in this chapter are listed below.

- The robot has pneumatic tyres.

- The floor is flat, hard and smooth with in-significant inclination variation over the area robots are working on.
- The floor is static and temperatures do not change over the testing period.
- The surface adhesion forces between floors and the robot are insignificant.
- The two wheels of the robot have similar properties.
- The velocity of the robot is sufficiently low that air resistance is insignificant.
- No significant lateral sliding forces are present.
- No obstacles contact the robot or affect (e.g., strong magnetic field) the movement of the robot.

Wheel slippage is known to cause odometry estimation errors. A thorough study is provided in [68], where the authors present a kinematic EKF approach to improve the localisation accuracy of two skid-steer outdoor robots with extensive field testing. They report open-loop prediction errors on one wheeled robot which is about 15% of the travelling distance compared to 45% for the no-slip model. A robust yet non-conservative motion planner is designed for outdoor tracked mobile robots working on deformable high slip terrain [69]. They also point out that such a path planner that considers high slippage can potentially work well for practical situations together with terrain classification methods.

Other researchers have contributed to improving estimation of wheel slippage. Ojeda et al. [70] designed a slippage detection system using wheel motor current sensors for planetary rovers working on soft soils and compensate this effect to improve the localisation accuracy. In [71], the authors aimed to predict slip

remotely using visual information only for Mars rovers. They developed a predictive learning system which managed to map the appearance and geometry of the upcoming terrain to its mechanical properties for traversing. Ding et al. [72] comprehensively studied the performance of driving wheels for planetary rovers running in deformable soils. However, these publications mostly focused on outdoor unmanned mobile robots.

Balakrishna and Ghosal [73] proposed a model where the adhesion coefficient between the wheels of a mobile robot and hard flat floors was a function of wheel slip. Using numerical simulations, they demonstrated the necessity to model the slippage effect for wheeled mobile robots when tracking their trajectories. Only when the adhesion coefficient was chosen large representing conditions closer to no-slip rolling, the path deviation was small. It is also worth mentioning that Konduri et al. [74] researched into the conditions of slip for multiple robot coordination control. They developed a trajectory tracking controller to set the maximum value of the input wheel torque before slip happened. This can help multiple mobile robots to locate each other more accurately, which is desirable in a coordinating task. In the experiments, they realised localisation errors were larger especially when the robots accelerated or decelerated. This falls into the research focus of this chapter.

There have been significant research efforts into pneumatic tyres for road vehicles. In [75], researchers presented their model predictive and feedback controllers which prioritised collision avoidance and stabilisation for autonomous vehicles by incorporating tyre nonlinearities when slipping. They demonstrated their controller's capability to handle emergency scenarios when the vehicle operated at its limited speed under a tyre friction. Song et al. [76] proposed a close loop controller to improve path tracking at high speeds and near tyre traction limits. They used the Dugoff tyre model to calculate the remaining tyre force instead of measuring slip angles online. They verified their controller in three test scenarios in simulation tool, CarSim. With the same aim as [76], researchers from Stanford University [77] applied PD-type and quadratically optimal (Q-ILC) iterative

learning controllers on a real autonomous Audi TTS race vehicle driving around Thunderhill Raceway in Willows, CA at combined vehicle accelerations of up to 8 m/s^2 .

To the author’s knowledge, few publications have investigated negative effects on the quality of maps built by wheeled mobile robots on indoor smooth floors when experiencing wheel slip using differential driven odometry models except the author’s recent publication [4]. Wheel slip can occur even for low speeds ($<0.6 \text{ m/s}$) and low accelerations ($<0.35 \text{ m/s}^2$), which is verified on a Pioneer2 DX H-8 robot via two independent experiments in an office environment [4]. As explained in [67], pneumatic tyres deform due to contact patch dynamics. Compression and stretching of tyres result in a change of effective wheel radii when the vehicle accelerates. Note that the author refers to this effect as slip and the term slide will refer to the effect where an entire contact patch of the tyre loses grip with the floor.

This chapter develops and experimentally justifies a novel differential drive wheeled mobile robot odometry model that includes changes in effective wheel radius due to linear acceleration. Section 4.2 introduces the tyre road contact theory including the definition of longitudinal slip which is the cause of wheel radius changes during accelerating and decelerating. A detailed time synchronisation approach to overcome the time synchronisation problems between LRFs and wheel encoders is developed in Section 4.3. Section 4.4 is dedicated to calibrate the distance measurement model for two LRFs taking into account of warm-up time, incidence angle of a laser beam and margin of errors of LRFs. In Section 4.5, laboratory experiments are presented showing the significance of acceleration on effective wheel radii under conditions encountered by indoor mobile robots. It is demonstrated that the developed model is better than a calibration that neglects the effects of robot linear acceleration (i.e. the status quo). This research demonstrates that the acceleration model improves the map quality in OG maps produced by a LRF and odometry. This chapter is summarised in Section 4.6.

4.2 Tyre Road Interaction Theory

The primary cause of tyre rolling resistance on hard surfaces is the hysteresis in tyre materials due to the deflection of the tyre carcass while rolling [67]. Friction between tyres and roads caused by sliding, the resistance due to air circulating inside the tyre, and other effects are less significant [67]. This is also valid for a robot with pneumatic tyres running on flat level indoor floors. The effective rolling radius is defined as the ratio of linear speed of tyre centre to the angular speed of the tyre when rolling [67, 78]. This corresponds to the wheel radius used by wheeled mobile robot odometry in Equations (3.3-3.5). Tyres often experience both longitudinal and lateral forces when rolling. However, the effective rolling radius is mainly related to the longitudinal force. In contrast, lateral forces on tyres could influence the wheel separation distance. Tyre lateral force effects are reserved for Chapter 5. Here we consider only straight line paths.

4.2.1 Longitudinal Slip

A tractive force will be generated at the tyre-ground contact patch if a driving torque is applied to a pneumatic tyre. Hence, there will be compression between the tyre tread in front of and within the contact patch. This effect, as noted in [67], will result in a shortened travelling distance of the tyre when subject to a driving torque compared to a free rolling tyre. This is longitudinal slip, i , defined in [67] as:

$$i = \left(1 - \frac{V}{r\omega}\right) = \left(1 - \frac{r_e}{r}\right) \quad (4.1)$$

where V is the speed of the tyre centre, ω is the angular velocity of the tyre, r is the radius of the free-rolling tyre, and r_e is the effective rolling radius of the tyre for a driven wheel. Under a driving torque, the tyre experiences a positive value of longitudinal slip as result of $r\omega > V$. Many research papers [79, 80, 81] conclude that the wheel torque and tractive force increase linearly with slip as in Equations (4.2) when the slip is below a critical value (typically larger than 5% under different surface conditions from experiments [81]), because slip is mainly

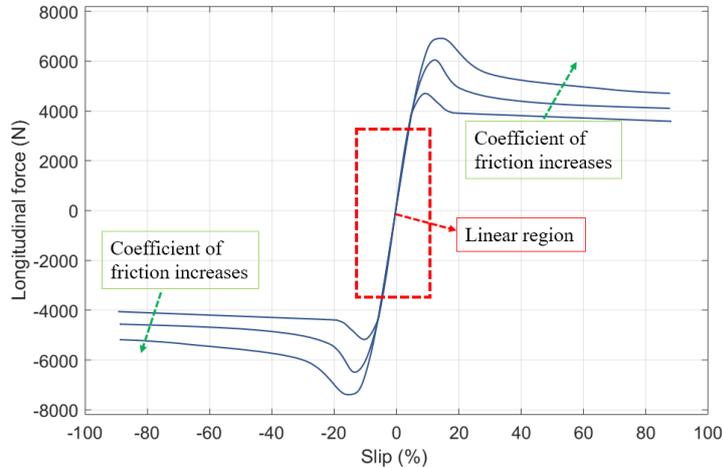


Figure 4.1: One typical Magic Formula tyre model curve. Reused from [4] ©2017 IEEE.

due to elastic deformation of the tyre tread in that region. This can be graphically shown in Figure 4.1 from the empirical tyre model known as the Magic Formula [79].

$$F_x = C_i i \quad (4.2)$$

F_x stands for the tractive force, C_i the longitudinal stiffness of the tyre and i is defined above as longitudinal slip. Indoor mobile robots are normally running within this linear region when their speeds are low without severe acceleration or deceleration. To calculate the effective radius for a driven tyre, we substitute Equations (4.2) into Equations (4.1).

$$r_e = r \left(1 - \frac{F_x}{C_i} \right) \quad (4.3)$$

For the same robot running slowly and steadily on a straight path on different hard indoor floors over a short time interval, r and C_i do not change. Therefore, the tractive force F_x (and hence acceleration) applied on the robot can reveal the effective rolling radius in a linear way.

4.3 Time Synchronisation of Lasers and Encoders

In this section the experimental results to justify the acceleration model are presented and the effects of acceleration on odometry estimation are shown to be significant for mobile robots under normal conditions. In order to perform accurate measurements for use as ground truth the author uses distance measurements from LRFs data fitted by straight lines of walls. These must be carefully calibrated and synchronised to the odometry wheel angle measurements since the experiments for acceleration necessarily involve dynamic measurement of robot position from odometry and laser measurements concurrently.

Compared to Chapter 3, this chapter improves the calibration accuracy of the extrinsic parameters of two LRFs by applying the detailed time synchronization techniques based on the method in [82, 83]. For the time synchronisation between host computer and one LRF, the author customised the implementation of the algorithm of [83] in ROS, i.e., *urg_node-hydro-devel*. The algorithm in [82] to compensate for the delays of each beam in one laser scan was utilised, because encoder data can only be synchronised with the first beam of a laser scan.

The data processing described above is sufficient for one LRF for mapping or localisation purposes. However, multiple LRFs are used in most robotic applications to cover the surroundings of the robot. In this experiment, measurements of two independent LRFs were fused into one virtual scan sitting at the origin of the robot odometry coordinate frame. Therefore, the timestamps of two LRFs and robot encoder sensors were synchronised. The sampling rate of the LRF and encoders were 10 Hz and 30 Hz, respectively. The timestamp of each front laser scan was matched to a rear laser scan. Then the system synchronised both these scans to the mean timestamp of the corresponding scans. Afterwards, the author performed laser beam pose adjustment for the two scans separately by considering both laser beam sweeping delays and the offset of the scan timestamp to the mean timestamp of two LRFs. The final step was to fuse the adjusted beams from two LRFs to the odometry centre as in [62]. Figure 4.2 and Figure 4.3 shows

4.3. TIME SYNCHRONISATION OF LASERS AND ENCODERS

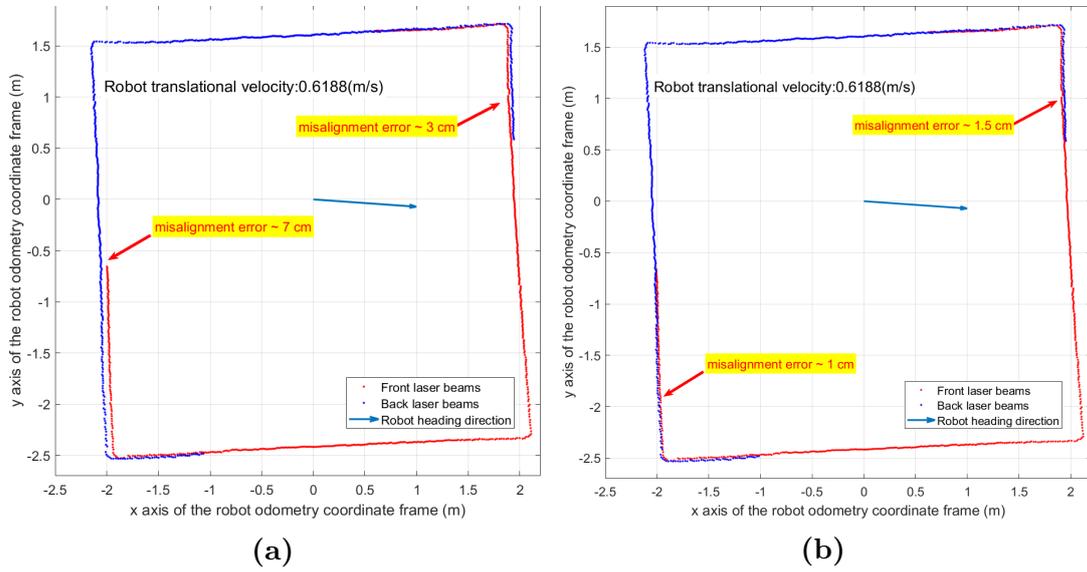


Figure 4.2: Front and rear laser registration errors with beam synchronisation adjustment for robot straight line motion. (a) Before. (b) After. Reused from [4] ©2017 IEEE.

the improvement in terms of the misalignment error from a pair of scans with the nearest timestamps from two LRFs.

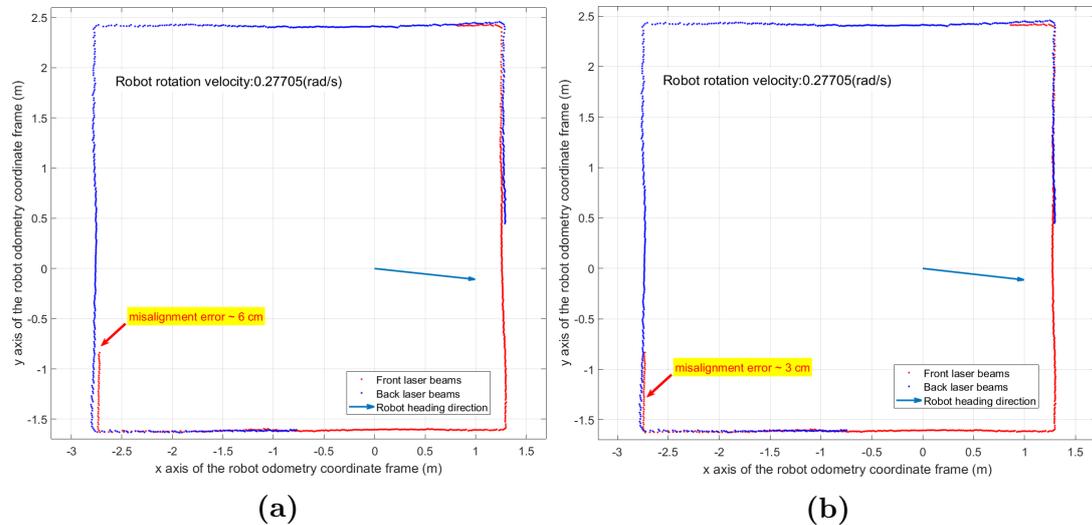


Figure 4.3: Front and rear laser registration errors with beam synchronisation adjustment for robot rotation on the spot motion. (a) Before. (b) After. Reused from [4] ©2017 IEEE.

4.4 Laser Distance Measurement Calibration

From robot mapping experiments using LRFs, the author noticed that their distance measurements were slightly different when the target distance varied. For example a measurement difference of 38 mm was obtained when measuring an object sitting 4 m away. This matches the accuracy of $\pm 1\%$ quoted in the specification sheet of Hokuyo URG-04LX [84]. Characterisation of the same Hokuyo URG-04LX LRF is reported in [85, 86]. To reach a higher accuracy of scan measurements needed for the acceleration experiments, a series of calibration experiments were performed. As suggested in [86], the designed experiments allowed at least 90 minutes of warm-up time prior to recording laser data in all tests in Chapter 4 and Chapter 5. Kneip et al. [85] and Okubo et al. [86] also pointed out the effect of incidence angle for LRFs. Okubo et al. [86] concluded that an incidence in the range of ± 20 degrees would not cause significant errors, so the measurement was narrowed to that range of incident angles.

4.4. LASER DISTANCE MEASUREMENT CALIBRATION

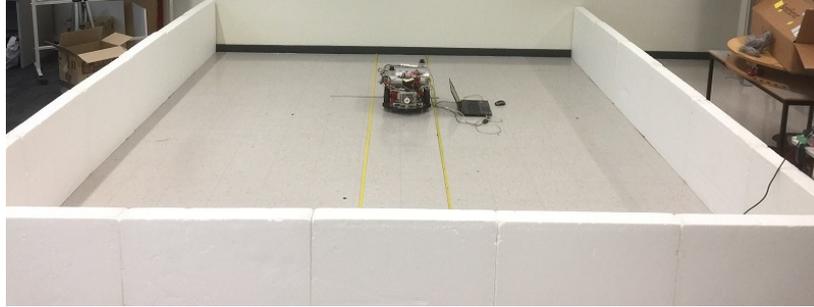


Figure 4.4: Experimental setup for calibration of the laser rangefinders using tape measurements. Reused from [4] ©2017 IEEE.

The calibration arrangement for the two LRFs is shown in Figure 4.4 and the results are in Figure 4.5. The author used two parallel tape measures as the reference to locate the robot moving perpendicular towards a straight smooth white wall from 4 m to 0.5 m with increments of -0.5 m. Eight independent scan datasets (5 minutes, i.e., 3000 scans) were collected. The offsets of tape measurements were manually measured, because there was approximately 1.8 cm offset from the wall to the start point of the floor and some offsets from the lasers to the reference points on the robot when the author manually aligned the robot to certain tape measurement readings.

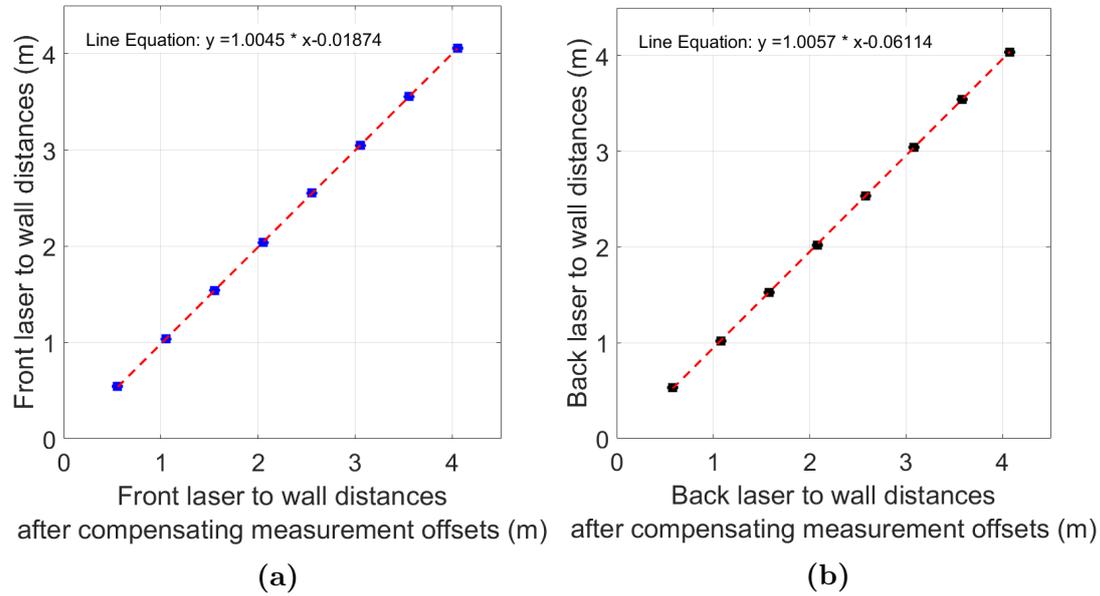


Figure 4.5: Results of calibration of the scales and offsets of the LRFs using the setup in Figure 4.4. (a) Front LRF. (b) Back LRF. Reused from [4] ©2017 IEEE.

4.5 Experiments and Results

4.5.1 Constant Linear Acceleration Tests

The averages of effective left and right wheel radii of the robot under different linear accelerations were estimated. The robot moved straight and perpendicular to the front and the back walls. Wheel radii can be estimated by comparing the robot odometry position with the LRF measurements of both walls. A square (≈ 3.9 m per side) environment was designed with white polystyrene boards as in Figure 4.6. The robot was commanded to accelerate straight for 1 m and decelerate for 1 m before rotating 180° and repeating in the opposite direction. This process was repeated at least 14 times for different acceleration values.

The scans during constant accelerating and decelerating were segmented for processing. Line fitting was performed for the laser beams on the front and the back walls, since the walls were constructed to be as straight as possible. The algorithm

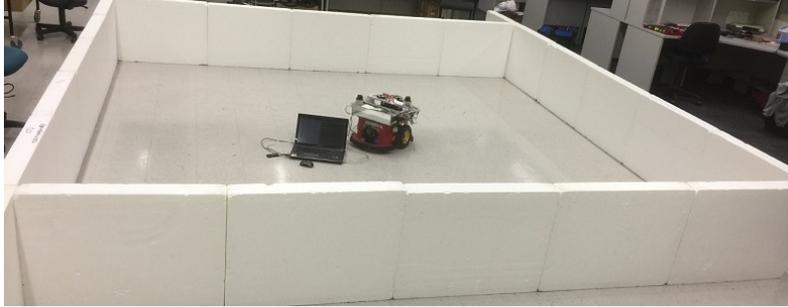


Figure 4.6: Square environment for experiments of linear acceleration scenarios. Reused from [4] ©2017 IEEE.

restricted the measurements to only laser beams within 20° of the wall perpendicular for line fitting to avoid erroneous incident angles to the wall as reported in [86]. Next the author calculated the distance from the odometry centre to the cross point between the robot heading direction and the walls. The changes of distances to the front wall and the back wall were averaged, functioning as the ground truth robot distance compared to robot odometry estimation. See Figure 4.7, Figure 4.8 and Figure 4.9.

From Figure 4.10, it is clear that a linear model could fit the effective wheel radii and linear acceleration data as suggested by the theory in Section 4.2.1. The author also tried to fit a sigmoid function to these data, but discarded this idea due to the concern of overfitting. This also indicates that the robot was running in the linear region of Figure 4.1. This relationship can be expressed in the following equations.

$$R_r = R_{r0} + k_a a \quad (4.4)$$

$$R_l = R_{l0} + k_a a \quad (4.5)$$

where a is robot linear acceleration, R_{r0} is the right wheel effective radius when free rolling and R_{l0} is the left wheel effective radius when free rolling. Equation (4.4) and Equation (4.5) compensate wheel radius shrinking during accelerating and enlarging during decelerating using robot linear acceleration (when k_a is negative). Therefore, the novel odometry estimation model for differential drive wheeled mobile robot on flat floors including Equations (3.1), (3.2),

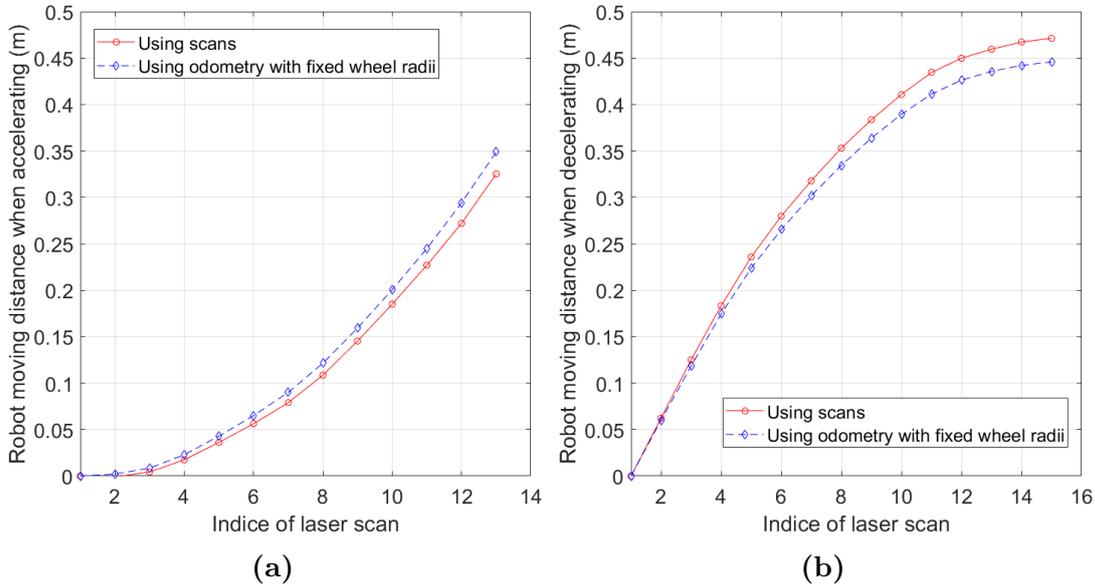


Figure 4.7: Robot moving distances over one straight path using conventional odometry estimation with equal and constant (0.098 m) wheel radius assumption for left and right wheels. Distance measurements using scans are treated as ground truth. (a) Accelerating. (b) Decelerating.

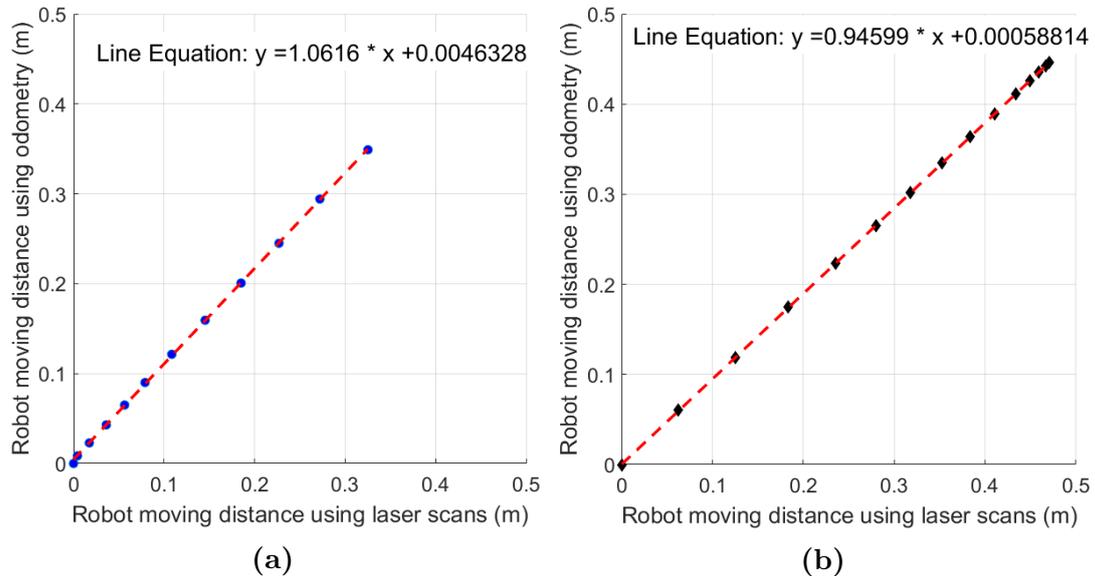
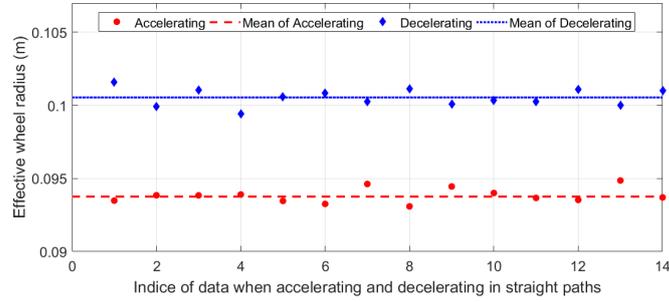
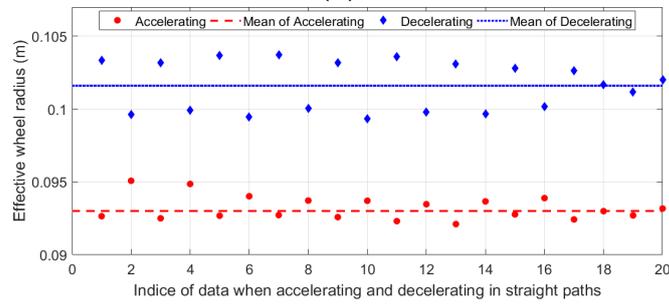


Figure 4.8: Linear correlation between the distance measurements in Figure 4.7 using odometry estimation and scans. (a) Accelerating. (b) Decelerating. Reused from [4] ©2017 IEEE.



(a)



(b)

Figure 4.9: Effective wheel radius. The odd (even) indices traverse the floor in the North (South) direction. (a) Acceleration = 0.3 m/s^2 ; (b) Acceleration = 0.45 m/s^2 . Reused from [4] ©2017 IEEE.

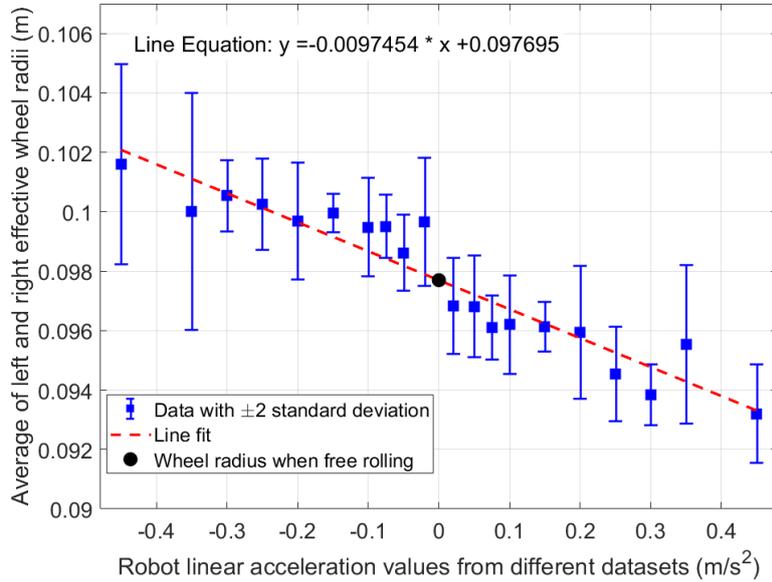


Figure 4.10: Summary of wall experiments with ± 2 standard deviation error bars confirming the linear variation of effective wheel radius with acceleration. Reused from [4] ©2017 IEEE.

(3.3), (3.4), (3.5), (4.4) and (4.5) is developed. It has four parameters, namely R_{r0} , R_{l0} , B and k_a .

4.5.2 Calibration Incorporating Linear Acceleration

As an independent verification of the previous experimental results, the author calibrated the four odometry parameters using PSO based method in Chapter 3. An OG map was generated from overlapping laser scans and odometry alone (i.e. no SLAM processing) and the details can be found in Chapter 3. As a set of odometry parameters better match the robot experimental results, the map becomes crisper and more consistent. Using a map quality measure of the number of occupied cells (i.e. a laser measurement is recorded in an occupied cell) is a measure of how well different laser scans of the same region, such as a wall, overlap appropriately. A fuzzy map due to poor localization of the robot odometry will generate more occupied cells than a well calibrated odometry map. Therefore, the map quality of the experiments are shown in both metrics, number of grid cells and MI, in Table 4.1.

In experiments, the robot was commanded with constant acceleration/deceleration value of 0.35 m/s^2 and followed a square path in an environment like Figure 4.6. The robot was first driven in two counter-clockwise square loops followed by a 180° rotation at a low rotation velocity of 0.17 rad/s . Then it performed two clockwise square loops. The system sampled evenly 245 laser scans corresponding to robot movements of at least 0.2 m or 20° for mapping the environment with robot pose estimation. The grid cell of the map was set to be 5 mm square. These operations were designed to illuminate the influence of other non-systematic odometry estimation errors.

The PSO based odometry calibration was applied after collecting several independent datasets including two LRFs data, encoder data, their timestamps and other necessary data. The PSO tried to optimise the quality of the occupancy grid maps generated by attaching the fused two laser scans to the robot pose estimation based only on odometry estimation. PSO was configured to have 49 particles with the Von-Neumann neighbourhood topology.

Note that the developed four parameters robot odometry model has one more parameter compared to the conventional model. This will add one more dimension of parameter space for modelling the problem, resulting in a potential advantage when estimating the robot pose. For fairness in the comparison, the author also compares the developed model with another simple four parameter odometry estimation model where the 4th parameter is k_v in Equation (4.6) and Equation (4.7). This is a naive idea consisting of 4 odometry parameters for the purpose of comparison in this section.

$$R_r = R_{r0} + k_v v \quad (4.6)$$

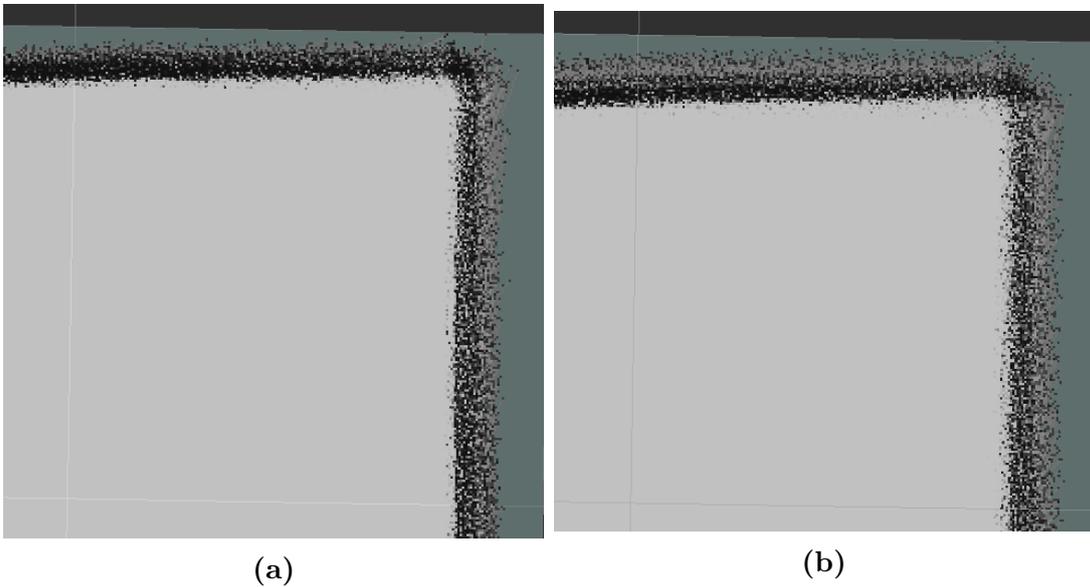
$$R_l = R_{l0} + k_v v \quad (4.7)$$

where v is robot speed, k_v is a constant factor for compensating fictional effects of robot velocity on the effective wheel radii. This (invalid) model is not expected to result in better map quality but acts as an experimental control.

After about 310 generations of evolutions, PSO converged to steady optimal estimation results. The odometry calibration results are shown in Table 4.1. From the table, it is clear that the proposed model results a better quality OG map (in both map quality metric) using the same LRF data and encoder data. The alternative model with a velocity factor beats the conventional three parameter model, because it has the advantage of one more parameter dimension for the PSO to search the optimal solution. By calculating the number of cells in the OG map, it is shown the proposed model has a 0.3% improvement over the conventional model. It is also worth mentioning the slope (-0.0097) in Figure 4.10 and the optimal k_a (-0.0073) in Table 4.1 are similar. This shows the proposed model is consistent in the two independent experiments. Figure 4.11 allows a zoom-in view of visual inspection of two OG maps built using the developed model with opposite signs of acceleration factor k_a .

Table 4.1: Odometry Calibration Results Using Different Models on the Same Real Environment Dataset. Reused from [4] ©2017 IEEE.

| | Proposed model | Conceivable model with velocity factor | Traditional model (3 parameters) | The proposed model with opposite sign of k_a |
|-----------------------------|--------------------|--|----------------------------------|--|
| R_l (m) | 0.098360 | 0.101256 | 0.098028 | 0.098360 |
| R_r (m) | 0.097966 | 0.100856 | 0.097627 | 0.097966 |
| B (m) | 0.352937 | 0.363715 | 0.352098 | 0.352937 |
| k_a (s ²) | -0.00726184 | N/A | N/A | 0.00726184 |
| k_v (s) | N/A | -0.00619651 | N/A | N/A |
| Cell number in OG map | 671,964 | 673,742 | 674,006 | 681,134 |
| Map Information (bits/cell) | 0.984728 | 0.982253 | 0.981816 | 0.972798 |

**Figure 4.11:** Top right corner (1 m by 1 m) of the occupancy grid map built by attaching 245 scans to robot pose estimation results after more than 300 generations of PSO based odometry calibration. (a) Using optimal odometry parameters of the developed model. (b) Using opposite sign of k_a , i.e., 0.00726184 s². Reused from [4] ©2017 IEEE.

4.6 Summary and Conclusions

Below is a suggested procedure of calibrating the differential driven mobile robot using the proposed four odometry parameters model under linear acceleration conditions.

1. Calibrate sensor extrinsic and intrinsic parameters. Users can choose a range of sensors, such as LRF, Kinect, sonar, various cameras, with the aim to measure the robot pose accurately. The sensors can be onboard sensors or external ones.
2. Perform time synchronisation between host computer and the sensors, including wheel encoders [82, 83].
3. Calibrate four odometry parameters including k_a using an appropriate method [5, 21, 23, 1, 50, 3, 87, 88]. An offline optimisation method [3] has been used to choose values of the four odometry parameters that maximises the map quality.

The following conclusions are reached:

1. The effect of acceleration on the robot odometry wheel radius is significant. A wheel radii variation of 9% was measured when acceleration ranges from -0.45 m/s^2 to 0.45 m/s^2 .
2. The linear change in wheel radius with acceleration predicted by theory has been validated by experiments. Improvements in laser map quality have also been demonstrated using the new odometry model.
3. The developed model requires little extra computation and knowledge of the current robot acceleration in addition to the usual wheel angle changes

on each time step. Whilst it is possible to extract the robot acceleration by twice differentiating odometry position or the use of accelerometers, these are both relatively noisy. Since the robot motion is produced by programmed commands, the acceleration can be taken directly from the motion commands (as the author did in the experiments). The errors in the proportional integral derivative (PID) motion control loop of the robot are anticipated to be much smaller than twice differentiating position or wheel encoder measurements or using accelerometers.

4. The two experimental justifications of the acceleration odometry model also provide two methods for calibrating a robot and floor combination. Optimisation of the map quality can be applied to existing environments, and can provide the bonus of the calibration of laser extrinsic parameters.
5. The author anticipates that the four calibration parameters of the odometry will depend on the floor surface type like research findings in Chapter 3. It is assumed that the robot only has linear motion and on the spot turns in this chapter.

In the next chapter, the restriction of straight paths of this chapter is lifted in the investigation of odometry changes due to acceleration.

Odometry Model for Curved Paths

5.1 Introduction

The odometry model in Chapter 4 only incorporates robot accelerations, where the robot travels in a straight line. However, a wheeled robot can maneuver in more complex trajectories. This motivates the study of curved paths that affect odometry accuracy due to centrifugal forces and scrubbing of tyres on the floor as robots twist. An aim of this chapter is to investigate the relationship between wheel radii and wheel separation distance B and centrifugal forces on the robot. From this investigation a new odometry model is proposed.

The research in this chapter is applicable to differential drive wheeled mobile robots with pneumatic tyres travelling on flat and smooth surfaces. This has been chosen to simplify the problem and also because these conditions apply to many popular robot applications. This chapter proceeds as follows. Section 5.1 and Section 5.2 introduce the research aims and provide a literature review respectively. A theoretical analysis of the linear relationship between wheel separation distance and the square of robot speed is introduced in Section 5.3. The analysis is consistent with experimental results in Section 5.4. However, the estimates of B with very slow wheel angular velocities (e.g., radius of curvature near $B/2$, half of the wheel separation) did not match our model in Section 5.3. This motivated further investigation of the wheel radius at slow wheel angular velocity in Section 5.5 where a new phenomenon is observed. The effective wheel radius increases as wheel angular velocities become small. This effective wheel radius effect is modelled with a saturation growth curve in Section 5.5. The saturation

growth model is applied to a new dataset for a robot running on circular paths to justify the proposed linear relationship between wheel separation and square of robot speed in Section 5.6. Section 5.7 justifies the two models developed in Section 5.3 and Section 5.5 on a soft carpet with real experiments. Section 5.8 proposes a 5 parameter model based on the results of previous sections. This model is tested on a varied set of robot paths. Finally, a summary of this chapter is given in Section 5.9.

5.2 Literature Review of Odometry with Slip

Gonzalez et al. [89] developed a closed-loop controller for off-road robots to improve visual odometry accuracy by compensating longitudinal robot slip. The outdoor tracked robot was equipped two commercial level cameras and driven on gravel terrain for circular and U-shape trajectories for experiments. Their results showed some improvement of localisation accuracy despite a small mean slip of 5%. They expected their controller to show higher improvements on terrains with higher slip values. However, their controller was designed to include longitudinal slip only without considering lateral slip effects.

In the research field of vehicle motion control, some other researchers have developed systems to allow wheel slippage in both longitudinal and lateral directions. Tian and Sarkar [90] developed regulation and turning control techniques to deal with the underactuated nonlinear dynamic system, where both lateral and longitudinal slips were included for a wheeled mobile robot. They implemented control methods in simulations to verify the proposed algorithms aiming to better maneuver a mobile robot on slippery surfaces. These two directions of slippage were also considered in [91], where Khan et al. developed a logic-based control scheme for a wheeled mobile robot, which could successfully follow a designed circular trajectory in the presence of wheel slippage. Low and Wang [92] designed a control scheme using real-time kinematic (RTK)-GPS and other aiding sensors to measure the posture, velocities, and perturbations of an outdoor wheeled mobile

robot in the presence of wheel skidding and slipping. They related the lateral velocity of the robot with the rear slip angle via geometric relations. They improved the path following accuracy with RTK-GPS, which was normally not available for indoor robots.

Some other researchers focus on the effects between the lateral force and wheel slip angles. A simplified bicycle model of a robot was used to derive an explicit lateral force control to minimise the force error in simulation environment with different stiffness of floors in [93]. Fauroux and Vaslin [94] increased the steering efficiency of an all-road 6×6 electric wheelchair by analysing lateral and normal forces on wheels with minor adjusting of the vehicle suspension.

In the field of robot formation control, Tian and Sarkar [95] developed a sensitive-controller that could stabilize the formation of wheeled mobile robots even when slip was significant. They declared that was the first paper to consider wheel slip in formation control. However, they only justified their method using simulation without real robot experiments.

The wheel slip mentioned in the literature review above is predominantly incorporated in outdoor mobile robot control systems. In contrast, the research in this chapter focuses on indoor wheeled mobile robot odometry calibration with a novel model for wheel radius and wheel angular velocity. Some readers may wonder about the existence and importance of wheel slippage for indoor mobile robots. In this chapter, the author theoretically and experimentally investigates this issue and finds there can be up to a 9.2% estimation improvement for the wheel radius under common operating conditions of the Pioneer 2 robot. In the end, a five parameter odometry model that augments the conventional three parameters odometry model is developed and justified with real experiments.

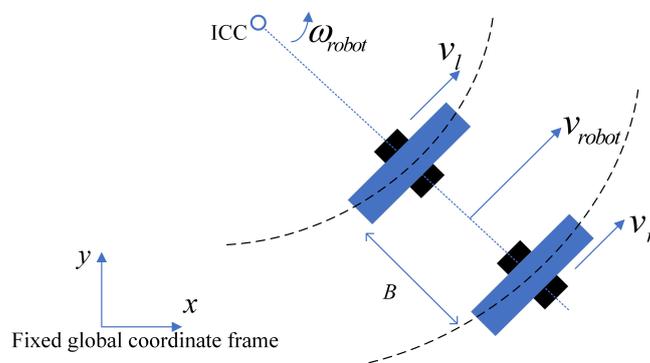


Figure 5.1: Instantaneous centre of curvature (ICC) of a differential drive robot. Adapted from [7].

5.3 Theoretical Investigation

5.3.1 Radius of Curvature

For a wheeled mobile robot to exhibit curved rolling motion there exists a point around which each wheel on the vehicle follows a circular course [7]. This point is called the instantaneous centre of curvature (ICC) or the instantaneous centre of rotation (ICR). Figure 5.1 shows the ICC of a differential drive robot consisting of two wheels mounted on a common axis controlled by separate motors. In Figure 5.1, v_l and v_r are the left and the right wheel velocities and v_{robot} and ω_{robot} are robot velocity and robot angular velocity all with respect to a global coordinate system. At any instant in time, we can derive the equations from Equation (5.1) to Equation (5.13) when denoting radius of curvature as R .

$$v_l = \left(R - \frac{B}{2}\right)\omega_{robot} \quad (5.1)$$

$$v_r = \left(R + \frac{B}{2}\right)\omega_{robot} \quad (5.2)$$

By solving Equation (5.1-5.2) for R , we get Equation (5.3). The curvature of the trajectory is $\kappa = 1/R$. There are several special cases summarised in Table 5.1 in terms of the robot trajectory.

Table 5.1: Relationship between Radius of Curvature of Robot Paths and Wheel Velocities without Slippage.

| Wheel velocities (m/s) | Radius of curvature (m) | Robot motion |
|------------------------------------|-------------------------|--|
| $v_l = v_r$ | $R = \infty$ | Travel straight or stationary. |
| $v_l = -v_r$ | $R = 0$ | Rotate about a point midway between two wheels. |
| $v_l = 0, v_r > 0$ | $R = \frac{B}{2}$ | Rotate counter clockwise with ICC at the centre of left wheel. |
| $v_l > 0, v_r = 0$ | $R = -\frac{B}{2}$ | Rotate clockwise with ICC at the centre of right wheel. |
| $v_l < 0, v_r > 0, v_l \neq -v_r$ | $0 < R < \frac{B}{2}$ | Rotate counter clockwise with ICC between the middle of two wheels and the centre of left wheel. |
| $v_l > 0, v_r < 0, v_l \neq -v_r$ | $-\frac{B}{2} < R < 0$ | Rotate clockwise with ICC between the middle of two wheels and the centre of right wheel. |
| $v_l \times v_r > 0, v_l \neq v_r$ | $ R > \frac{B}{2}$ | Travel forward or backward with ICC outside the robot. |

$$R = \frac{B (v_r + v_l)}{2 (v_r - v_l)} \quad (5.3)$$

5.3.2 Effective B due to Tyre Vertical Elastic Displacement

This section investigates the effect of vertical elastic deformation of wheel tyres due to centrifugal forces on a robot. In this analysis, it is assumed that both wheels of the robot remain on the ground. The following analysis is primarily based on a left turning case. For simplification the trajectory is assumed to have an instantaneous radius of curvature greater than $B/2$. Figure 5.2 shows the forces of interest. The notation is defined here:

R : radius of curvature of robot path.

v : robot speed.

g : gravitational acceleration.

m : mass of the robot.

F_c : centrifugal force caused by the left turning.

h : height of the robot centre of mass.

T: middle point between the contact points of two wheels.

F_{gL} and F_{gR} : supporting force upon the left wheel and the right wheel from the floor respectively.

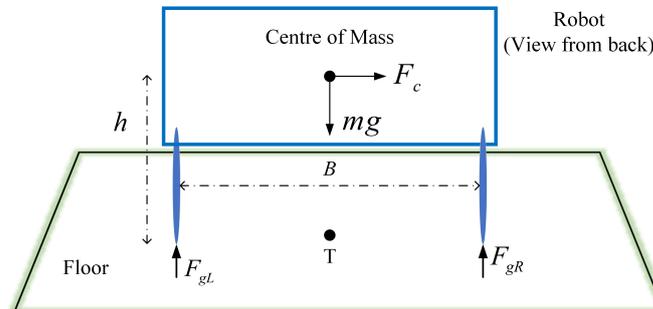


Figure 5.2: Schematic graph of the forces of interest.

It is assumed that the camber angles of both driving wheels are zero (i.e., wheels remain in a vertical plane) and that the lateral forces on each tyre are equal, resulting in no change in B due to lateral forces. This is reasonable since both tyres are physically constrained to be B apart and have similar lateral stiffness. This may naturally generate the same lateral forces when displacing the same amount during a turn. The analysis assumes very little lateral slip occurs. The change in radius of each wheel is modelled, which is caused by uneven distribution of load when cornering.

In Figure 5.2, weight of the robot is balanced by the ground opposing forces through tyres. Hence we have Equation (5.4).

$$mg = F_{gL} + F_{gR} \quad (5.4)$$

The total torque about the point T must be 0. This gives Equation (5.5).

$$F_{gR} \frac{B}{2} - F_{gL} \frac{B}{2} - \frac{mv^2}{R} h = 0 \quad (5.5)$$

Solving Equation (5.4) and Equation (5.5)

$$\begin{cases} F_{gL} = \frac{mg}{2} - \frac{mv^2 h}{RB} \\ F_{gR} = \frac{mg}{2} + \frac{mv^2 h}{RB} \end{cases} \quad (5.6)$$

It is assumed that both tyres have a vertical stiffness k_v . So we have from Equation (5.6)

$$\begin{cases} R_L = R_{L0} + \frac{1}{k_v} \left(\frac{mg}{2} - F_{gL} \right) = R_{L0} + \frac{mv^2 h}{RBk_v} \\ R_R = R_{R0} - \frac{1}{k_v} \left(F_{gR} - \frac{mg}{2} \right) = R_{R0} - \frac{mv^2 h}{RBk_v} \end{cases} \quad (5.7)$$

where R_{L0} and R_{R0} are the wheel radii when the robot is travelling on a straight line path. For a robot turning to the left by a small angle $d\theta$, the travelling distances of left and right wheels are calculated as d_L and d_R in Equation (5.8).

Thus the rotating angles of left and right wheels, σ_L and σ_R , during the $d\theta$ robot heading change can be calculated in Equation (5.9).

$$\begin{cases} d_L = (R - \frac{B}{2})d\theta \\ d_R = (R + \frac{B}{2})d\theta \end{cases} \quad (5.8)$$

$$\begin{cases} \sigma_L \cong \frac{d_L}{R_L} = \frac{(R - \frac{B}{2}) d\theta}{R_{L0} + \frac{mv^2h}{RBk_v}} \\ \sigma_R \cong \frac{d_R}{R_R} = \frac{(R + \frac{B}{2}) d\theta}{R_{R0} - \frac{mv^2h}{RBk_v}} \end{cases} \quad (5.9)$$

We can model this change in terms of changes in B . The author defines B_{eff} as the new value of B that produces the same robot turn angle of $d\theta$ when assuming the radii are still R_{L0} and R_{R0} . Thus we get Equation (5.10) and Equation (5.11).

$$d\theta = \frac{d_R - d_L}{B_{eff}} \quad (5.10)$$

$$\begin{cases} d_L \cong \sigma_L R_{L0} \\ d_R \cong \sigma_R R_{R0} \end{cases} \quad (5.11)$$

Therefore, B_{eff} can be calculated as in Equation (5.13). Denote $\Gamma = \frac{mv^2h}{RBR_0k_v}$, assuming $R_0 = R_{L0} = R_{R0}$.

$$\begin{aligned}
 B_{eff} &\cong \frac{d_R - d_L}{d\theta} \\
 &= \frac{\frac{(R + \frac{B}{2}) d\theta}{mv^2h} R_{R0} - \frac{(R - \frac{B}{2}) d\theta}{mv^2h} R_{L0}}{R_{R0} - \frac{mv^2h}{RBk_v} - \left(R_{L0} + \frac{mv^2h}{RBk_v} \right)} \\
 &= \frac{\frac{(R + \frac{B}{2})}{1 - \Gamma} - \frac{(R - \frac{B}{2})}{1 + \Gamma}}{1 - \Gamma^2} \\
 &= \frac{(R + \frac{B}{2})(1 + \Gamma) - (R - \frac{B}{2})(1 - \Gamma)}{1 - \Gamma^2}
 \end{aligned} \tag{5.12}$$

for small Γ

$$\begin{aligned}
 B_{eff} &\cong \left(R + \frac{B}{2} \right) (1 + \Gamma) - \left(R - \frac{B}{2} \right) (1 - \Gamma) \\
 &= B + 2R\Gamma \\
 &= B + 2R \frac{mv^2h}{RBR_0k_v} \\
 &= B + \left(\frac{2mh}{BR_0k_v} \right) v^2
 \end{aligned} \tag{5.13}$$

where the unitless Γ is assumed to be very small. Let us use an example calculation to get a sense of the result above. Tyre vertical stiffness k_v [96, 67] is assumed to be 100000 N/m, and $m = 20$ kg, $v = 0.1$ m/s, $h = 0.2$ m, $R = 1$ m, $B = 0.36$ m, $R_0 = 0.1$ m for a robot. $\Gamma^2 = 1.2e^{-10} \ll 1$. Equation (5.13) means the effective wheel separation increases approximately proportional to the square of robot speed if radius of curvature of trajectories is larger than $B/2$, i.e., both wheels rotate in the same direction (clockwise or counter-clockwise).

5.4 Linear Dependency of B with the Square of Robot Speed

In this section, the author experimentally investigates the model in Subsection 5.3.2. Figure 5.3 is a picture of the robot and its surroundings. The floor is cleaned with a vacuum cleaner before every experiment, because the robot may run over grit or dust resulting in significant non-systematic odometry errors. The odometry model used in this section is the conventional three parameter odometry model including R_l , R_r and B , since the robot speed is kept constant and accelerations modelled in the previous chapter can be ignored.

To get an estimate of R_l and R_r when there is little longitudinal and centrifugal accelerations under slow movements, a PSO based odometry calibration was performed when the robot speed was below 0.05 m/s. Afterwards R_l and R_r were kept constant and only B was optimised with PSO based optimisation for other datasets, where the robot travels steadily on different circular paths. The baseline estimate of R_l and R_r is designed to reduce the correlation effect among these three odometry parameters when optimising B . Calibrated Laser Range Finders (LRFs) extrinsic parameters in Chapter 3 were adopted, since the set up of LRFs was the same. The time synchronisation technique between encoders and LRFs as in Chapter 4 was performed. These datasets were collected on September 27, 2017. This is noted, since changes in temperature and air pressure can affect pneumatic tyre calibration results.

Figure 5.4 shows the calibration results for wheel separation distance B . The 2D view of relevant calibration results of wheel separation is shown in Figure 5.5 to remove the ambiguity of viewing 3D data on a 2D page.

The experimental results are discussed under two different robot working conditions. The first one is for results where the robot travels with radius of curvature greater than $B/2$ as assumed in the model analysis in Subsection 5.3.2. Figure 5.6 shows a trend that is consistent with the analysis, i.e., the effective wheel separation increases linearly with respect to the square of robot speed. However,

5.4. LINEAR DEPENDENCY OF B WITH THE SQUARE OF ROBOT SPEED



Figure 5.3: Robot and its surrounding environment in the dataset collected on September 27, 2017.

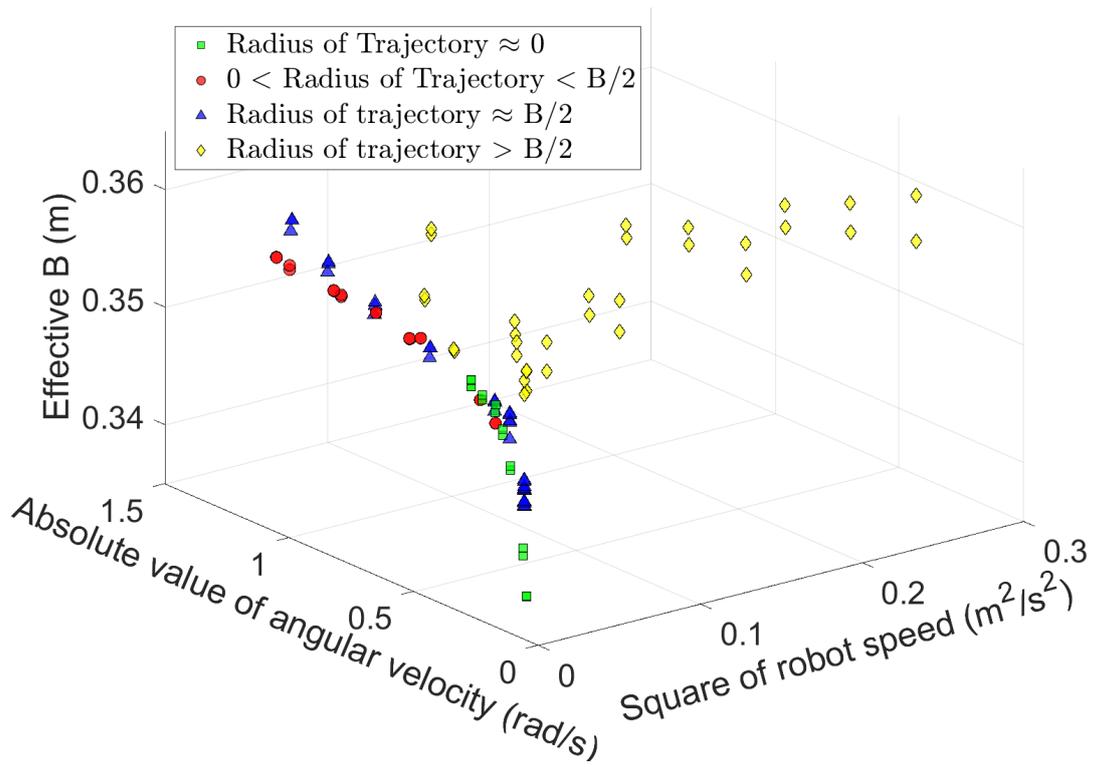
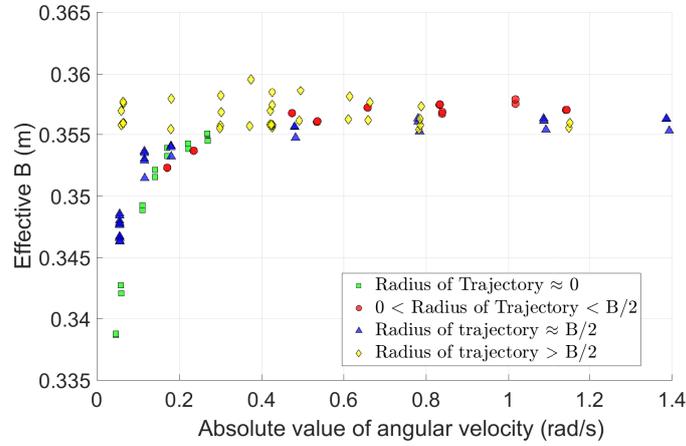
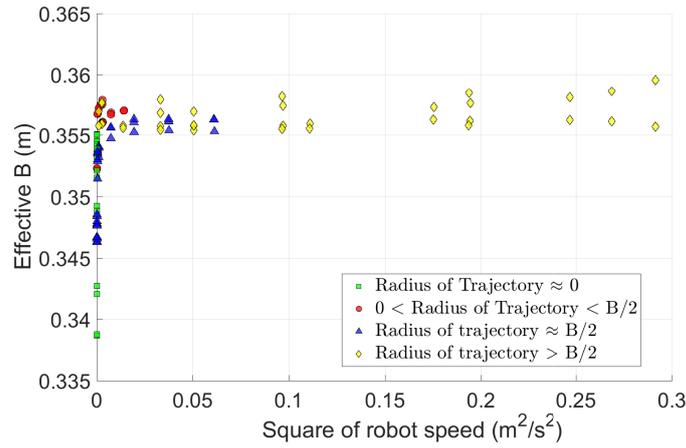


Figure 5.4: Calibration results of wheel separation distance B for datasets on September 27, 2017.

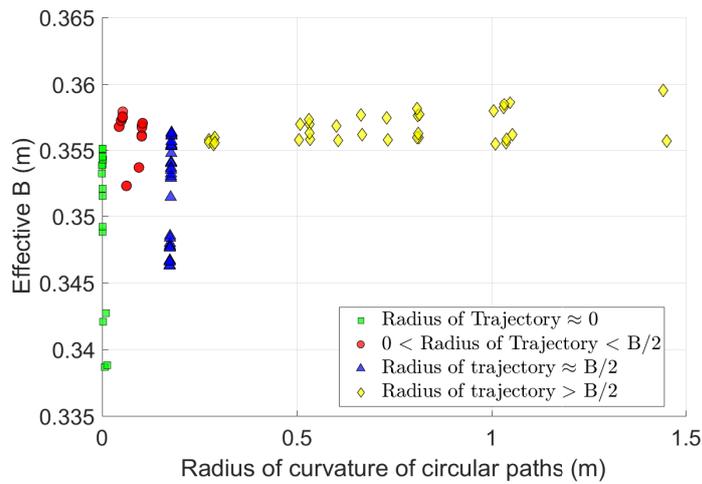
5.4. LINEAR DEPENDENCY OF B WITH THE SQUARE OF ROBOT SPEED



(a)



(b)



(c)

Figure 5.5: 2D view of estimated wheel separation distances B for datasets on September 27, 2017. (a) B vs. ω . (b) B vs. v^2 . (c) B vs. Radius of Curvature

5.4. LINEAR DEPENDENCY OF B WITH THE SQUARE OF ROBOT SPEED

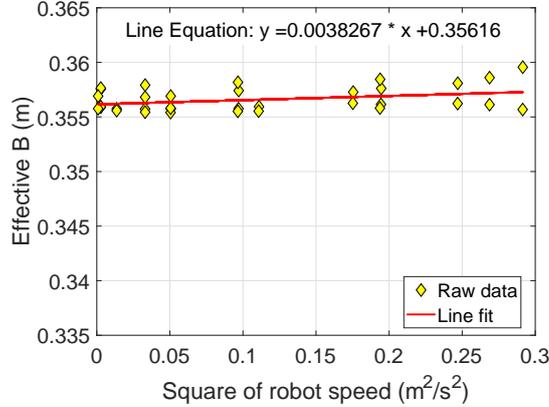


Figure 5.6: Calibration results of effective wheel separation for radius of curvature greater than $B/2$ cases of datasets on September 27, 2017.

this is not significant statistically nor are the effects pronounced. Hence this effect will not be included in the odometry model in Section 5.8. This effect may be more significant if the robot travels at higher speeds and angular velocities. The author leaves this investigation to the future work.

The physical wheel separation distance is measured using a ruler as shown in Figure 5.7. The values of the estimated B are within the measured range $[0.33, 0.38]$ m. The second group of data corresponds to the cases where the robot travels with radius of curvature less than $B/2$. From Figure 5.5 (b), we can see that it is not clear what the relationship between effective wheel separation and the square of robot speed is for very slow movements. Considering the fact that wheel separation distance is correlated with wheel radii, the author wondered whether wheel radii may change with respect to different wheel velocities. This leads to a further investigation of the relationship between effective wheel radius and wheel angular velocity in the next section.

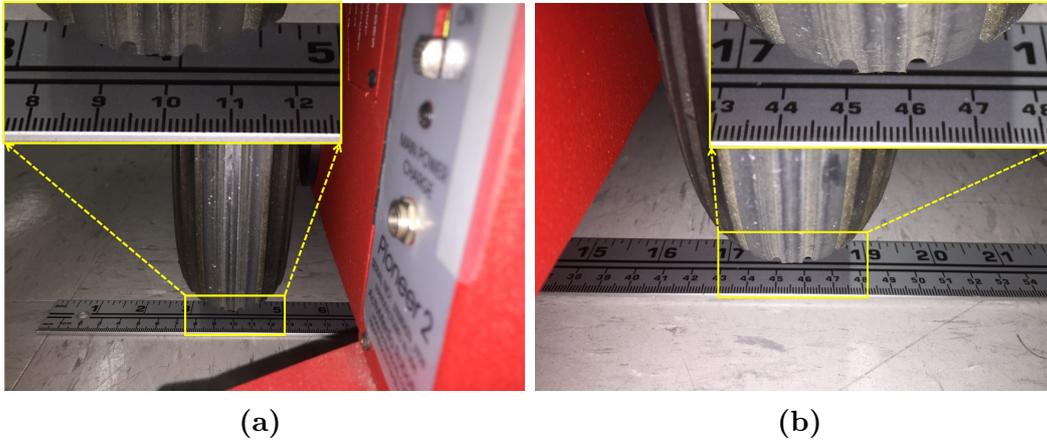


Figure 5.7: Estimate B using a ruler. (a) Left wheel. (b) Right wheel.

5.5 Model of Wheel Radius and Wheel Angular Velocity

5.5.1 Nonlinear Regression with Saturation Growth Model

The testing environment for this section and Section 5.6 is shown in Figure 5.8. These datasets were collected on October 13, 2017. The robot was controlled to steadily accelerate before maintaining a constant speed for about 2.5 meters. Then it steadily decelerated to a stop and rotated around 90 degrees. Afterwards, the robot repeated this pattern but ran at different speeds. One example plot of robot speeds is shown in Figure 5.9. The laser scans were sampled for mapping and calibration according to the distance travelled by the robot. For this experiment, a laser scan after every 0.1 m was utilised for future processing, including building a map. PSO cannot always guarantee to find the global optimal point of a cost function, in this case, the maximum of MI map quality metric introduced in Chapter 3. Therefore the PSO based odometry calibration for each straight trajectory with constant speeds was repeated 10 times. 36 particles and Von-Neumann neighbourhood topology were chosen for the PSO. Each optimisation runs for 100 generations, i.e., 3600 function evaluations of the OG maps. The map is set to have a resolution of 5 mm square. Odometry parameters corresponding to the highest map quality test are selected.

5.5. MODEL OF WHEEL RADIUS AND WHEEL ANGULAR VELOCITY

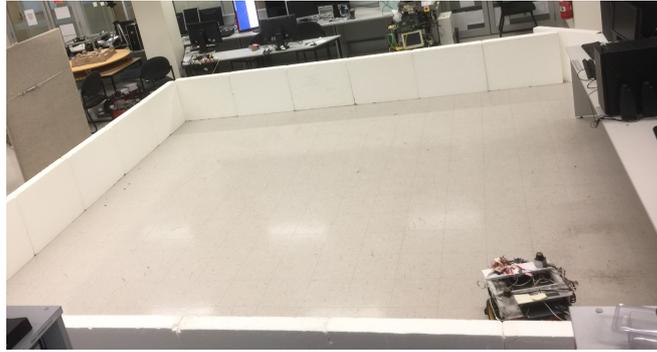


Figure 5.8: Robot and its surrounding environment in Subsection 5.5.1 and Section 5.6.

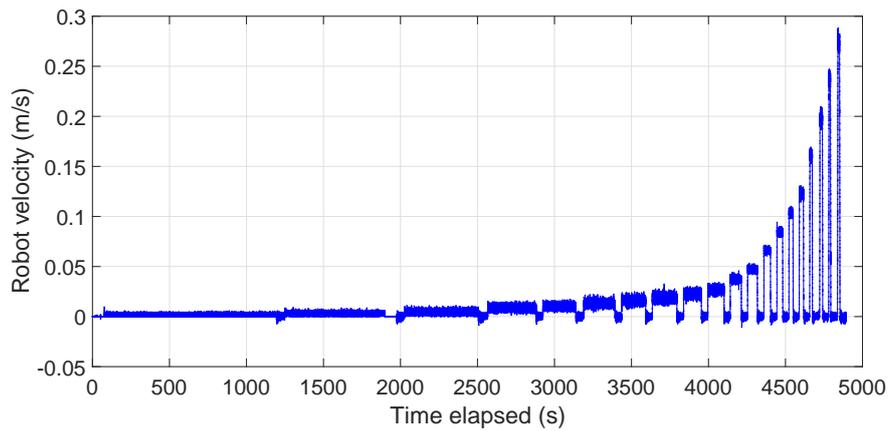


Figure 5.9: Robot speeds on straight paths.

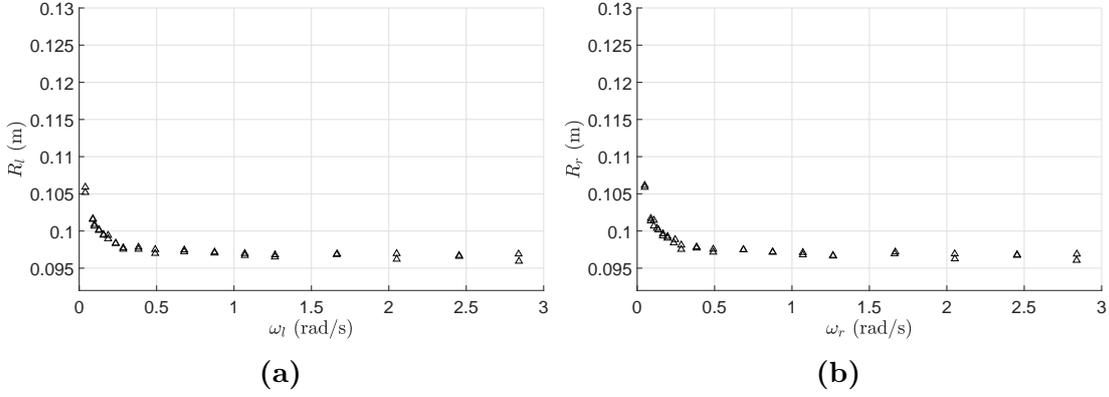


Figure 5.10: Wheel radii variation with respect to wheel angular velocities. (a) Left wheel. (b) Right wheel.

The results of experiments are shown in Figure 5.10. Interestingly, there is an unanticipated reduction of wheel radius at slow speeds. The author’s aim is to find a model for the wheel radius given the wheel angular velocity, which can fit the data in Figure 5.10. Afterwards, this model can be applied to predict the wheel radius using the direct measurement of wheel angular velocity from encoder data in the real time. Nonlinear regression is used here.

One of the commonly used error metric of the differences between values predicted by a model and the values actually observed is root-mean-square deviation (RMSD) [97]. Equation (5.14) is the RMSD formula of predicted values \hat{y}_i for observations i of a regression’s dependent variable y_i over n samples.

$$RMSD = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (5.14)$$

Two candidate models, namely exponential decay in Equation (5.15) and saturation growth in Equation (5.16) were tested after visually inspecting the data in Figure 5.10. Both models have three independent parameters or degrees of freedom. The parameters are intuitively explained below. An absolute sign is applied to wheel angular velocity ω in both models assuming wheel radii follow

the same model when wheels rotate either clockwise or counter clockwise.

$$r_e(\omega) = R_{e\infty} + (R_{e0} - R_{e\infty}) \times e^{-\frac{|\omega|}{\tau_e}} \quad (5.15)$$

$$r_s(\omega) = R_{s\infty} + \frac{R_{s0} - R_{s\infty}}{\frac{|\omega|}{\tau_s} + 1} \quad (5.16)$$

- R_{e0} , R_{s0} : wheel radius when a robot wheel does not rotate, $\omega = 0$ rad/s.
- $R_{e\infty}$, $R_{s\infty}$: wheel radius when a robot wheel rotates at high speed, $\omega \rightarrow \infty$ rad/s. These variables represent the commonly recognised wheel radius parameter in terms of odometry calibration.
- τ_e : exponential decay constant with units of rad/s.
- τ_s : wheel angular speed when wheel radius is equal to $(R_{s0} + R_{s\infty})/2$.

Using the *patternsearch* function in MATLAB [98], the author constructs the models and minimise RMSD of the regression of data in Figure 5.10. The results are shown in Figure 5.11 and in Table 5.2. The mean absolute error (MAE) is also reported. It is clear that the saturation growth model fits the data better. Therefore, the saturation growth model is selected. During the experiments, it was clear that the robot wheels cannot rotate continuously and smoothly for wheel angular velocities below 0.035 rad/s. This appears to be an issue related to friction and minimum motor torque. Therefore speeds greater than 0.035 rad/s were used in experiments.

5.5. MODEL OF WHEEL RADIUS AND WHEEL ANGULAR VELOCITY

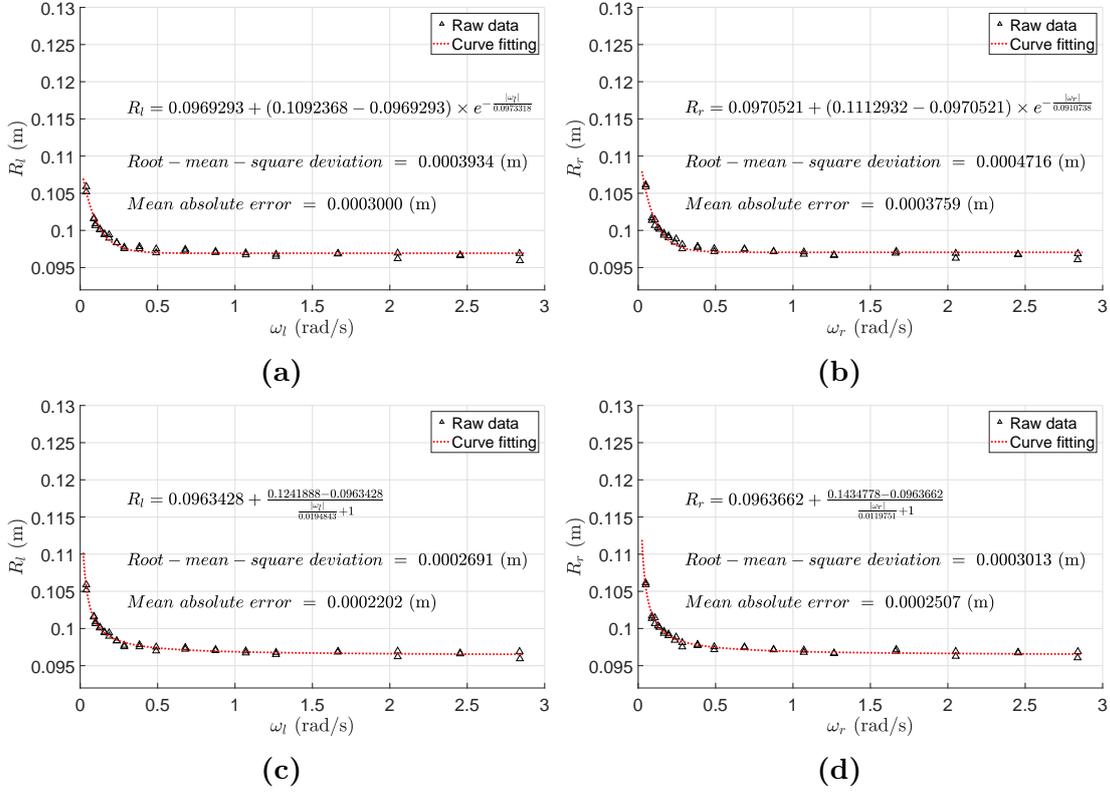


Figure 5.11: Curve fitting of wheel radii with respect to wheel angular velocities. (a) Left wheel using exponential model fitting. (b) Right wheel using exponential model fitting. (c) Left wheel using saturation growth model fitting. (d) Right wheel using saturation growth model fitting.

Table 5.2: Curve Fitting Results (RMSD and MAE) of Wheel Radii and Angular Velocities.

| | Left wheel | Right wheel |
|--|------------|-------------|
| Exponential model RMSD (mm) | 0.393 | 0.472 |
| saturation growth model RMSD (mm) | 0.269 | 0.301 |
| Exponential model MAE (mm) | 0.300 | 0.376 |
| saturation growth model MAE (mm) | 0.220 | 0.251 |

5.5.2 Model Verification with Independent Datasets

To investigate the accuracy of the estimation of wheel radii using the method in Subsection 5.5.1, the author performed the same experimental method upon different datasets collected on October 8, 2017. The testing environment is shown in Figure 5.12. The robot is commanded to travel straight multiple times on Lino. Robot velocities are depicted in Figure 5.13. The author repeated 10 times (except the last group which had 9 repeating paths with the same speed) for the same robot speeds of the last 49 straight paths as shown in Figure 5.13. Figure 5.14 shows the normal fitting results of those five repeating experiments with error bars of ± 2 standard deviation in both axes. These results confirm the finding in Subsection 5.5.1 is valid with very small errors. Therefore it is possible to apply the model to predict the instantaneous wheel radius of any wheel angular velocities. The predicted results are more accurate than the commonly assumed constant wheel radius value in the literature. It is possible to numerically calculate the relative wheel radius difference ratio δR_{ratio} for $\omega = 3$ rad/s and $\omega = 0.05$ rad/s. Use the saturation growth model for the right wheel as shown in Figure 5.11 (d), we have

$$\delta R_{ratio} = \frac{g(0.05) - g(3)}{g(3)} \times 100 \cong 9.2\% \quad (5.17)$$

$g(3)$ is used as the denominator instead of $g(0.05)$, because a wheel angular velocity of 3 rad/s is closer to normal working speed of an indoor robot. This 9.2% increase of wheel radius has not been mentioned in the literature to the author's knowledge despite its importance in odometry calibration.

Some readers may now question the validity of Figure 5.6, considering that R_l and R_r are set to be constant instead of applying the saturation growth model proposed in this subsection. However, we can see that the curve fitting results in Figure 5.11 indicate that wheel radii are varying significantly only at wheel angular velocities less than 0.5 rad/s, i.e., approximately $0.0025 \text{ m}^2/\text{s}^2$ in terms of square of robot speeds. These slow movement data correspond to the first 4

5.5. MODEL OF WHEEL RADIUS AND WHEEL ANGULAR VELOCITY

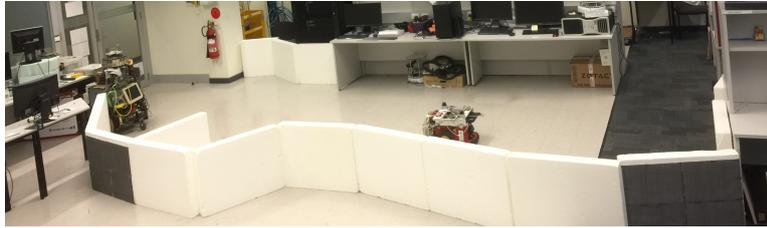


Figure 5.12: Robot and its surrounding environment in Subsection 5.5.2.

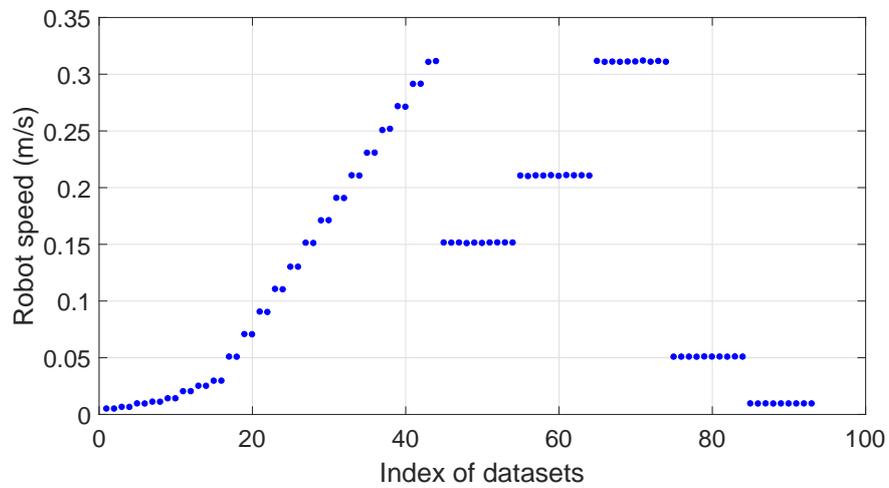


Figure 5.13: Robot speeds of each straight paths of datasets in Subsection 5.5.2.

data points in Figure 5.6. Therefore, the slope of the line fit in Figure 5.6 will be almost unchanged.

5.5. MODEL OF WHEEL RADIUS AND WHEEL ANGULAR VELOCITY

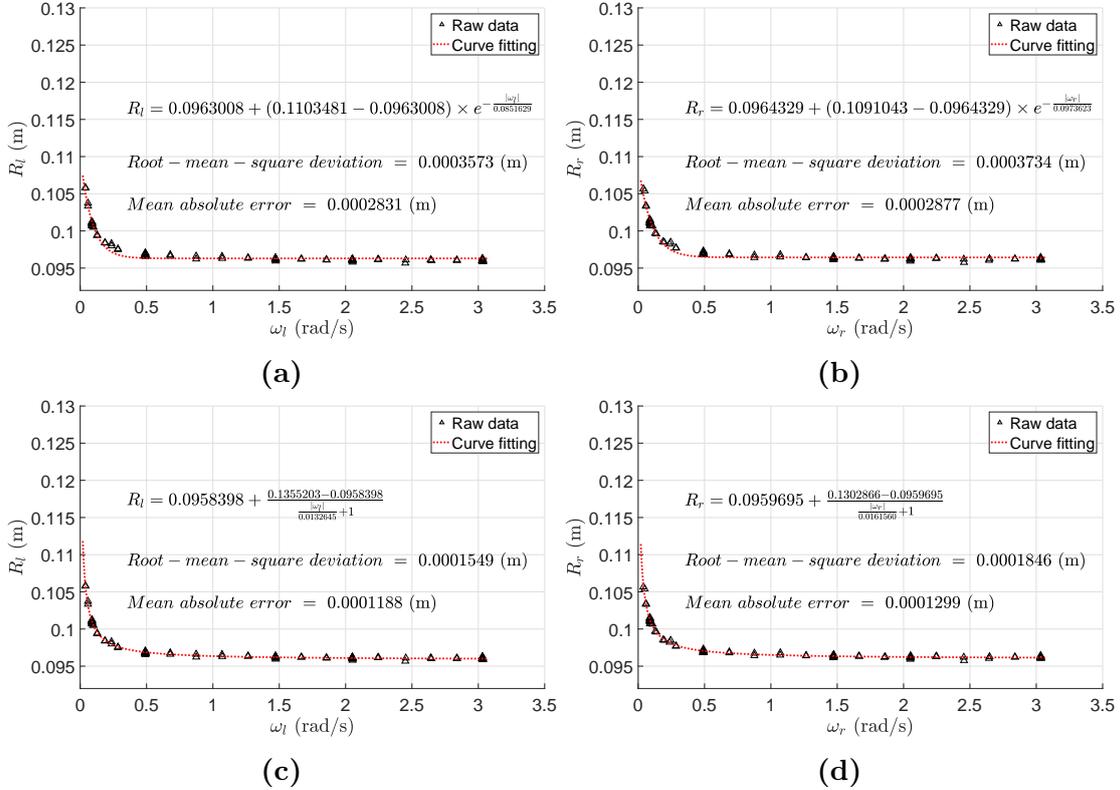


Figure 5.14: Curve fitting of wheel radii with respect to wheel angular velocities of datasets in Subsection 5.5.2. (a) Left wheel using exponential model fitting. (b) Right wheel using exponential model fitting. (c) Left wheel using saturation growth model fitting. (d) Right wheel using saturation growth model fitting.

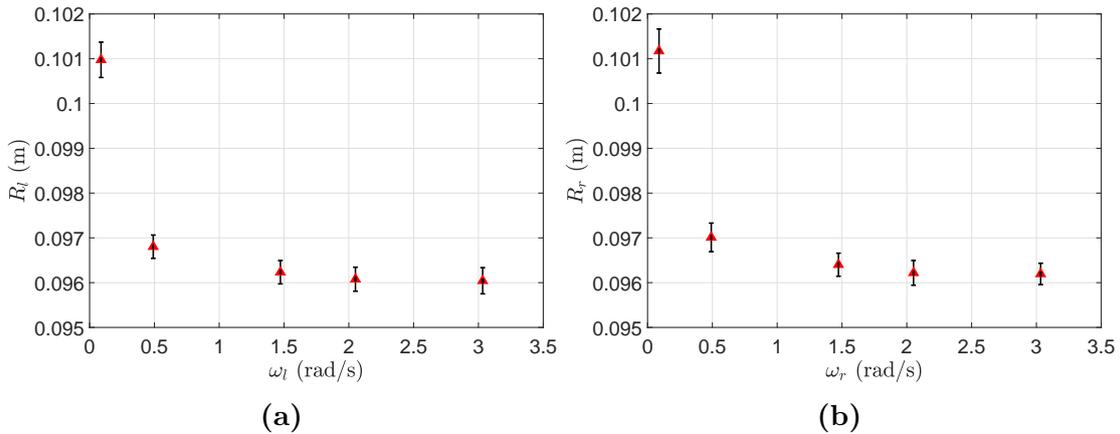


Figure 5.15: Normal fitting results of five repeating experiments in Figure 5.14 with error bars of ± 2 standard deviation in both axes. (a) Left wheel. (b) Right wheel.

5.6 Calibration of Wheel Separation on Circular Paths

The saturation growth model in Subsection 5.5.1 is applied to estimate wheel radii for all the rotation experiments in this section. A robot will have a constant wheel separation distance if it turns on circular paths with a constant angular velocity and a constant speed on a certain floor. 74 independent datasets including different circular paths were collected on October 13, 2017. Figure 5.16 shows the calibration results of wheel separation when using the saturation growth model for wheel radii and wheel angular velocities.

Figure 5.16 shows the calibration results of wheel separation distances when applying the saturation growth model built in Figure 5.11 (c) and (d). The saturation growth model helps to better predict the wheel separation distances especially for slow wheel rotation cases, i.e., those effective wheel separation distances below 0.355 m in Figure 5.5 were compensated and became closer to other data points in Figure 5.16. This represents a continuous change of B over various robot speeds and angular velocities, which is considered to be natural and more reasonable. From these results, we can see the following trends.

- Radius of Trajectory = 0: wheel separation distances spread more widely than the other three cases. This may be due to the scrubbing of tyres on the floor bringing more non-systematic errors of the estimation.
- $0 < \text{Radius of Trajectory} < B/2$: These cases tend to have larger effective B values in Figure 5.16 (c). The reason behind is left for future research.
- Radius of Trajectory = $B/2$: These cases tend to have smaller effective B values in Figure 5.16 (c). The reason behind is also left for future research.
- Radius of Trajectory $> B/2$: These cases correspond to the assumption of the model developed in Subsection 5.3.2, i.e., both robot wheels rotate in

the same direction, forward or backwards. Effective B values appear to be positively proportional to the square of robot speeds as the analysis in Section 5.3. Figure 5.17 shows the line fit result of this linear relationship.

From Figure 5.17 (b) we have

$$\hat{B} = f(v^2) = 0.0046465v^2 + 0.35603 \quad (5.18)$$

We can calculate how much this model can improve the estimation of B . Consider B estimation relative ratio δB_{ratio} for $v^2 = 0 \text{ m}^2/\text{s}^2$ and $v^2 = 0.3 \text{ m}^2/\text{s}^2$. We have

$$\delta B_{ratio} = \frac{f(0.3) - f(0)}{f(0)} \times 100 \cong 0.39\% \quad (5.19)$$

This means that our linear model of wheel separation and square of speed improved the estimation of B by 0.39% for these datasets.

5.6. CALIBRATION OF WHEEL SEPARATION ON CIRCULAR PATHS

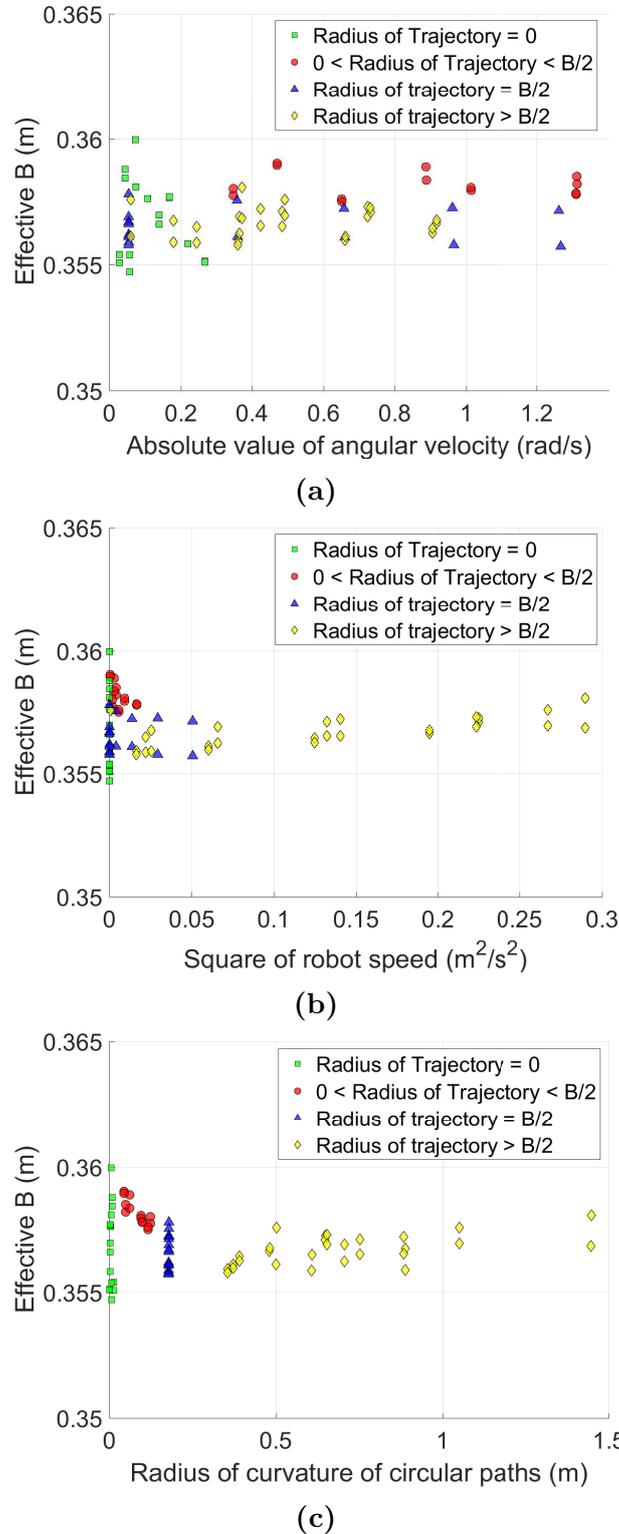


Figure 5.16: Calibration results of wheel separation distances when applying the saturation growth model built in Figure 5.11 (c) and (d). (a) B vs. ω . (b) B vs. v^2 . (c) B vs. Radius of Curvature.

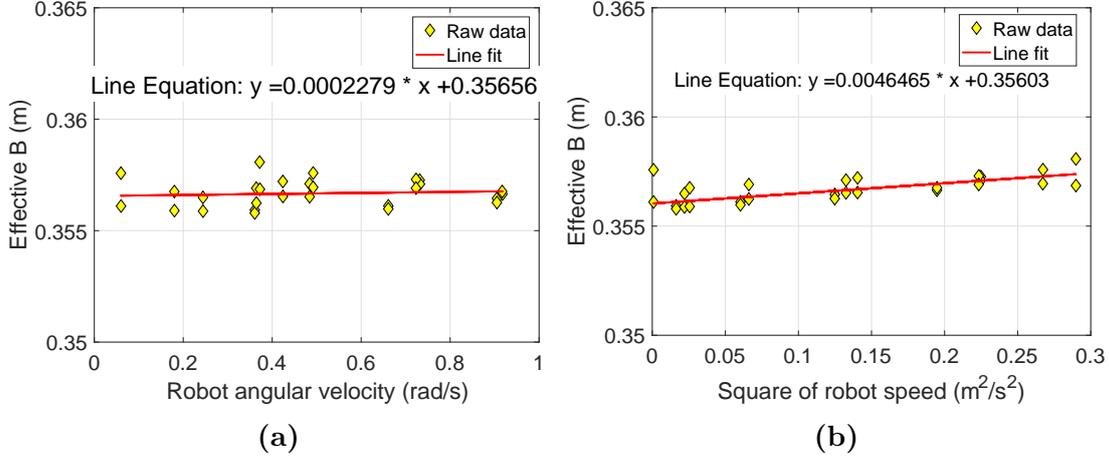


Figure 5.17: Partial calibration results of B on a lino floor when using saturation growth model for wheel radii and wheel angular velocities under the assumption in Subsection 5.3.2. (a) B vs. ω . (b) B vs. v^2 .

5.7 Experiments on a Carpet

Changes in B are investigated on a carpet, which is softer and more uneven compared to Lino. The testing environment is shown in Figure 5.19. The robot velocities during the straight running experiments are shown in Figure 5.18.

Firstly, the exponential decay curve fitting and saturation growth curve fitting are applied to wheel radii and corresponding wheel angular velocities as in Figure 5.20. It indicates that the saturation growth model can fit better than the exponential decay model. Secondly, the raw data of both R_l and R_r show clear differences for the same wheel angular velocities. The corresponding R_l and R_r pairs are estimated from opposite travelling directions on the carpet. The directional carpet pile may have caused these differences. This pattern related phenomenon is left for future research. No research papers are known to address directional odometry properties on carpets.

Figure 5.21 and Figure 5.22 show the 3D and 2D views of the calibration results of effective wheel radii on the carpet after applying saturation growth model

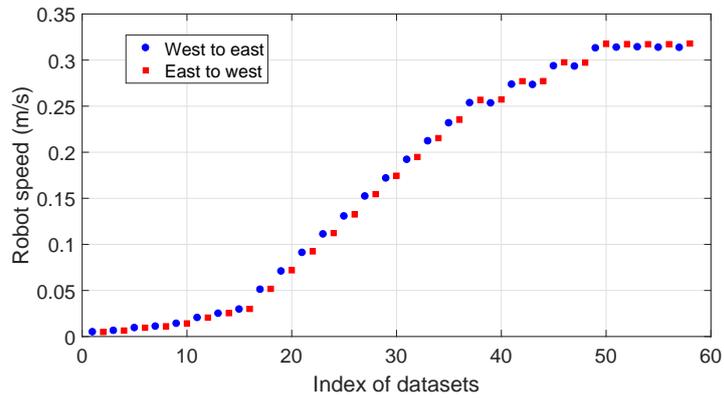


Figure 5.18: Robot speeds during straight running tests.

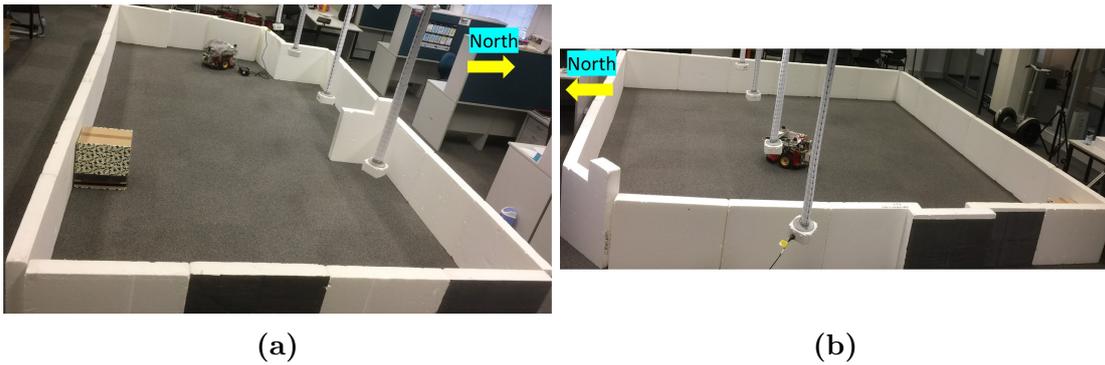


Figure 5.19: Robot and its surrounding environment in Section 5.7 (a) Straight running paths. (b) Circular running paths.

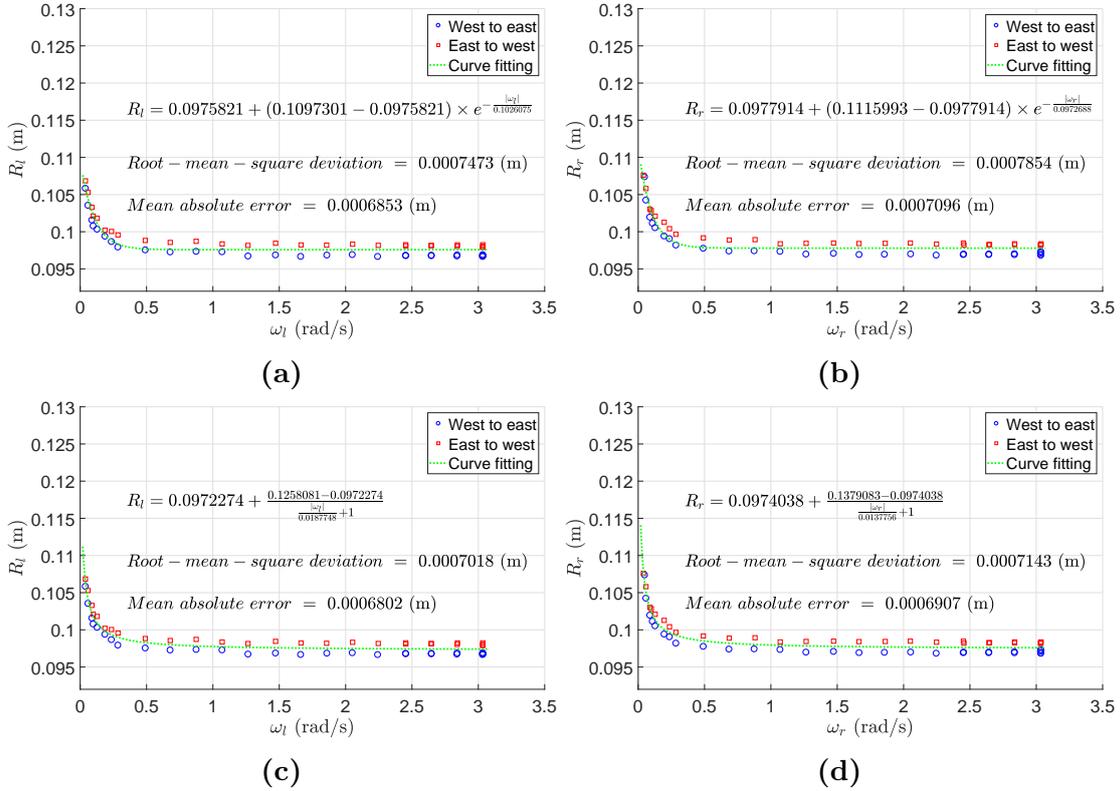


Figure 5.20: Curve fitting of wheel radii with respect to wheel angular velocities for datasets on Carpet 2. Note that the travel direction affects wheel radii due to the directional surface of the carpet or floor inclination. (a) Left wheel using exponential model fitting. (b) Right wheel using exponential model fitting. (c) Left wheel using saturation growth model fitting. (d) Right wheel using saturation growth model fitting.

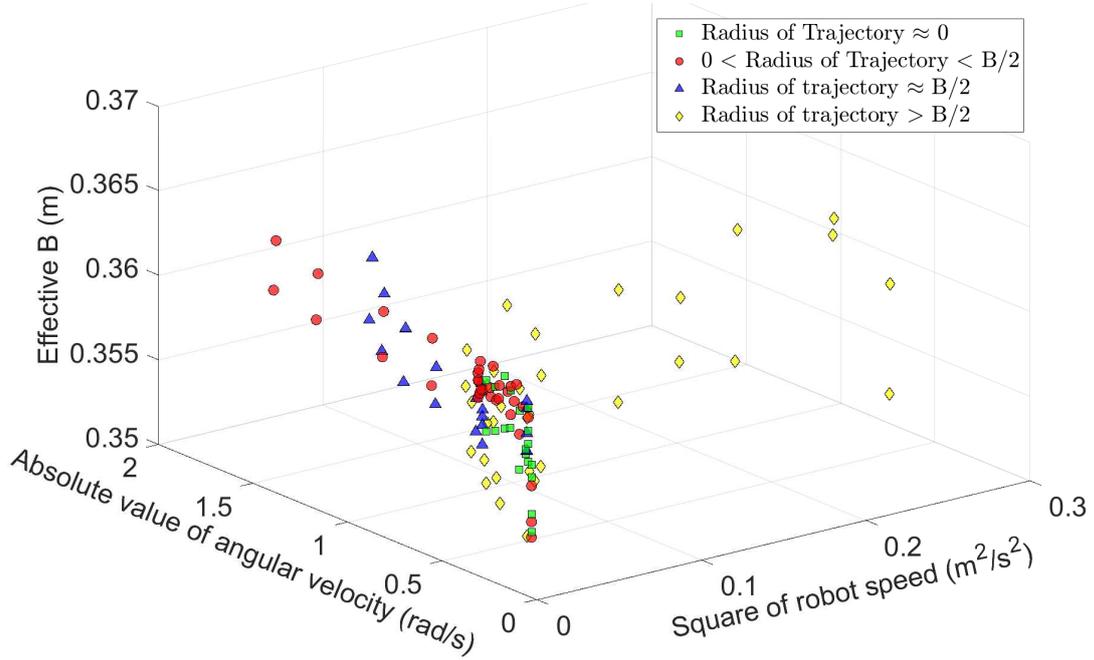
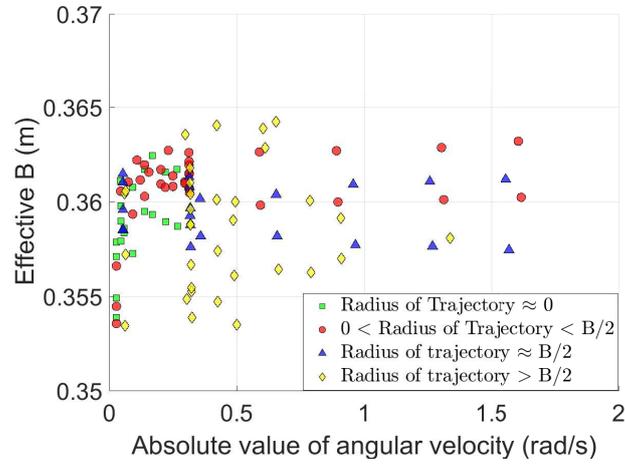


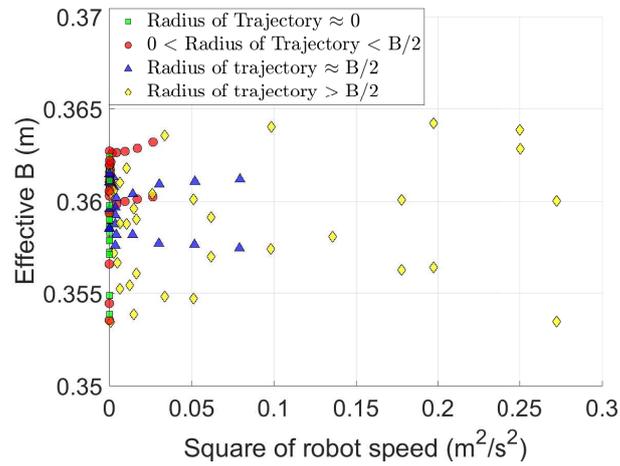
Figure 5.21: 3D view of the calibration results of wheel separation on a carpet.

for wheel radii and wheel angular velocities. For radii of curvature of circular trajectories greater than $B/2$ cases in Figure 5.23, the results match the linear model in Subsection 5.5.1.

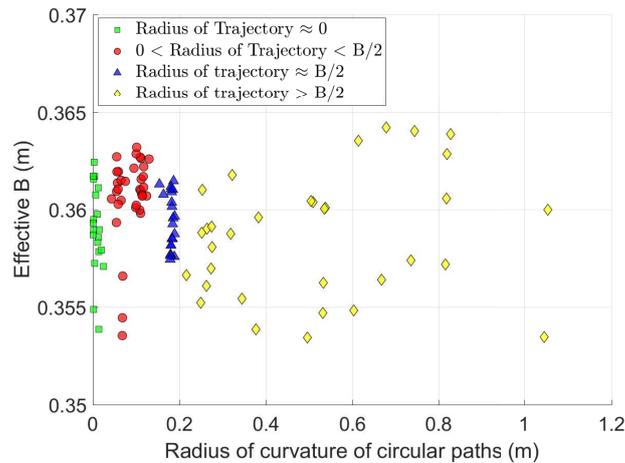
From the results in Figure 5.23, it is reasonable to assign a larger estimation variance to B on the carpet compared to that on the lino. The surface texture and softer materials of the carpet may cause significant non-systematic errors of the odometry system. However, the experimental results here still show that B increases linearly with respect to v^2 when the radii of curvature of robot trajectories are greater than $B/2$. It would be better to repeat this experiment for multiple times and report the statistical results of the proposed model, but this is not achievable due to the limit of the candidature period. The results in Figure 5.22 show the similar trend as listed in Section 5.6. For example, wheel separation distances spread more widely than the other three cases when Radius of Trajectory = 0. The effective wheel separation distances when $0 < \text{Radius of Trajectory} < B/2$ are generally larger than those when Radius of Trajectory =



(a)



(b)



(c)

Figure 5.22: Calibration results of wheel separation distance B when using saturation growth model for wheel radii and wheel angular velocities on a carpet. (a) B vs. ω . (b) B vs. v^2 . (c) B vs. Radius of Curvature.

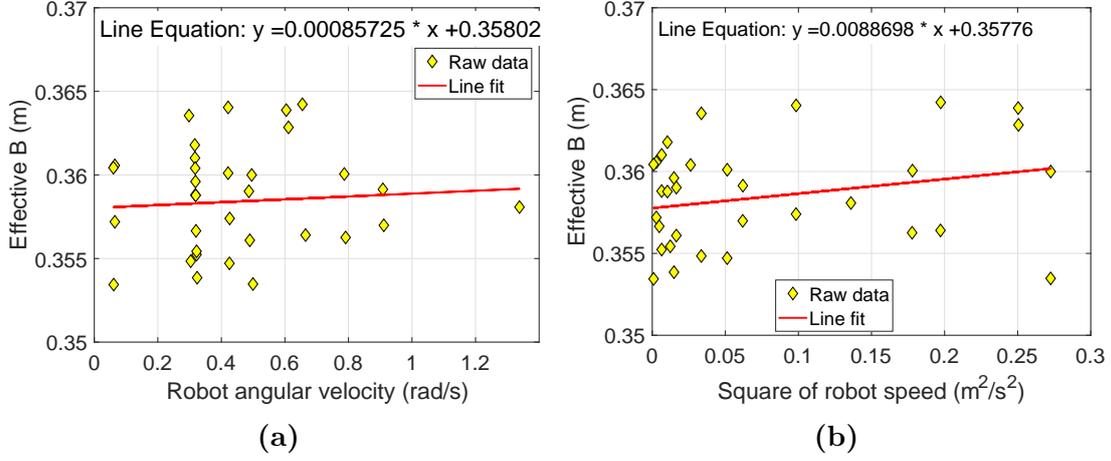


Figure 5.23: Partial calibration results of B on a carpet when using saturation growth model for radii of curvature greater than $B/2$ cases on a carpet. (a) B vs. ω . (b) B vs. v^2 .

$B/2$. Note that the tyre pressures of both wheels may have been different for datasets on Lino in the last section and datasets on the carpet in this section. Hence conclusions cannot be inferred from the comparisons of experiments on Lino and on the carpet.

5.8 Five Parameter Differential Drive Mobile Robot Odometry Model

Experiments and modelling in Section 5.5 suggest the following differential drive mobile robot odometry model with five independent variables as Equation (5.20) and Equation (5.21). Although the cases with wheel angular velocity less than 0.035 rad/s could not be checked experimentally, the implementation of the model limits robot wheel radii to be equal to the radii at wheel angular velocity of 0.035 rad/s for the utilised robot. This makes little difference in practice since robot movement will be very small at this speed. The robot ground speed is approximately 3.5 mm/s if we assume the wheel radius is 0.1 m.

$$R_l = g_1(\omega_l) = \begin{cases} R_{\infty l} + \frac{R_0 - R_{\infty l}}{\frac{|\omega_l|}{\tau} + 1}, & |\omega_l| \geq 0.035 \text{ rad/s} \\ R_{\infty l} + \frac{R_0 - R_{\infty l}}{\frac{0.035}{\tau} + 1}, & |\omega_l| < 0.035 \text{ rad/s} \end{cases} \quad (5.20)$$

$$R_r = g_2(\omega_r) = \begin{cases} R_{\infty r} + \frac{R_0 - R_{\infty r}}{\frac{|\omega_r|}{\tau} + 1}, & |\omega_r| \geq 0.035 \text{ rad/s} \\ R_{\infty r} + \frac{R_0 - R_{\infty r}}{\frac{0.035}{\tau} + 1}, & |\omega_r| < 0.035 \text{ rad/s} \end{cases} \quad (5.21)$$

- R_0 : wheel radius when the robot wheel does not rotate, $\omega_l = \omega_r = 0$ rad/s.
- $R_{\infty l}, R_{\infty r}$: limiting wheel radii when the left and the right wheels approach infinite speeds respectively, $\omega_l = \omega_r = \infty$ rad/s.
- τ : wheel angular velocity when wheel radius is equal to $(R_0 + R_{\infty l})/2$ and $(R_0 + R_{\infty r})/2$ for the left and the right wheels respectively.
- B : constant wheel separation distance.

5.8.1 Verification Experiments

This section calibrates the five parameter model developed above with real robot experiments under changing speeds and robot angular velocities on a hard lino floor.



Figure 5.24: Testing environment for a robot running with different speeds and angular velocities.

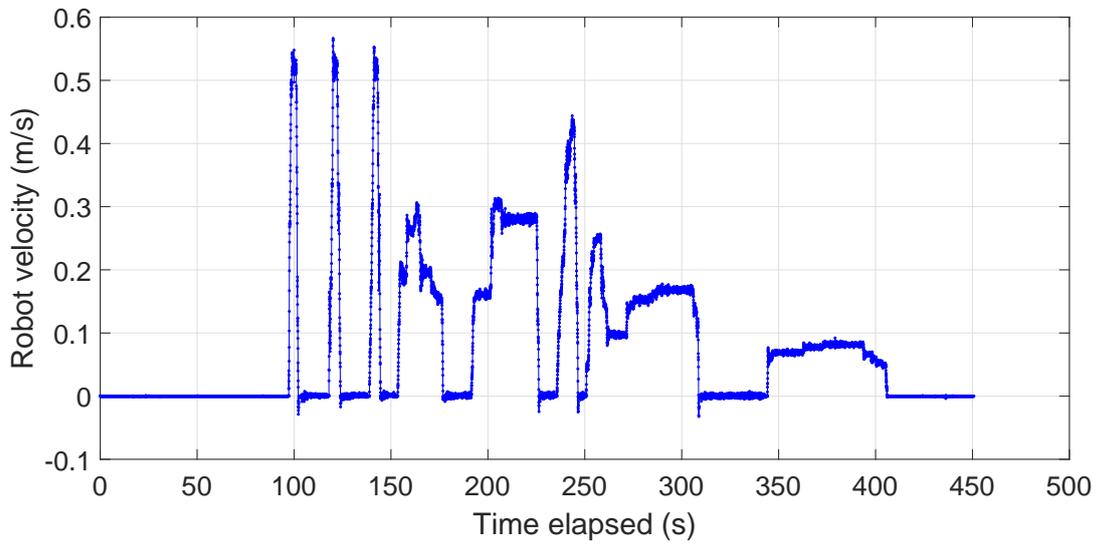
Datasets in this section were collected directly after experiments in Section 5.4 on September 27, 2017. Figure 5.24 shows the testing environment in this subsection. Figure 5.25 shows the robot speeds and linear accelerations during dataset collection in this subsection.

Figure 5.26 and Figure 5.27 show the mean and maximum Map Information over 500 generations of a PSO algorithm aiming to maximise the OG map quality respectively. Laser scans were sampled for every 0.3 m travelling distance and 15 degree rotation of the robot when being used to generate the OG map in the PSO. There are 283 different laser scans in total. Figure 5.28 shows the converging evolutions of five independent variables in the PSO. The optimised result is shown in Table 5.3.

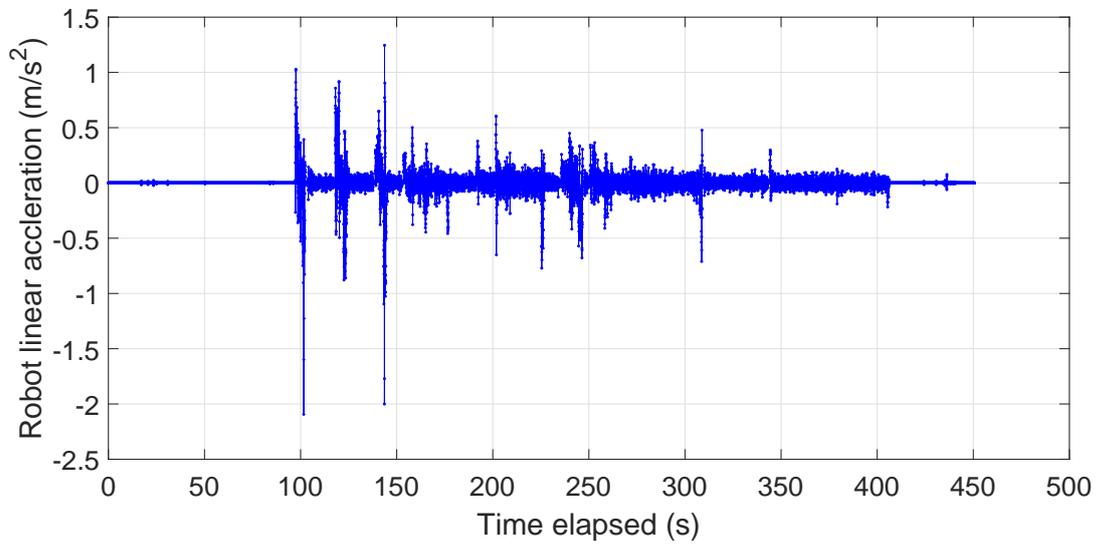
Figure 5.29 is another way to demonstrate that the five parameter odometry model is effective. Here a brute force search for one of the five parameters was performed by assigning the other four parameters to be the optimised result in Table 5.3. Every independent variable is showing a clear converging basin.

Figure 5.30 is a zoom-in view of two objects in the testing environment. The approximate widths of the wood block, the gap and the foam are labelled in the

5.8. FIVE PARAMETER DIFFERENTIAL DRIVE MOBILE ROBOT
ODOMETRY MODEL



(a)



(b)

Figure 5.25: Robot speeds and accelerations in Subsection 5.8.1. (a) Robot speeds. (b) Robot linear accelerations.

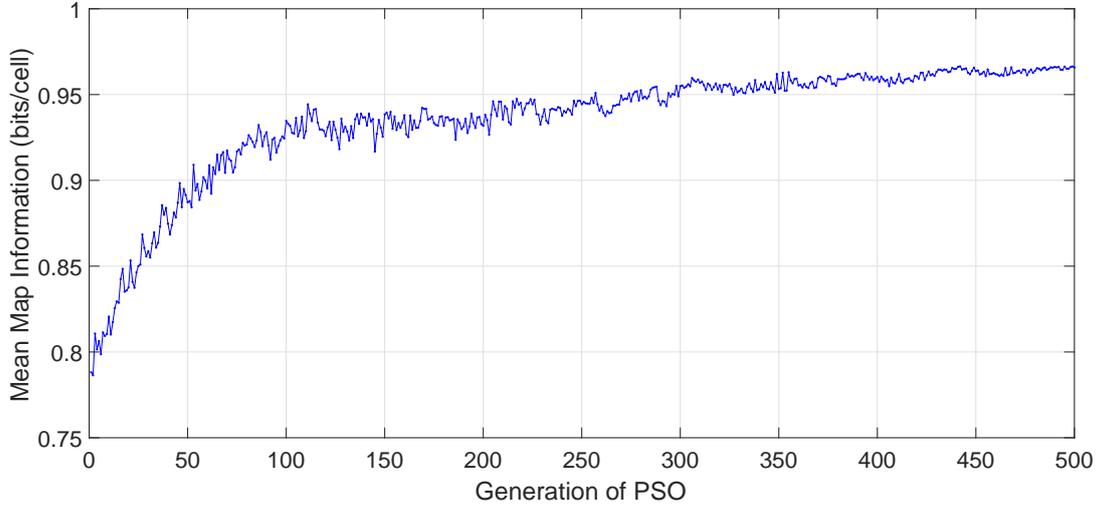


Figure 5.26: Mean Map Information over 500 generations of PSO in Subsection 5.8.1.

Table 5.3: Optimised Parameters of the Complete Odometry Model with a PSO in the Experiments in Figure 5.24.

| R_0 (m) | $R_{\infty l}$ (m) | $R_{\infty r}$ (m) | τ (rad/s) | B (m) | MI (bits/cell) |
|-----------|--------------------|--------------------|----------------|----------|----------------|
| 0.137829 | 0.0970833 | 0.0972093 | 0.0128328 | 0.358674 | 0.967074 |

figure. These measurements can be used as ground truths to justify the OG maps built with the calibrated odometry system as in Figure 5.31.

5.8. FIVE PARAMETER DIFFERENTIAL DRIVE MOBILE ROBOT
ODOMETRY MODEL

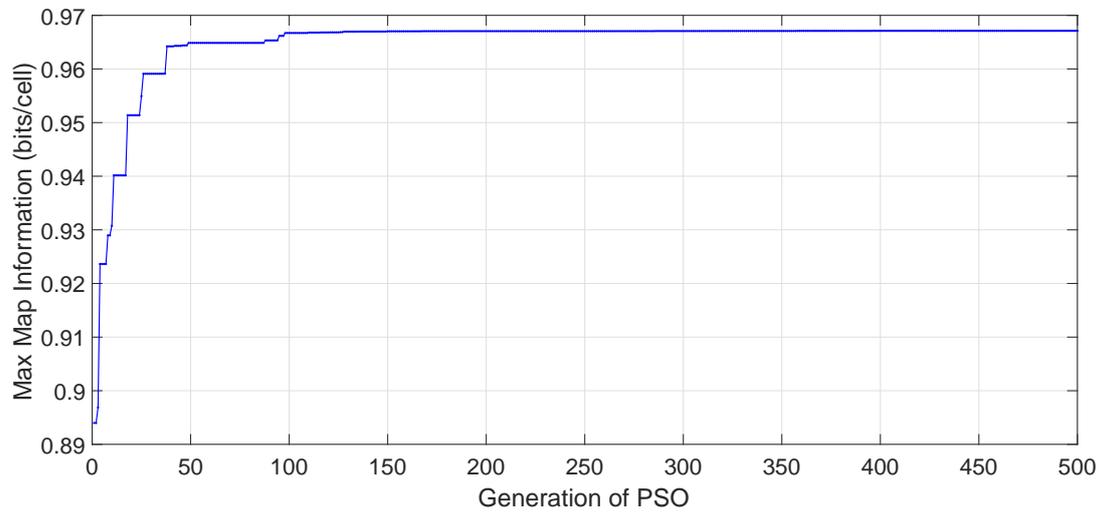


Figure 5.27: Maximum Map Information over 500 generations of PSO in Subsection 5.8.1.

5.8. FIVE PARAMETER DIFFERENTIAL DRIVE MOBILE ROBOT
ODOMETRY MODEL

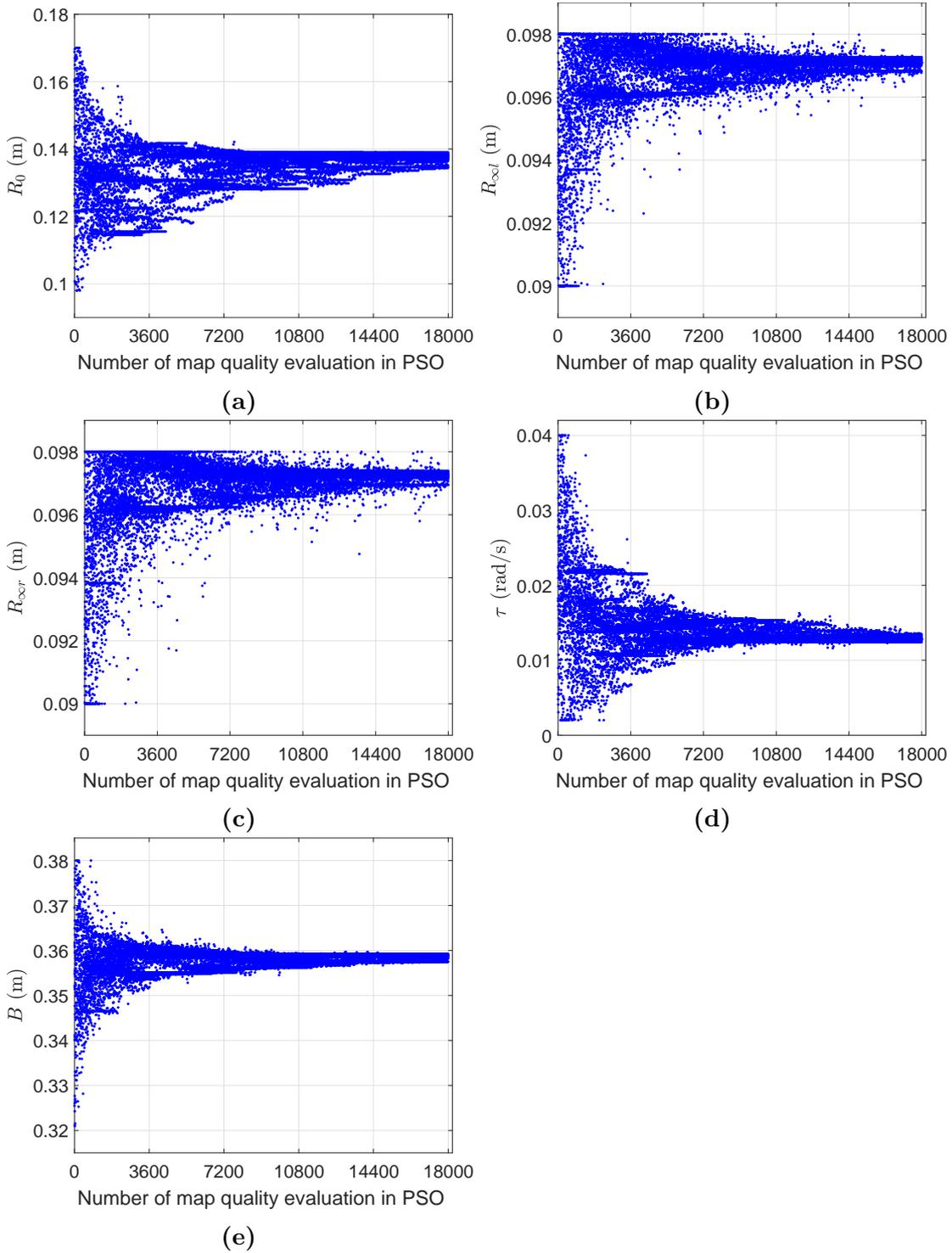


Figure 5.28: Evolutions of five independent variables in the PSO process. (a) R_0 . (b) R_{ool} . (c) R_{oor} . (d) τ . (e) B .

5.8. FIVE PARAMETER DIFFERENTIAL DRIVE MOBILE ROBOT
ODOMETRY MODEL

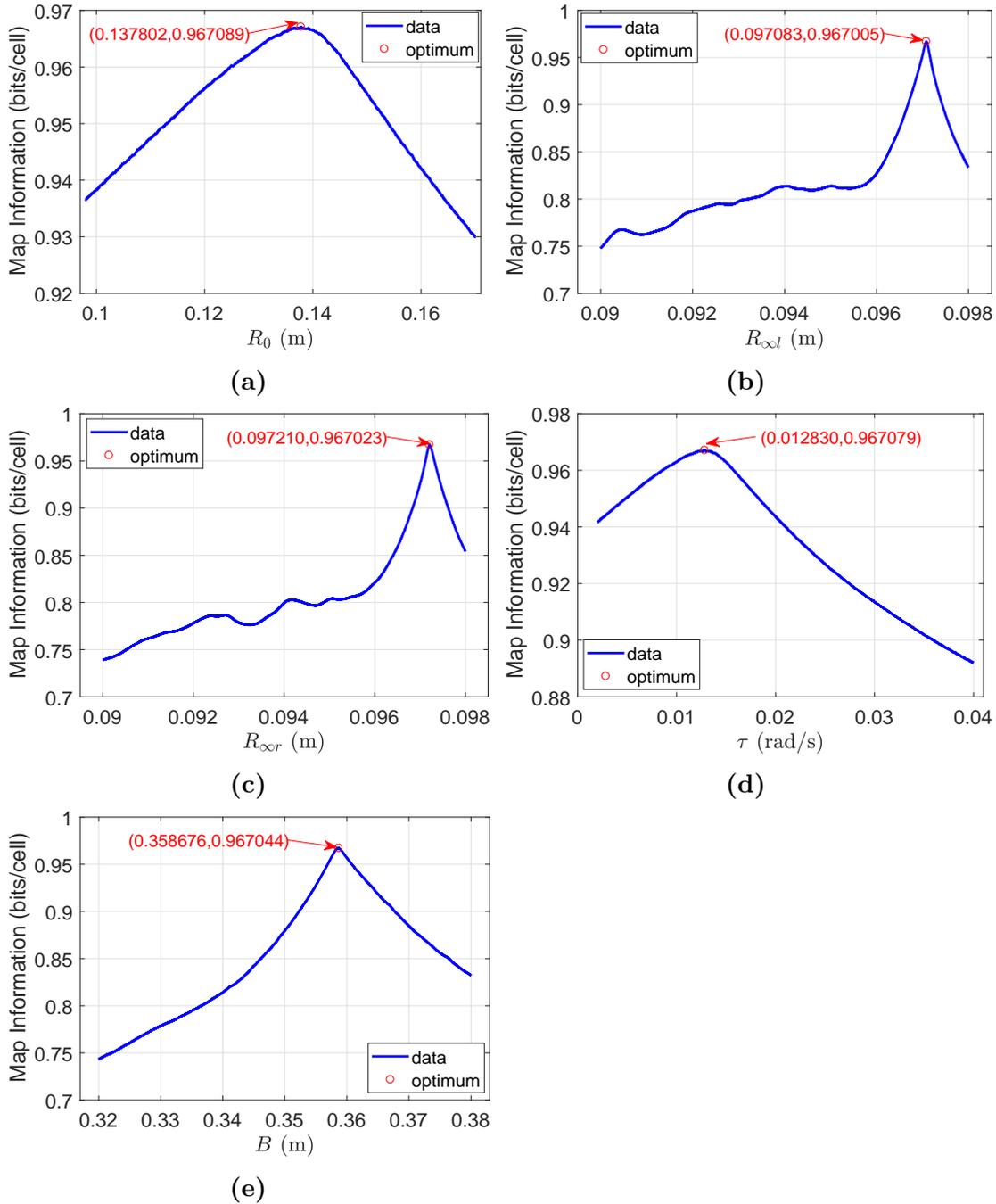


Figure 5.29: Grid searching of one of five independent variables in the complete odometry model by fixing the other four parameters to be the optimised result in Table 5.3. (a) R_0 . (b) $R_{\infty l}$. (c) $R_{\infty r}$. (d) τ . (e) B .

5.8. FIVE PARAMETER DIFFERENTIAL DRIVE MOBILE ROBOT
ODOMETRY MODEL

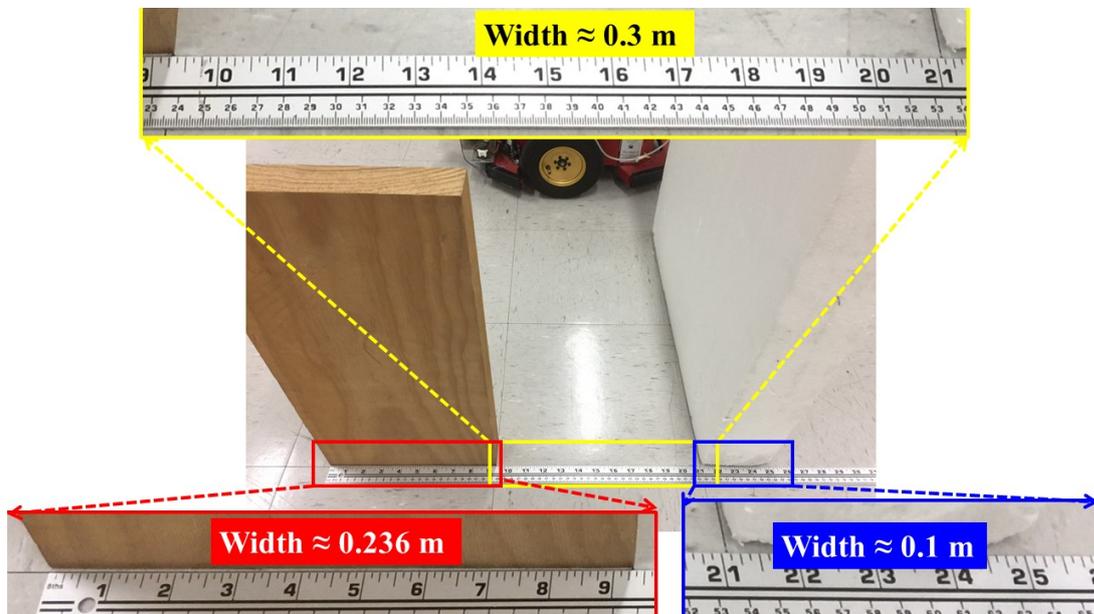
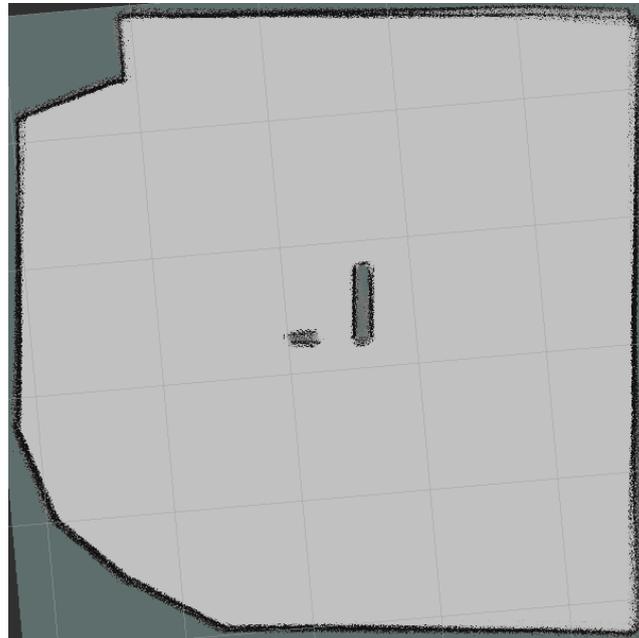
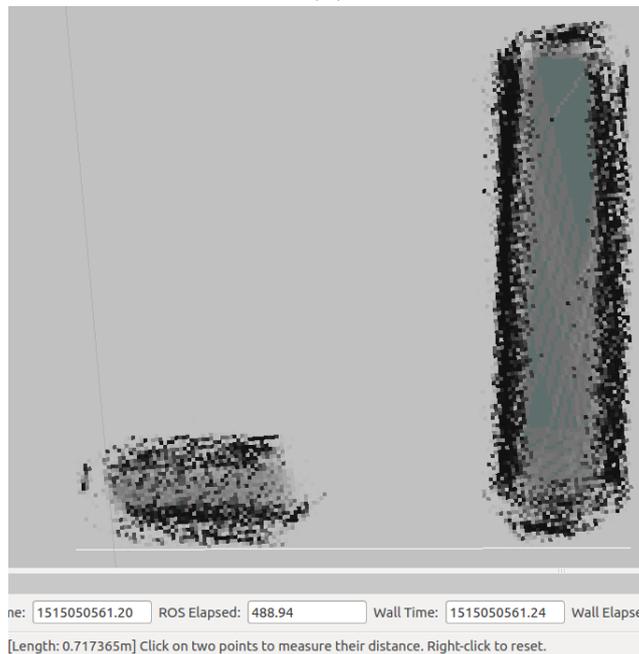


Figure 5.30: Zoom-in view of two objects in the testing environment.



(a)



(b)

Figure 5.31: OG map generated using optimised result in Table 5.3. There are 283 different laser scans by sampling every 0.3 m travelling distance and 15 degree rotation of the robot. (a) OG map. (b) Zoom-in view. Using the distance measurement tool in the ROS visualisation tool, *rviz*, the author measures a distance of 0.717 m in the grid map between the left edge of the wood block and the right edge of the white foam block. This distance is very close to the physical measurement of 0.636 m (i.e., summing the widths of all three segments) in Figure 5.30 considering there are 283 scans over a relatively long period and running distances.

5.9 Summary and Conclusion

This chapter has focused on odometry calibration on curved robot paths. A five parameter model for differential drive mobile robot equipped with pneumatic tyres is developed after some key modelling efforts. In the beginning, theoretical analysis based on the first principle force analysis and experiments about the relationship between wheel separation and the square of robot speed are conducted in Section 5.3 and Section 5.4. Then the author develops a saturation growth model representing the relationship between wheel radii and wheel angular velocities relevant for slow speeds in Section 5.5. The extra two parameters are due to wheel radii increases at slow speeds. The model shows that the wheel radius increases significantly as the wheel angular velocity decreases. This makes a difference of 9.2% and is significant for wheel angular velocities less than 0.035 rad/s for the Pioneer 2 robot used in this research. This model is utilised while investigating the relationship between robot wheel separation distance and the square of robot speeds in Section 5.6. Interestingly, the author intended to investigate the wheel separation with regard to the square of robot speed in the beginning. Then the shrinking effect of wheel radius was found with an increasing of wheel angular velocity. After applying the saturation growth model to the wheel shrinking effect, the wheel separation distances for different robot speeds did not show significant differences. In Section 5.7, the author provides justification experiments on a soft carpet, where much higher non-systematic errors were experienced compared to those on the lino. In both experiments on a hard lino and those on a soft carpet, some trends of the estimation of B values appeared. B values are noisier when Radius of Trajectory = 0, i.e., robot rotating on a spot compared to other cases. In addition, smaller B values were obtained for $0 < \text{Radius of Trajectory} < B/2$ cases compared to those when Radius of Trajectory = $B/2$. Lastly a five parameter differential drive odometry model is developed in Section 5.8.

Summary and Future Work

6.1 Summary

There has been intensive and continuous research into differential drive mobile robot odometry modelling and calibration over the last few decades. However, not much effort is focused on calibrating odometry models on heterogeneous indoor floors, despite many researchers pointing out that robot odometry parameters would vary depending on floor surfaces. Furthermore, there has been little investigation of the modelling the effects on odometry estimation brought about by robot linear accelerations and lateral accelerations. This thesis has focused on developing novel odometry models and comprehensive experiments to solve the above mentioned problems from the first principle taking advantage of state-of-the-art optimisation methods and classification techniques. Key achievements and findings of this thesis are summarised below:

Two Novel Floor Classification Systems: The author has developed two floor classification systems that allow fast differentiation among heterogeneous floors. The first one is based on a colour intensity sensor equipped beneath the robot. The colour intensity values of the light reflected by the floor surface are fed into a Support Vector Machine (SVM) classifier. This system achieves greater than 98.5% precision and recall rates for 6 different indoor floors. The other floor classification system uses motor current compensated by online floor inclination angle sensing from an accelerometer as the input data for a SVM classifier. This system can successfully differentiate floors with the same colour or colour patterns as shown in Chapter 2.

Calibration of Odometry and Sensor Extrinsic Poses on Heterogeneous

Floors: Secondly, two odometry calibration methods for mobile robot odometry and sensor extrinsic parameters on multiple floors have been developed in Chapter 3. An online odometry calibration system based on Extended Kalman Filter (EKF) for differential drive mobile robot on several floors has been designed and experimentally validated in both simulation and real lab environment. In addition, the author has developed an offline Particle Swarm Optimisation (PSO) based optimisation system aiming to calibrate odometry parameters and laser range finder (LRF) extrinsic parameters. The Map Information (MI) map quality metric of Occupancy Grid (OG) is chosen as the cost function of PSO. These two different calibration methods are shown to converge into a reasonable basin and result in better location and mapping quality.

Novel Odometry Model Incorporating Linear Acceleration: Thirdly, the author has incorporated robot linear acceleration into the widely utilised conventional differential drive robot odometry model and develops a four parameter odometry model. This new approach has been found after many real experiments and also validated. It has solid theoretic support from contact mechanics, in particular tyre mechanics. Comprehensive sensor synchronisations and LRF measurement calibrations have been performed before validating the proposed model. Results have shown a 9% difference of wheel radius estimation when acceleration ranges from -0.45 m/s^2 to 0.45 m/s^2 . This research finding has been published in the IEEE ICMA 2017 and won the Best Student Paper Award.

Novel Odometry Model Incorporating Lateral Acceleration: Lastly, wheel separation distance on curved paths has been investigated. A saturation growth model for the relationship between wheel radii and wheel angular velocities is proposed. This relationship is normally assumed constant. Afterwards, a five parameter differential drive mobile robot odometry model is proposed and validated using real experiments.

6.2 Future Work

Despite the contributions in this dissertation, there are still many interesting research problems outstanding relating to mobile robot odometry calibration. The author would like to list the future work below:

Odometry Modelling Incorporating Robot Slip Angles

In [78], Pacejka explains slip angle as the angle between a rolling wheel's actual direction of travel and the direction towards which it is pointing. How does a tyre slip angle contribute to the estimation of robot trajectories? The author has done some preliminary research on this topic. Due to the limit time of his PhD candidature, the author includes his theoretical analysis and preliminary experimental results in Appendix B. The differential drive mobile robot odometry model may be improved by incorporating slip angles in the future.

Sensor Fusion and Machine Learning for Floor Classification

Different floor sensors may have different strengths in particular applications. It is possible to develop a multifaceted indoor floor classification system by taking advantage of several sensors, including motor current sensors, colour intensity sensors, cameras, tactile sensors etc.. Their sensing data can be fused using a Kalman filter (KF). Another way to combine the information from different sensors is to use machine learning techniques. The author encourages researchers to apply unsupervised learning algorithms to classify different floor surfaces where human operators are not able to give predefined labels to surfaces. It is also possible to predict and classify floor types for each individual wheel, allowing odometry calibration or robot motion control in a more refined strategy. Some challenging floors, such as textured directional carpets, may require more careful

consideration, where we can use motor current sensors to detect the instantaneous rolling resistance, cameras to detect the pattern of light reflection if possible, tactile sensors to detect the roughness etc..

Minimum Energy Path Planning with a Floor Inclination Map

The compensated motor current can be considered in the energy efficient path planning problem [99] with the benefit of detecting robot load changes and adjusting motion accordingly. If a robot is equipped with inclination angle detection sensors, it can generate a map with different inclination angles of the seemingly flat indoor floors after traversing most areas of the floor. This inclination map is beneficial for planning a minimum energy cost path from point to point. In addition, this map could be integrated into the conventional map from a Simultaneous Localisation and Mapping (SLAM) system. Therefore, a location in a map will encode not only its position information but also its relative height.

Faster Odometry Calibration for Robots on Multiple Floors via Information Sharing

To the knowledge of the author, there have been few publications about multiple robot odometry calibration on different floors. Calibrating odometry parameters for a *group* of mobile robots on different floors has several benefits. Firstly, it may be possible to achieve a faster and more accurate odometry calibration with a proper coordination strategy. For example, one robot can adjust its initial estimate of the odometry parameters on one floor if another robot has calibrated its odometry on that floor. Then the convergence speed of EKF or other optimisation algorithms used for odometry calibration will be faster. Secondly, multiple robots can share the floor classification results to reduce classification errors. Last but not least, users may only need to calibrate one of the homogeneous robots to

get an initial estimate of odometry parameters. This could save time but with some loss of accuracy of the odometry parameters.

Test the Developed Model in Larger Environments

The LRF used in this thesis, Hokuyo URG-04LX LRF, has limited range measurements from 0.02 m to 4 m. In addition, the lab does not have larger space for experiments. These two limitations constrain the testing conditions in terms of robot angular velocities, robot linear acceleration, robot speeds, map size and so on. The author would like to perform experiments beyond these limits to see the influence of greater tyre forces (longitudinal and lateral) to mobile robot odometry systems. It appears that no research papers have gone this far. This investigation will be beneficial to those robots working at higher speeds, such as those in a warehouse for logistic purposes.

Odometry Model for Non-constant Centrifugal Acceleration on Curved Paths

Anecdotal evidence suggests that robot trajectories have a consistent bias to the outward/inward side of curved paths when accelerating/decelerating. This phenomenon is worth more investigation. For this reason, the author did not include this situation in the modelling in Section 5.8. According to [67], researchers have attempted to formulate an analytical framework for predicting the longitudinal force and the cornering force as a function of combined longitudinal slip and slip angles. For instance, it is possible to predict cornering force available at a specific slip angle in the presence of a tractive or braking force using a friction ellipse plot. The friction ellipse for a specific tyre requires many experimental tests. Cornering force of tyres can be related to the robot wheel elastic displacement and therefore more physically meaningful odometry models may be developed.

Investigate Odometry Model for Robots with Other Tyres (Not Pneumatic)

The research findings in Chapter 4 and Chapter 5 are applicable to mobile robots with pneumatic tyres. How about robots with solid rubber tyres or other materials? These types of tyres have different characteristics of contact mechanics against floors. They may or may not have significant effects on robot odometry estimation.

Optimise Map Quality with Other Algorithms to Calibrate Odometry Parameters

Since the author shows that the function between map quality and odometry parameters has a reasonably smooth convergence basin in Subsection 3.4.6, it is possible to use other commonly available optimisation methods, like gradient descent, to calibrate odometry parameters faster than a PSO algorithm.

Validate Odometry Models with Open Source Datasets

It is possible to validate and compare different odometry models using publicly available datasets. One of those is Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets (RAWSEEDS) dataset [100]. It is a EU-financed project focusing on providing benchmarking in Robotics from 2006 to 2009. It includes accurate ground truth data together with high quality, multi-sensor and different environments (indoor and outdoor, static and dynamic, artificial lighting and daylight etc.). All data from the sensors, including encoder data, were logged in raw formats with accurately synchronized timestamp according to a single reference clock source. The ground truth in outdoor environments were collected using a standard RTK-GPS apparatus with a

precision up to 2-6 cm. External fixed LRFs and fiducial marker detection cameras were used in the indoor ground truth of robot pose collections. In addition, they built up a validation system with accurate laser meter measurements and proper data processing. Data fusion was implemented using a KF for robot poses within the perception of the fixed laser range finders and cameras for ground truth collection.

Investigate the Saturation Growth Model for Robot Moving Backwards, Clockwise and Anti-clockwise

The saturation growth model developed in Section 5.5 only involves forward movement cases of the testing robot. One of the assumptions is that the model will be the same for backward movements of the robot. Although the tyres of the robot show symmetrical patterns regarding forward and backward rotations, it would be better to investigate the backward movement model. This work is left for the future research. In this thesis, the author assumes that the robot rotation direction, i.e. clockwise or anti-clockwise, would not affect the odometry estimation. More investigation about this research direction is also worthy.

6.3 Concluding Remarks

Mobile robots are deployed in more and more scenarios like schools, factories and even on other planets. This pushes forward research about modelling and calibrating mobile robot odometry systems. This thesis shines some light upon this topic. It investigates the dependence of floor surfaces, wheel slip, linear acceleration and lateral acceleration. There is still much space for improvement by other researchers. Supporting theories in the tyre mechanics field exist but yet to be applied to wheeled mobile robots. The longitudinal and lateral forces on the tyre are complicated. Fortunately, there are several different tyre models for robotics researchers to use. More accurate localisation and higher quality map will

6.3. CONCLUDING REMARKS

be achievable with better odometry calibrations. Robotics, in general, is inspiring and challenging. The author believes more exciting ideas and experiments will be developed in the future.

Appendices

Derivation of Jacobian with a Rank 2 on Constant Curvature Turning in Subsection 3.3.1

This appendix contains a significant proportion of material developed by A/Prof. Lindsay Kleeman.

The following derivation is done in a continuous time domain, t . When a differential drive robot is running on an arc trajectory with a constant curvature on a flat floor, its motion is defined by its left and right wheel angular speeds as below.

$$\begin{cases} \omega_l(t) = k\omega_c \\ \omega_r(t) = \omega_c \end{cases} \quad (\text{A.1})$$

where k and ω_c are constants. For the convenience of derivation without losing generality, the starting pose of the robot was set to be $\theta(0) = 0$, $x(0) = 0$ and $y(0) = 0$. R_d and R_s are defined as

$$R_d \triangleq R_r - kR_l \quad (\text{A.2})$$

$$R_s \triangleq R_r + kR_l \quad (\text{A.3})$$

Therefore we have the robot pose at time t

$$\begin{aligned}
\theta(t) &= \int_0^t \frac{R_r \omega_r(t) - R_l \omega_l(t)}{B} dt \\
&= \omega_c \frac{R_r - k R_l}{B} t \\
&= \omega_c \frac{R_d}{B} t
\end{aligned} \tag{A.4}$$

$$\begin{aligned}
x(t) &= \int_0^t \frac{R_r \omega_r(t) + R_l \omega_l(t)}{2} \cos(\theta(t)) dt \\
&= \frac{\omega_c R_s}{2} \int_0^t \cos\left(\omega_c \frac{R_d}{B} t\right) dt \\
&= \frac{R_s B}{2 R_d} \sin\left(\frac{R_d \omega_c t}{B}\right)
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
y(t) &= \int_0^t \frac{R_r \omega_r(t) + R_l \omega_l(t)}{2} \sin(\theta(t)) dt \\
&= \frac{\omega_c R_s}{2} \int_0^t \sin\left(\omega_c \frac{R_d}{B} t\right) dt \\
&= \frac{R_s B}{2} \left[\frac{B}{R_d \omega_c} \left(1 - \cos\left(\frac{R_d \omega_c t}{B}\right)\right) \right] \\
&= \frac{R_s B}{2 R_d} \left(1 - \cos\left(\frac{R_d \omega_c t}{B}\right)\right)
\end{aligned} \tag{A.6}$$

From Equations (A.4-A.6), we can see that the robot pose $\mathbf{z}(t) = [\theta(t), x(t), y(t)]^T$, only depends on the parameters, subject to Equation (A.7).

$$\begin{cases} R_s = R_r + kR_l \\ R_{dB} \triangleq \frac{R_d}{B} = \frac{R_r - kR_l}{B} \end{cases} \quad (\text{A.7})$$

For a constant arc motion defined by Equation (A.1), what is the parameter space of odometry parameters, $[R_l, R_r, B]^T$? Let R_s and R_{dB} be fixed. We can get Equation (A.8), which represents a parametric line in the 3D parameter space spanned by $[R_l, R_r, B]^T$ with independent variable $R_l \triangleq \eta$. Any point on this 3D straight line will produce the same motion when Equation (A.1) is satisfied.

$$\begin{cases} R_l = \eta \\ R_r = R_s - k\eta \\ B = \frac{R_s}{R_{dB}} - \frac{2k}{R_{dB}}\eta \end{cases} \quad (\text{A.8})$$

For constant arc turning example, the robot pose has only 2 degrees of freedom. Hence the Jacobian of robot pose with respect to odometry parameters, $\mathbf{J}(t)$, can only have a rank of less or equal to 2. This will be explained in the following derivation. Follow the previous convention to define $l(t) = 1/B(t)$ and $\mathbf{x}(t) = [R_l(t), R_r(t), l(t)]^T$. These variables are in the continuous time domain.

$$\mathbf{J}(t) = \frac{\partial \mathbf{z}(t)}{\partial \mathbf{x}(t)} = \begin{bmatrix} \frac{\partial \theta(t)}{\partial R_l(t)} & \frac{\partial \theta(t)}{\partial R_r(t)} & \frac{\partial \theta(t)}{\partial l(t)} \\ \frac{\partial x(t)}{\partial R_l(t)} & \frac{\partial x(t)}{\partial R_r(t)} & \frac{\partial x(t)}{\partial l(t)} \\ \frac{\partial y(t)}{\partial R_l(t)} & \frac{\partial y(t)}{\partial R_r(t)} & \frac{\partial y(t)}{\partial l(t)} \end{bmatrix} \quad (\text{A.9})$$

For the second and third row of $\mathbf{J}(t)$, we need the following three equations. (t) will be omitted for the related variables from now on.

$$\begin{aligned}
\frac{\partial (R_s/R_d)}{\partial R_l} &= \frac{R_d \frac{\partial R_s}{\partial R_l} - R_s \frac{\partial R_d}{\partial R_l}}{R_d^2} \\
&= \frac{k(R_d + R_s)}{R_d^2} \\
&= \frac{2kR_r}{R_d^2}
\end{aligned} \tag{A.10}$$

$$\begin{aligned}
\frac{\partial (R_s/R_d)}{\partial R_r} &= \frac{R_d \frac{\partial R_s}{\partial R_r} - R_s \frac{\partial R_d}{\partial R_r}}{R_d^2} \\
&= \frac{(R_d - R_s)}{R_d^2} \\
&= \frac{-2kR_l}{R_d^2}
\end{aligned} \tag{A.11}$$

$$\begin{aligned}
\frac{\partial B}{\partial l} &= \frac{\partial 1/l}{\partial l} \\
&= -B^2
\end{aligned} \tag{A.12}$$

Now we can calculate each term of $\mathbf{J}(t)$ as below.

$$\frac{\partial \theta}{\partial R_l} = \frac{-k\omega t}{B} \tag{A.13}$$

$$\frac{\partial \theta}{\partial R_r} = \frac{k\omega t}{B} \tag{A.14}$$

$$\frac{\partial \theta}{\partial l} = R_d \omega t \tag{A.15}$$

$$\begin{aligned}
\frac{\partial x}{\partial R_l} &= \frac{B}{2} \left[\frac{\partial (R_s/R_d)}{\partial R_l} \sin \left(\frac{R_d \omega t}{B} \right) - \frac{k \omega t R_s}{R_d B} \cos \left(\frac{R_d \omega t}{B} \right) \right] \\
&= \frac{B}{2} \left[\frac{2k R_r}{R_d^2} \sin \left(\frac{R_d \omega t}{B} \right) - \frac{k \omega t R_s}{R_d B} \cos \left(\frac{R_d \omega t}{B} \right) \right] \tag{A.16}
\end{aligned}$$

$$\begin{aligned}
&= \frac{k B R_r}{R_d^2} \sin \theta - \frac{k R_s}{2 R_d} \omega t \cos \theta \\
\frac{\partial x}{\partial R_r} &= \frac{B}{2} \left[\frac{\partial (R_s/R_d)}{\partial R_r} \sin \left(\frac{R_d \omega t}{B} \right) + \frac{\omega t R_s}{R_d B} \cos \left(\frac{R_d \omega t}{B} \right) \right] \\
&= \frac{B}{2} \left(\frac{-2k R_l}{R_d^2} \sin \theta + \frac{\omega t R_s}{R_d B} \cos \theta \right) \tag{A.17}
\end{aligned}$$

$$\begin{aligned}
&= \frac{-k B R_l}{R_d^2} \sin \theta + \frac{R_s}{2 R_d} \omega t \cos \theta \\
\frac{\partial x}{\partial l} &= \frac{R_s}{2 R_d} \left(\frac{\partial B}{\partial l} \sin \theta + B \frac{\partial \sin \theta}{\partial l} \right) \\
&= \frac{R_s}{2 R_d} (-B^2 \sin \theta + B R_d \omega t \cos \theta) \tag{A.18}
\end{aligned}$$

$$\begin{aligned}
&= \frac{-B^2 R_s}{2 R_d} \sin \theta + \frac{B \omega t R_s}{2} \cos \theta \\
\frac{\partial y}{\partial R_l} &= \frac{B}{2} \left[\frac{\partial (R_s/R_d)}{\partial R_l} (1 - \cos \theta) - \frac{R_s}{R_d} \frac{\partial \cos \theta}{\partial R_l} \right] \\
&= \frac{B}{2} \left[\frac{2k R_r}{R_d^2} (1 - \cos \theta) - \frac{R_s k \omega t}{R_d B} \sin \theta \right] \tag{A.19} \\
&= \frac{k B R_r}{R_d^2} (1 - \cos \theta) - \frac{k R_s \omega t}{2 R_d} \sin \theta
\end{aligned}$$

$$\begin{aligned}
\frac{\partial y}{\partial R_r} &= \frac{B}{2} \left[\frac{\partial (R_s/R_d)}{\partial R_r} (1 - \cos \theta) - \frac{R_s}{R_d} \frac{\partial \cos \theta}{\partial R_r} \right] \\
&= \frac{B}{2} \left[\frac{-2kR_l}{R_d^2} (1 - \cos \theta) + \frac{R_s \omega t}{R_d B} \sin \theta \right] \tag{A.20}
\end{aligned}$$

$$\begin{aligned}
&= \frac{-kBR_l}{R_d^2} (1 - \cos \theta) + \frac{R_s \omega t}{2R_d} \sin \theta \\
\frac{\partial y}{\partial l} &= \frac{-B^2 R_s}{2R_d} (1 - \cos \theta) + \frac{BR_s \omega t}{2} \sin \theta \tag{A.21}
\end{aligned}$$

Summarise the individual element of \mathbf{J} as Equation (A.22). Simplify the symbols $\sin \theta$ and $\cos \theta$ as $s\theta$ and $c\theta$ respectively.

$$\mathbf{J} = \begin{bmatrix} \frac{-k\omega t}{B} & \frac{\omega t}{B} & R_d \omega t \\ \frac{kBR_r}{R_d^2} s\theta - \frac{kR_s \omega t}{2R_d} c\theta & \frac{-kBR_l}{R_d^2} s\theta + \frac{R_s \omega t}{2R_d} c\theta & \frac{-B^2 R_s}{2R_d} s\theta + \frac{BR_s \omega t}{2} c\theta \\ \frac{kBR_r}{R_d^2} (1 - c\theta) - \frac{kR_s \omega t}{2R_d} s\theta & \frac{-kBR_l}{R_d^2} (1 - c\theta) + \frac{R_s \omega t}{2R_d} s\theta & \frac{-B^2 R_s}{2R_d} (1 - c\theta) + \frac{BR_s \omega t}{2} s\theta \end{bmatrix} \tag{A.22}$$

If \mathbf{J} has full rank, then its inverse \mathbf{J}^{-1} exists and any robot pose error can be corrected by a change in the odometry parameter vector. The author aims here to show that for a constant curvature arc motion, \mathbf{J} has a rank of 2 at most and therefore \mathbf{J}^{-1} does not exist. By observing Equation (A.22), we have the following denotations.

$$\mathbf{v}_1 = \left[\frac{kR_r}{R_d} \quad \frac{-kR_l}{R_d} \quad \frac{-BR_s}{2} \right]^T \tag{A.23}$$

$$\mathbf{v}_2 = \left[\frac{-k}{2R_d} \quad \frac{l}{2R_d} \quad \frac{B}{2} \right]^T \tag{A.24}$$

$$\mathbf{J} = \begin{bmatrix} \alpha\omega t\mathbf{v}_1^T + \beta\omega t\mathbf{v}_2^T \\ \frac{B}{R_d}\sin\theta\mathbf{v}_1^T + R_s\omega t\cos\theta\mathbf{v}_2^T \\ \frac{B}{R_d}(1 - \cos\theta)\mathbf{v}_1^T + R_s\omega t\sin\theta\mathbf{v}_2^T \end{bmatrix} \quad (\text{A.25})$$

Assume Equation (A.26) from the first row of \mathbf{J} has a solution of $[\alpha, \beta]^T$.

$$[\mathbf{v}_1 \ \mathbf{v}_2][\alpha \ \beta]^T = \begin{bmatrix} -k & 1 \\ B & B \end{bmatrix} R_d \quad (\text{A.26})$$

Solving Equation (A.26), we get

$$\begin{cases} \alpha = 0 \\ \beta = \frac{2R_d}{B} \end{cases} \quad (\text{A.27})$$

Hence this verifies the assumption that \mathbf{J} has rank 2.

B

Investigation of Robot Slip Angle Effects

B.1 Theoretical Analysis

When cornering with centrifugal force F_s , a car or truck tyre is known to experience a slip angle α . At forces well below the adhesion limit of F_s , the slip angle is proportional to F_s [67] if the camber angle is zero.

$$\alpha = C_\alpha F_s \quad (\text{B.1})$$

where C_α is called cornering stiffness. C_α is defined as the derivative of the cornering force with respect to slip angle evaluated at zero slip angle:

$$C_\alpha = \left. \frac{\partial F_s}{\partial \alpha} \right|_{\alpha=0} \quad (\text{B.2})$$

According to [67, 101], the normal load on a tyre strongly influences tyre cornering characteristics. The transfer of load from inside to outside tyres during a turning reduces the total cornering force that a pair of tyres can develop. For a pair of tyres on a beam axle to develop the required amount of cornering force to balance a given centrifugal force under a steady-state turn, the lateral load transfer results in an increase in the slip angle of tyres [67].

Note that the robot heading for each time step $\Delta\theta(i)$ will remain the same as Equation (3.4). However the actual robot moving direction will be

$$h_r(i) = \theta(i-1) + \alpha(i) + \Delta\theta(i)/2 \quad (\text{B.3})$$

where $\alpha(i)$ is robot slip angle at time step i . In addition, define the sign of $\alpha(i)$ to be positive when robot turns right and negative when robot turns left. Here the slip angle is considered in terms of the robot instead of each tyre. Equation (3.1-3.4) and (B.4) are the modified differential drive robot odometry model considering slip angles.

$$\begin{bmatrix} \theta(k) \\ x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} \theta(0) + \sum_{i=1}^k \Delta\theta(i) \\ x(0) + \sum_{i=1}^k \frac{\Delta L(i)}{\cos \alpha(i)} \cos[\theta(i-1) + \alpha(i) + \Delta\theta(i)/2] \\ y(0) + \sum_{i=1}^k \frac{\Delta L(i)}{\cos \alpha(i)} \sin[\theta(i-1) + \alpha(i) + \Delta\theta(i)/2] \end{bmatrix} \quad (\text{B.4})$$

For small α in radians, we have

$$\begin{aligned} \frac{\Delta L}{\cos \alpha} &= 1 + \frac{1}{2}\alpha^2 + O(\alpha^4) \\ &\cong 1 + \frac{1}{2}\alpha^2 \end{aligned} \quad (\text{B.5})$$

According to [67], slip angles for road vehicles with pneumatic tyres under normal conditions are very small. If $\alpha = 4^\circ$, the relative error between $\frac{\Delta L}{\cos \alpha}$ and ΔL is approximately 0.25%.

B.2 Robot Slip Angle Measurements with Circular Paths

As defined in [102], the *vehicle slip angle* β is the difference between the vehicle heading and direction travelled (the path tangent). The odometry centre of the robot is treated as the position of the robot trajectory. For a circular path, the *vehicle slip angle* is illustrated in Figure B.1.

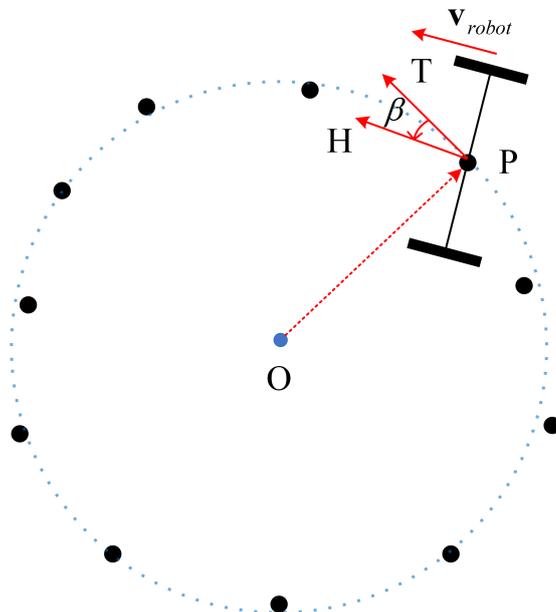


Figure B.1: Schematic of robot slip angle on circular paths. Solid black dots represent robot positions. O represents the centre of a fitted circle of robot positions. \overrightarrow{PH} and \overrightarrow{PT} represent the vehicle heading and the path tangent, respectively. $\overrightarrow{PT} \perp \overrightarrow{OP}$.

The experiments in this section were performed on Lino as in Figure B.2. Different speed and angular velocity commands were configured to let the robot automatically drive in circles. For several circles, the robot maintained a constant speed and angular velocity before changing to another new set of speed and angular velocity. The data during speed changing periods were eliminated for analysis. Only the data during the constant speed were taken into account. The median of these estimated slip angles on a certain circular path is treated as the slip angle of the robot on that path. Every laser scan is utilised for scan matching and generating the OG map in this section.

To get the heading of the robot the author applies the scan matching algorithm Normal Distributions Transform (NDT) [103] upon a scan measurement and localise this scan with respect to the first laser scan in the dataset. NDT is chosen, since it is implemented in the Robotics System Toolbox of MATLAB. For example, the robot has the following states: $v_{robot} = 0.052$ m/s,

B.2. ROBOT SLIP ANGLE MEASUREMENTS WITH CIRCULAR PATHS



Figure B.2: The testing environment for measuring the robot slip angles on circular paths on Lino. (a) Robot is being charged. (b) Robot is automatically driving in a circle.

Radius of Curvature = 0.281 m, ω_{robot} = 0.187 rad/s, B = 0.3660 m and counter clockwise rotating trajectory in one of the datasets. Figure B.3 shows scan matching results on circular paths using the NDT algorithm in MATLAB. Figure B.4 is the OG map generated using the scan matching results of robot poses and its related scans. The robot slip angle estimation results from different datasets is shown in Figure B.6. It is clear that β is highly correlated to the radius of robot circular trajectory. Intuitively, robot slip angles are positively correlated to centrifugal accelerations. In contrast, our preliminary results in Figure B.6 (d) do not show a clear correlation between robot slip angles and robot centrifugal accelerations. This is worth more investigation in the future.

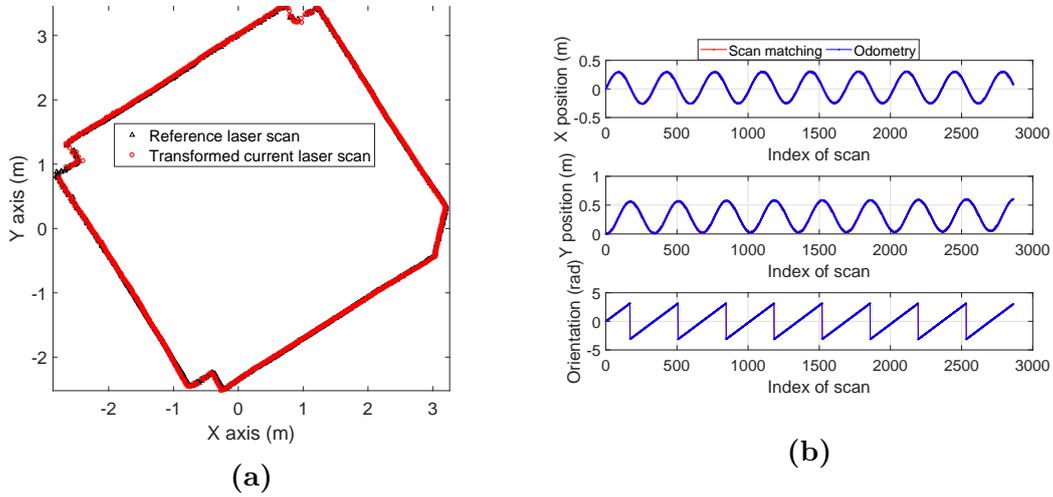


Figure B.3: Scan matching results using NDT algorithm in MATLAB when measuring robot slip angles on circular paths on Lino. (a) Example scan matching. (b) Robot poses.

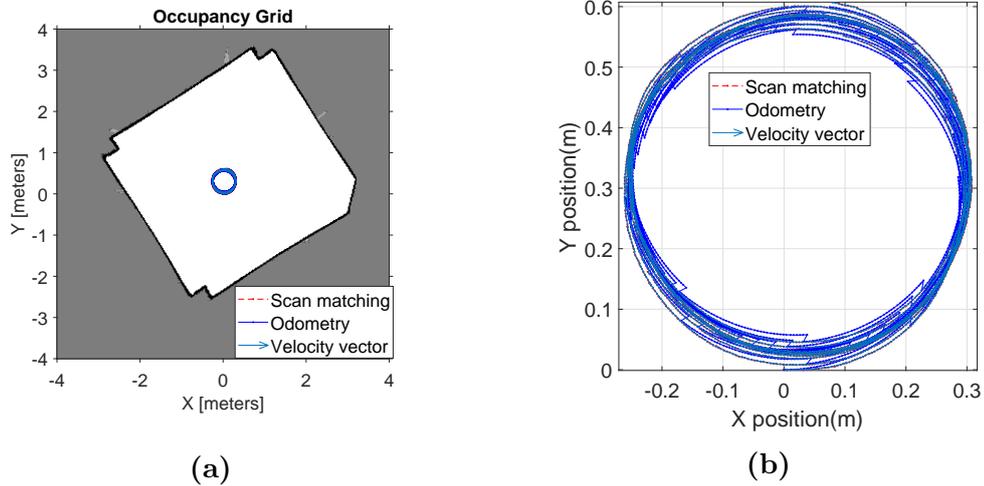


Figure B.4: (a) OG map generated using the scan matching results of robot poses and its related scans during the circular movement on Lino. The OG map is built by attaching 2,864 laser scans to robot poses. (b) Zoom in view of Figure B.4 (a).

B.2. ROBOT SLIP ANGLE MEASUREMENTS WITH CIRCULAR PATHS

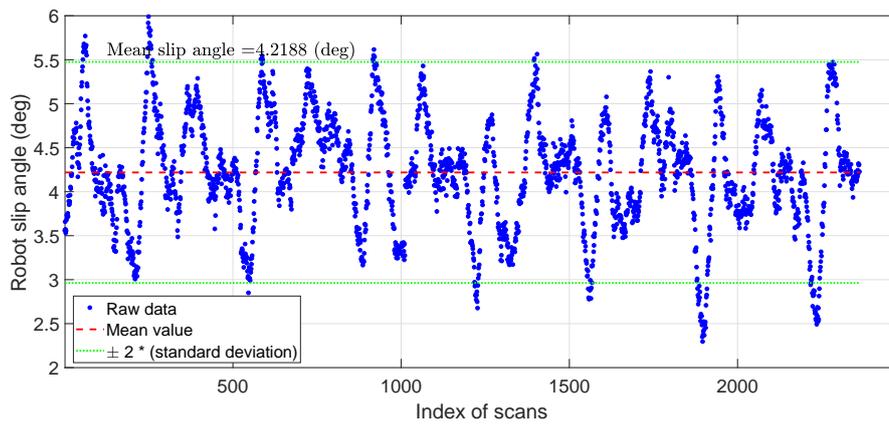


Figure B.5: Robot slip angles during the circular movement on Lino.

B.2. ROBOT SLIP ANGLE MEASUREMENTS WITH CIRCULAR PATHS

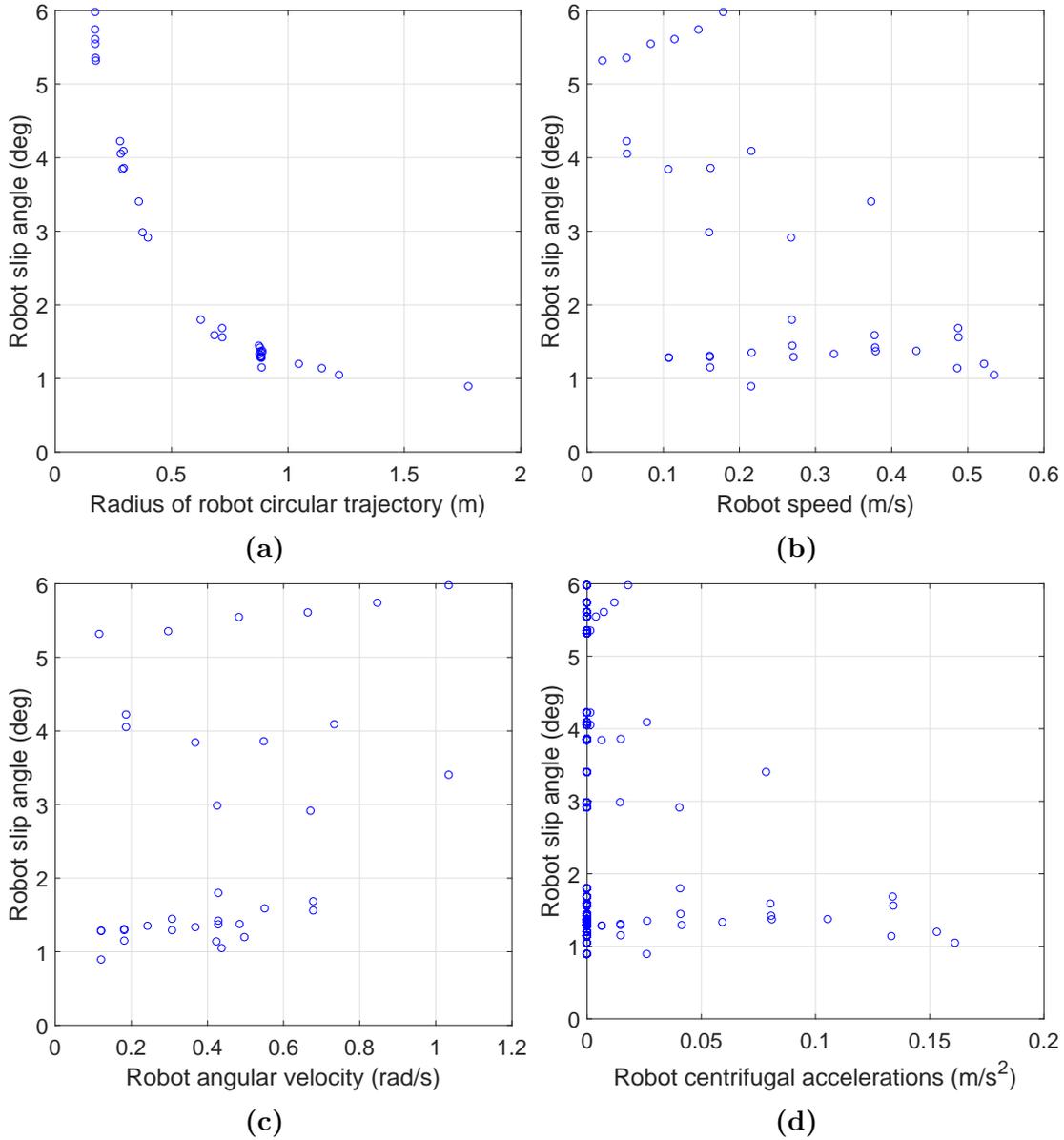


Figure B.6: Robot slip angle estimations of different datasets. (a) β vs. radius of trajectory. (b) β vs. robot speed. (c) β vs. robot angular velocity. (d) β vs. robot centrifugal acceleration.

References

- [1] Y. Pei and L. Kleeman, “Online robot odometry calibration over multiple regions classified by floor colour,” in *IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 2589–2596, IEEE, 2015.
- [2] Y. Pei and L. Kleeman, “Mobile robot floor classification using motor current and accelerometer measurements,” in *IEEE International Workshop on Advanced Motion Control (AMC)*, pp. 557–564, IEEE, 2016.
- [3] Y. Pei and L. Kleeman, “Robot calibration of sensor poses and region based odometry using offline optimisation of map information,” in *IEEE International Conference on Information and Automation (ICIA)*, pp. 462–468, IEEE, 2015.
- [4] Y. Pei and L. Kleeman, “A novel odometry model for wheeled mobile robots incorporating linear acceleration,” in *IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1396–1403, IEEE, 2017.
- [5] R. Kümmerle, G. Grisetti, and W. Burgard, “Simultaneous calibration, localization, and mapping,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3716–3721, IEEE/RSJ, 2011.
- [6] A. Martinez and E. Fernández, *Learning ROS for robotics programming*. Packt Publishing Ltd, 2013.

- [7] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Cambridge University Press, 2010.
- [8] M. Schwarz, T. Rodehutsors, D. Droeschel, M. Beul, M. Schreiber, N. Araslanov, I. Ivanov, C. Lenz, J. Razlaw, S. Schüller, *et al.*, “Nimbro rescue: Solving disaster-response tasks with the mobile manipulation robot momaro,” *Journal of Field Robotics*, vol. 34, no. 2, pp. 400–425, 2017.
- [9] Y. Okada, K. Nagatani, K. Yoshida, S. Tadokoro, T. Yoshida, and E. Koyanagi, “Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning,” *Journal of Field Robotics*, vol. 28, no. 6, pp. 875–893, 2011.
- [10] J. Forlizzi and C. DiSalvo, “Service robots in the domestic environment: a study of the roomba vacuum in the home,” in *ACM SIGCHI/SIGART conference on Human-robot Interaction*, pp. 258–265, ACM, 2006.
- [11] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart, “Long-term 3d map maintenance in dynamic environments,” in *IEEE International Conference on Robotics and Automation (ICRA)*, (Hong Kong, China), pp. 3712–3719, IEEE, May 3 – June 7 2014.
- [12] N. DAdemo, W. L. D. Lui, W. H. Li, Y. A. Sekercioglu, and T. Drummond, “ebug-an open robotics platform for teaching and research,” 2011.
- [13] D. Ball, B. Upcroft, G. Wyeth, P. Corke, A. English, P. Ross, T. Patten, R. Fitch, S. Sukkariéh, and A. Bate, “Vision-based obstacle detection and navigation for an agricultural robot,” *Journal of Field Robotics*, vol. 33, no. 8, pp. 1107–1130, 2016.
- [14] M. Bergerman, S. M. Maeta, J. Zhang, G. M. Freitas, B. Hamner, S. Singh,

- and G. Kantor, “Robot farmers: Autonomous orchard vehicles help tree fruit production,” *IEEE Robotics & Automation Magazine*, vol. 22, no. 1, pp. 54–63, 2015.
- [15] A. Kelly, “Fast and easy systematic and stochastic odometry calibration,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 4, pp. 3188–3194, IEEE, 2004.
- [16] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardós, “A comparison of slam algorithms based on a graph of relations,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2089–2095, IEEE/RSJ, 2009.
- [17] I. J. Cox, “Blanche—an experiment in guidance and navigation of an autonomous robot vehicle,” *IEEE Transactions on robotics and automation*, vol. 7, no. 2, pp. 193–204, 1991.
- [18] L. Kleeman, “Odometry error covariance estimation for two wheel robot vehicles,” *Intelligent Robotics Research Centre, Department of Electrical and Computer Systems Engineering, Monash University, Tech. Rep. MECSE-95-1*, 1995.
- [19] K. S. Chong and L. Kleeman, “Accurate odometry and error modelling for a mobile robot,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, pp. 2783–2788, IEEE, 1997.
- [20] A. Martinelli and R. Y. Siegwart, “Estimating the odometry error of a mobile robot during navigation,” 2003.

- [21] J. Borenstein and L. Feng, “Umbmark: A method for measuring, comparing, and correcting dead-reckoning errors in mobile robots,” tech. rep., UMMEAM-94-22, University of Michigan, 1994.
- [22] A. Censi, L. Marchionni, and G. Oriolo, “Simultaneous maximum-likelihood calibration of odometry and sensor parameters,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2098–2103, IEEE, 2008.
- [23] N. Roy and S. Thrun, “Online self-calibration for mobile robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 2292–2297, IEEE, 1999.
- [24] L. Kleeman, “Advanced sonar and odometry error modeling for simultaneous localisation and map building,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 699–704, IEEE/RSJ, 2003.
- [25] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [26] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, “Efficient sparse pose adjustment for 2d mapping,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 22–29, IEEE/RSJ, 2010.
- [27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [28] P. Giguere and G. Dudek, “A simple tactile probe for surface identification by mobile robots,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 534–544, 2011.

- [29] D. W. Haldane, P. Fankhauser, R. Siegwart, and R. S. Fearing, “Detection of slippery terrain with a heterogeneous team of legged robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4576–4581, IEEE, 2014.
- [30] K. Walas and M. Nowicki, “Terrain classification using laser range finder,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5003–5009, IEEE/RSJ, 2014.
- [31] C. A. Brooks and K. Iagnemma, “Self-supervised terrain classification for planetary surface exploration rovers,” *Journal of Field Robotics*, vol. 29, no. 3, pp. 445–468, 2012.
- [32] P. Filitchkin and K. Byl, “Feature-based terrain classification for littledog,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1387–1392, IEEE/RSJ, 2012.
- [33] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, “Terrain characterization and classification with a mobile robot,” *Journal of Field Robotics*, vol. 23, no. 2, pp. 103–122, 2006.
- [34] G. Best, P. Moghadam, N. Kottege, and L. Kleeman, “Terrain classification using a hexapod robot,” in *Australasian Conference on Robotics and Automation (ACRA)*, 2013.
- [35] L. F. Posada, K. K. Narayanan, F. Hoffmann, and T. Bertram, “Floor segmentation of omnidirectional images for mobile robot visual navigation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 804–809, IEEE/RSJ, 2010.
- [36] S. Verstockt, P. Lambert, R. Van de Walle, B. Merci, and B. Sette, “State

- of the art in vision-based fire and smoke detection,” in *International Conference on Automatic Fire Detection*, vol. 2, pp. 285–292, 2009.
- [37] C.-Y. Lin, C.-H. Yeh, and C.-H. Yeh, “Real-time vehicle color identification for surveillance videos,” in *International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pp. 59–64, IEEE, 2014.
- [38] F. S. Khan, R. M. Anwer, J. van de Weijer, M. Felsberg, and J. Laaksoinen, “Compact color–texture description for texture classification,” *Pattern recognition letters*, vol. 51, pp. 16–22, 2015.
- [39] K. Van De Sande, T. Gevers, and C. Snoek, “Evaluating color descriptors for object and scene recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.
- [40] F. Tungadi and L. Kleeman, “Multiple laser polar scan matching with application to slam,” in *Australasian Conference on Robotics and Automation (ACRA)*, 2007.
- [41] A. Hughes and B. Drury, *Electric motors and drives: fundamentals, types and applications*. Newnes, 2013.
- [42] C. J. Fisher, “Using an accelerometer for inclination sensing,” *AN-1057, Application note, Analog Devices*, 2010.
- [43] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, p. 5, Kobe, Japan, 2009.
- [44] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on*

Computational learning theory, pp. 144–152, ACM, 1992.

- [45] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [46] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [47] J. Lee Rodgers and W. A. Nicewander, “Thirteen ways to look at the correlation coefficient,” *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.
- [48] F. L. G. Bermudez, R. C. Julian, D. W. Haldane, P. Abbeel, and R. S. Fearing, “Performance analysis and terrain classification for a legged robot over rough terrain,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 513–519, IEEE/RSJ, 2012.
- [49] T. D. Larsen, M. Bak, N. A. Andersen, and O. Ravn, “Location estimation for an autonomously guided vehicle using an augmented kalman filter to autocalibrate the odometry,” in *SPIE International Conference on Multisource-Multisensor Information Fusion*, 1998.
- [50] A. Martinelli, N. Tomatis, and R. Siegwart, “Simultaneous localization and odometry self calibration for mobile robot,” *Autonomous Robots*, vol. 22, no. 1, pp. 75–85, 2007.
- [51] J. Maye, P. Furgale, and R. Siegwart, “Self-supervised calibration for robotic systems,” in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 473–480, IEEE, 2013.

- [52] J. Levinson and S. Thrun, “Unsupervised calibration for multi-beam lasers,” in *Experimental Robotics*, pp. 179–193, Springer, 2014.
- [53] M. Sheehan, A. Harrison, and P. Newman, “Self-calibration for a 3d laser,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 675–687, 2012.
- [54] M. Reinstein, V. Kubelka, and K. Zimmermann, “Terrain adaptive odometry for mobile skid-steer robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4706–4711, IEEE, 2013.
- [55] L. E. Ray, “Estimation of terrain forces and parameters for rigid-wheeled vehicles,” *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 717–726, 2009.
- [56] L. Kleeman, “Understanding and applying kalman filtering,” in *the Second Workshop on Perceptive Systems*, Curtin University of Technology, 1996.
- [57] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [58] E. B. Olson, “Real-time correlative scan matching,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4387–4393, IEEE, 2009.
- [59] A. Censi, A. Franchi, L. Marchionni, and G. Oriolo, “Simultaneous calibration of odometry and sensor parameters for mobile robots,” *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 475–492, 2013.
- [60] J.-L. Blanco, J.-A. Fernandez-Madrigal, and J. González, “A novel measure of uncertainty for mobile robot slam with raoblackwellized particle filters,”

The International Journal of Robotics Research, vol. 27, no. 1, pp. 73–89, 2008.

- [61] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: An efficient probabilistic 3d mapping framework based on octrees,” *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [62] A. Diosi and L. Kleeman, “Laser scan matching in polar coordinates with application to slam,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3317–3322, IEEE/RSJ, 2005.
- [63] R. Eberchart and J. Kennedy, “Particle swarm optimization,” in *IEEE International Conference on Neural Networks*, 1995.
- [64] M. Clerc, “Standard particle swarm optimisation,” 2012. (<https://hal.archives-ouvertes.fr/hal-00764996>, last accessed November 9, 2017).
- [65] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, “Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements,” in *IEEE Congress on Evolutionary Computation (CEC)*, pp. 2337–2344, IEEE, 2013.
- [66] M. Gendreau and J.-Y. Potvin, “Metaheuristics in combinatorial optimization,” *Annals of Operations Research*, vol. 140, no. 1, pp. 189–213, 2005.
- [67] J. Y. Wong, *Theory of ground vehicles*. John Wiley & Sons, 2008.
- [68] J. Pentzer, S. Brennan, and K. Reichard, “Model-based prediction of skid-steer robot kinematics using online estimation of track instantaneous centers of rotation,” *Journal of Field Robotics*, vol. 31, no. 3, pp. 455–476, 2014.

- [69] S. U. Lee, R. Gonzalez, and K. Iagnemma, “Robust sampling-based motion planning for autonomous tracked vehicles in deformable high slip terrain,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2569–2574, IEEE, 2016.
- [70] L. Ojeda, D. Cruz, G. Reina, and J. Borenstein, “Current-based slippage detection and odometry correction for mobile robots and planetary rovers,” *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 366–378, 2006.
- [71] A. Angelova, L. Matthies, D. Helmick, and P. Perona, “Learning and prediction of slip from visual information,” *Journal of Field Robotics*, vol. 24, no. 3, pp. 205–231, 2007.
- [72] L. Ding, H. Gao, Z. Deng, K. Nagatani, and K. Yoshida, “Experimental study and analysis on driving wheels performance for planetary exploration rovers moving in deformable soil,” *Journal of Terramechanics*, vol. 48, no. 1, pp. 27–45, 2011.
- [73] R. Balakrishna and A. Ghosal, “Modeling of slip for wheeled mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 126–132, 1995.
- [74] S. Konduri, E. O. C. Torres, and P. R. Pagilla, “Dynamics and control of a differential drive robot with wheel slip: Application to coordination of multiple robots,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 1, p. 014505, 2017.
- [75] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, “Collision avoidance and stabilization for autonomous vehicles in emergency scenarios,” *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1204–1216, 2017.

- [76] S. Song, A. Wong, J. Huissoon, and S. L. Waslander, “Autonomous vehicle control near the limit of friction,” in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1008–1013, IEEE, 2015.
- [77] N. R. Kapania and J. C. Gerdes, “Path tracking of highly dynamic autonomous vehicle trajectories via iterative learning control,” in *American Control Conference (ACC), 2015*, pp. 2753–2758, IEEE, 2015.
- [78] H. Pacejka, *Tire and vehicle dynamics*. Elsevier, 2005.
- [79] C. R. Carlson and J. C. Gerdes, “Nonlinear estimation of longitudinal tire slip under several driving conditions,” in *American Control Conference (ACC)*, vol. 6, pp. 4975–4980, IEEE, 2003.
- [80] A. Nizard, B. Thuilot, and R. Lenain, “Tire longitudinal grip estimation for improved safety of vehicles in off-road conditions,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3368–3373, IEEE, 2015.
- [81] S. Muller, M. Uchanski, and K. Hedrick, “Estimation of the maximum tire-road friction coefficient,” *Journal of dynamic systems, measurement, and control*, vol. 125, no. 4, pp. 607–617, 2003.
- [82] F. Tungadi and L. Kleeman, “Time synchronisation and calibration of odometry and range sensors for high-speed mobile robot mapping,” in *Australasian Conference on Robotics and Automation (ACRA)*, 2008.
- [83] A. Carballo, Y. Hara, H. Kawata, T. Yoshida, A. Ohya, *et al.*, “Time synchronization between sokuiki sensor and host computer using timestamps,” in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 261–266, IEEE, 2008.

- [84] H. AUTOMATIC, *Hokuyo HRG-04LX Specifications*, 2017. (<https://www.hokuyo-aut.jp/search/single.php?serial=165>, last accessed November 9, 2017).
- [85] L. Kneip, F. Tâche, G. Caprari, and R. Siegwart, “Characterization of the compact hokuyo urg-04lx 2d laser range scanner,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1447–1454, IEEE, 2009.
- [86] Y. Okubo, C. Ye, and J. Borenstein, “Characterization of the hokuyo urg-04lx laser rangefinder for mobile robot obstacle negotiation,” in *Society of PhotoOptical Instrumentation Engineers (SPIE)*, 2009.
- [87] J. Maye, H. Sommer, G. Agamennoni, R. Siegwart, and P. Furgale, “Online self-calibration for robotic systems,” *The International Journal of Robotics Research*, vol. 35, no. 4, pp. 357–380, 2016.
- [88] C. Garcia-Saura, “Self-calibration of a differential wheeled robot using only a gyroscope and a distance sensor,” *arXiv preprint arXiv:1509.02154*, 2015.
- [89] R. Gonzalez, F. Rodriguez, J. L. Guzman, C. Pradalier, and R. Siegwart, “Control of off-road mobile robots using visual odometry and slip compensation,” *Advanced Robotics*, vol. 27, no. 11, pp. 893–906, 2013.
- [90] Y. Tian and N. Sarkar, “Control of a mobile robot subject to wheel slip,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, p. 915, 2014.
- [91] H. Khan, J. Iqbal, K. Baizid, and T. Zielinska, “Longitudinal and lateral slip control of autonomous wheeled mobile robot for trajectory tracking,” *Frontiers of Information Technology & Electronic Engineering*, vol. 16, pp. 166–172, Feb 2015.

- [92] C. B. Low and D. Wang, “Gps-based path following control for a car-like wheeled mobile robot with skidding and slipping,” *IEEE Transactions on control systems technology*, vol. 16, no. 2, pp. 340–347, 2008.
- [93] S. Jung and T. C. Hsia, “Explicit lateral force control of an autonomous mobile robot with slip,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 388–393, IEEE, 2005.
- [94] J.-C. Fauroux and P. Vaslin, “Modeling, experimenting, and improving skid steering on a 6×6 all-terrain mobile platform,” *Journal of Field Robotics*, vol. 27, no. 2, pp. 107–126, 2010.
- [95] Y. Tian and N. Sarkar, “Formation control of mobile robots subject to wheel slip,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4553–4558, IEEE, 2012.
- [96] R. Taylor, L. Bashford, and M. Schrock, “Methods for measuring vertical tire stiffness,” *Transactions of the ASAE*, vol. 43, no. 6, p. 1415, 2000.
- [97] T. Chai and R. R. Draxler, “Root mean square error (rmse) or mean absolute error (mae)?—arguments against avoiding rmse in the literature,” *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.
- [98] MathWorks Inc., “MATLAB Software Package.” Webpage, 2017. (<https://au.mathworks.com/help/gads/index.html>, last accessed November 9, 2017).
- [99] S. Liu and D. Sun, “Minimizing energy consumption of wheeled mobile robots via optimal motion planning,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 401–411, 2014.

- [100] G. Fontana, M. Matteucci, and D. G. Sorrenti, “Rawseeds: building a benchmarking toolkit for autonomous robotics,” in *Methods and Experimental Techniques in Computer Engineering*, pp. 55–68, Springer, 2014.

- [101] S. K. Clark, *Mechanics of pneumatic tires*. US Government Printing Office, 1981.

- [102] S. Song, M. C. K. Chun, J. Huissoon, and S. L. Waslander, “Pneumatic trail based slip angle observer with dugoff tire model,” in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pp. 1127–1132, IEEE, 2014.

- [103] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, pp. 2743–2748, IEEE, 2003.