Robot Assisted Floor Surface Mapping and Modelling for Prediction of Grind Finish

by

Scott D. Wilson

Submitted to the School of Engineering and Advanced Technology
in partial fulfillment of the requirements for the degree of

Master of Mechatronics Engineering
Mechatronics

at

Massey University, Auckland

April 2018

Author . . . .
School of Engineering and Advanced Technology
April 14, 2018

Supervised by . . . .
Dr Khalid Arif
Professor Johan Potgieter

Robot Assisted Floor Surface Mapping and Modelling for Prediction of Grind Finish

by

Scott D. Wilson

Submitted to the School of Engineering and Advanced Technology
on April 14, 2018, in partial fulfillment of the
requirements for the degree of
Master of Mechatronics Engineering

Abstract

Mapping and localisation is a fundamental aspect of mobile robotics as all robots must successfully know their surroundings and location for navigation and manipulation. For some navigation tasks, prior knowledge of the 3D environment, in particular the 3D surface profile, can greatly improve navigation and manipulation tasks such as contacting, sensor inspecting, terrain traversability, and modifications. This thesis presents the investigation into the capability of cheap and accessible sensors to capture the floor surface information and assesses the ability for the 3D representation of the floor to be used as prior knowledge for a model. A differential drive robotic platform was developed to perform testing and conduct the research. 2D localisation methods were extrapolated into 3D for the floor capturing process. The robotic system was able to successfully capture the floor surface profile of a number of different type floors such as carpet, asphalt, and a coated floor. Two different types of sensor, a 2D laser scanner and an RGB-D camera, were used for comparison of the floor capture ability. A basic model was developed to estimate the floor surface information. The captured surface was used as prior knowledge for the model, and testing was performed to validate the devised model. The model performed well in some areas of the floor, but requires further development to improve the performance. Further validation testing of the system is required, and the system can be improved by improvement of 3D localisation, minimisation of sensor errors, and further testing into the application.

Thesis Supervisors: Dr Khalid Arif, Professor Johan Potgieter

Acknowledgements

I would like to thank a number of people that have helped me succeed in this project, and helped open doors to opportunities along the way.

To my mother and father, for their endless support, assistance, encouragement, and willingness to help me in any way, their love is greatly appreciated. To my girlfriend Caitlin, for her support through late nights, long days, and stressful times that has helped me persevere and pushed me to succeed, as well as encouraging me to enjoy the moment and take fun breaks. To my brother and sister, their pleasant distractions and laughs are much appreciated.

I would like to offer a massive thanks to my supervisors; Dr Khalid Arif and Prof Johan Potgieter. To Khalid, his insights and discussions has helped push me for excellence and I have learnt alot from him. To Johan, his guidance and vision has taught me alot, and I know I will continue to learn from him as years come by. In addition to this, I would like to recognise all of the staff and students at Massey University's School of Engineering and Advanced Technology for helping create a supportive and fun work / study environment.

A special thanks to Jason Torbet from Polished Concrete for his invaluable expertise and vision for this project, and for his assistance in funding this project.

Thanks to Massey University for the Massey University Masterate Scholarship, as well as the Massey College of Sciences Scholar Award that has helped fund my studies over the past year.

Thanks to my friends, for keeping me sane and providing distractions from my studies throughout the project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automation in construction is a growing field, particularly around reducing the level of manual operations and increasing productivity on site. Concrete floor polishing is typically a significantly manual process requiring workers to push heavy grinding machines along the floor. Recent innovations in grinding machine design provide remote control features that increase productivity [64, 55]; however, opportunities for improvement remain. The work is hazardous, producing concrete dust which is potentially damaging to the lungs, resulting in increased risk of silicosis, lung cancer, chronic obstructive pulmonary disease, and kidney disease [78, 4]. In addition, the strenuous labour involved can result in injury. Due to the repetitive nature of the grinding process, this task has the potential to be automated, thus eliminating hazardous conditions and the need for hard labour, and potentially providing a significant increase in productivity. However, before the grinding process can be automated, there are a number of research areas that require consideration. In particular, capturing the profile of the floor to highlight areas to be ground (for accurate flatness control), machine localisation and control, and a control algorithm that can accurately control the grinder in the way an expert operator would.

1.1 Motivation

The motivation behind this research is to identify research areas that can be applied to the automated grinding process and to gain further understanding in these areas regarding this particular application. The grinding process is heavily manual and hazardous, and so automating the task can help reduce exposure, harm, and injury to operators. In addition, there is the opportunity to increase productivity whilst decreasing the physical load on operators. For this project, two main areas require consideration: how to capture floor surface profile information, and how to use this information to predict and eventually control the grinding machine to achieve accurate flatness.

Current methods for scanning large areas are often expensive and take a considerable amount of time [81]. Whilst 3D laser scanners can provide large point clouds of an area, prices start at approximately $50,000 NZD and so make their use in some applications unfeasible. Therefore, there is an advantage in using cheaper, more accessible sensors for capturing the floor profile in a fast and reliable method. Our approach is to use two cheaper sensors: a 2D laser scanner for accurate positioning and Simultaneous Localisation and Mapping (SLAM), and a second sensor for creating a dense point cloud of the floor. The second sensor used was initially a 2D laser scanner followed by an RGB-D camera.

A basic grinding model is required for predicting material removal along an uneven surface. The literature contains many models that have successfully captured the grinding process; however, in many of these models the grind wheel is fixed in position, and so the cut depth is able to be estimated using probabilistic approaches and the known position of the tool (wheel) and workpiece. For floor grinding, the tool (the grinding machine) is not fixed; instead, it moves with the surface of the floor. This results in a non-uniform grind along the floor that is dependent on the floor surface itself. There is opportunity to devise a basic grinding model that can take the floor profile as an input and estimate material removal across that profile. This could provide information that could help control an automated machine to enable it to accurately grind a floor.

The aims of this project are to carry out fundamental research into how the floor surface profile can be captured, and how this information can be used for concrete grinding applications.

1.2 Objectives

To achieve the stated aim of this research, the following objectives have been identified:

- Study of existing methods for capturing 3D environment information.
- The development of a research platform to apply existing 3D capture methods using a fast and relatively cheap approach.

- Conducting of experiments to assess the capability of the research platform in capturing the floor surface profile.
- Study of grinding process modelling methods and investigation of how they could be applied to the complex concrete floor grinding process.
- The development of an appropriate basic grinding model.
- Conducting of experiments to validate the basic grinding model and robotic system developed.

1.3 Project Scope

The scope of this project is to develop testing methods and identify areas of further research required for the development of an automated concrete grinder or similar. The scope will be restricted to developing a research platform to be used for capturing floor profiles; an automated grinding machine will be considered out of scope for this project. The floor profile capture process will be limited to the use of cheap and accessible sensors for capturing the 3D profile. 3D Terrestrial Laser Scanners (TLS) will not be used due to the substantial cost involved. The grinding model developed will be used to estimate material removal based on constant speed and captured floor profiles. Further development of the grinding model into a more complex system to be used for control is out of scope for this project. In addition, this research focuses on grinding a concrete floor, and so the model will be limited to concrete material.

1.4 Thesis Outline

Chapter 1 of this thesis outlines the background information relating to this research, and the motivations, objectives and scope.

Chapter 2 describes relevant areas of research and reviews the current literature on these topics. An extensive review of the current state of scanning processes, modelling methods, and robotic research platforms is conducted. Opportunities for improvement are identified, in addition to how proven methodologies can be applied to the process under consideration.

Chapter 3 describes the development of a replica robotic research platform and how this platform can be used to scan and model a floor surface profile. Testing methods and initial results are also discussed. From the initial results, the chapter discusses a number of areas for improvement, and how these were then implemented on the robotic platform.

Chapter 4 discusses the development of a concrete grinding model that can aid further research into accurate control of the grinding process. Ultimately, the automated grinding process should be able to identify high and low areas of a floor and grind all parts of the floor to the required floor flatness standard. Improvements to the model and justification for the assumptions made are also discussed.

Chapter 5 discusses the experiments undertaken with the improved floor capture system and the validation of the concrete model. In addition, the experimental results are given.

Chapter 6 is a discussion and review of the methodologies used and the results found. Justification of decisions made throughout the project are included, and future improvements to the system are discussed.

Chapter 7 concludes this thesis by discussing successful aspects of this project and offers areas of required future work.

Chapter 2

Literature Review

2.1 Chapter Overview

The development of a mobile robot for accurate concrete grinding involves the intersection of many areas of research. For the mobile robot to act autonomously, research from autonomous mobile robotics, such as Simultaneous Localisation and Mapping (SLAM), must be utilised. SLAM is important because the robot must know where in the global frame it is in order to successfully perform accurate grinding. Prior knowledge of the floor surface profile can assist with control of the grinding action. Capturing this information requires research into 3D scanning techniques, methodologies, and 3D point cloud processing. Finally, a model of the grinding action is required both to calculate the optimum grinding path and for control so that the mobile robot may successfully achieve its task. This requires research into modelling methods, with particular interest in current grinding models and how they relate to the niche area of concrete grinding.

2.2 Robot Platform

A robotic platform is required to conduct experiments and further the research. The platform should be able to act autonomously as well as accommodate the required sensors and systems. There are many types of robotic platform that can be used, and each has its own advantages and disadvantages for various applications. The robot platform requirements can be divided into mechanical structure and software system.

2.2.1 Mechanical

A number of existing robotic platforms are often used for research applications. The three robots discussed here are; Turtlebot, the Pioneer P3-DX Robot, and the Pioneer P3-AT.

The Turtlebot (Figure 2-1a) is a differential drive robot designed as a research platform by Melonee Wise and Tully Foote at Willow Garage in November 2010 [32]. The robot is an Open Hardware Design and information required to ensure compatibility with the system is therefore provided under the Robotic Operating System (ROS) REP 119 specification [32]. The Turtlebot platform comes with a Microsoft Kinect RGB-D camera and is ready to perform spatial operations such as 3D mapping and navigation with ROS integration out of the box. The Turtlebot is used in various areas of research due to its modularity and ease of use; for example, 3D Mapping [21].

The Pioneer P3-DX Robot (Figure 2-1b) is a research platform designed for a modular setup with the ability to add, switch, or customize sensors and effectors. The platform has a differential drive setup with embedded controllers and ultrasonic sensors. Pioneer robots also come with an extensive C++ library called ARIA. ROS and MATLAB interfaces with the ARIA library are available, making the robot simple to set up and effective to modify. A laser scanner can be used for implementation of the full 2D navigation stack. The Pioneer robot has been used in many mobile robot applications [10].

Similar to the P3-DX, the Pioneer P3-AT (Figure 2-1c) is a modular research platform designed to greatly accelerate the development process in new research. A large variety of sensors such as vision, GPS, grippers, and obstacle avoidance can easily be added to the platform. The P3-AT differs in that it is four-wheel drive and thus uses skid-steer for control. Skid-steer is different to differential drive in that it relies on the slip of the wheels for turning, and thus, the kinematics of a skid-steer platform is more difficult to estimate [53]. Regardless, this platform has been a popular choice for research due to its versatility and ease of use.

2.2.2 Software System

A fundamental component of robot operation is the software used. The software architecture is an important aspect that must be considered when designing robots, as it can either accelerate or greatly decelerate development time. The architecture and software used often

(a) Turtlebot [32] (b) Pioneer P3-DX robot [2] (c) Pioneer P3-AT robot [1]

Figure 2-1: Common robot research platforms

depends on the application and the hardware used. Industrial robots tend to have their own manufacturer's software and operating procedures, while that for less-developed mobile robots is largely up to the developer. For some applications, embedded systems and micro-controllers are sufficient control systems for the robot; however, these are limited to basic tasks and little computation. A Central Processing Unit (CPU) is often required for increased computational capability and complex tasks such as localisation.

A common software system used in robotics is the Robot Operating System (ROS) [58]. Developed by Willow Garage in 2007, it breaks down into a number of key components: each task is attributed to a node; nodes can communicate via topics and services; and these topics and services are easily remapped to create a highly modular environment. This highly modular environment is particularly useful for research applications where the methodology, hardware, and algorithms used must be flexible. Due to the high degree of modularity, ROS has grown to have a number of community contributors adding to, and supporting, the network. This provides a large spectrum of applications and software available for new users, greatly accelerating development time. ROS provides a complete system, in that robot control can be managed completely from within ROS with relative ease.

A number of other robot software systems, such as MATLAB and LabVIEW, provide the required computational and analytical capability for robot control, such as kinematics and path planning; however, these are not a complete, standalone system such as ROS.

2.3 3D Scanning Technologies

In order to accurately control the grinding process, a 3D representation of the floor surface profile is required as prior knowledge. This requires determination of the 3D geometry of the floor as well as the global position relative to the robot coordinate system. There are a number of different technologies available for creating a 3D map of an environment. These technologies fall into two main categories: 3D sensors to capture the map at once or moving a 2D sensor to capture the map successively. Both of these methods can create detailed 3D depictions of the environment; however, they each have their advantages and disadvantages. The various technologies available will be discussed in detail in the subsequent sections, followed by some conclusions about their suitability for the current application.

(a) Faro 3D laser scanner [44] (b) Sick LMS291 2D laser scanner [3] (c) RealSense D435 RGB-D camera [45]

Figure 2-2: Example sensor technologies

2.3.1 Terrestrial Laser Scanner

A Terrestrial Laser Scanner (TLS) (Figure 2-2a) uses either time of flight or phase-based laser scans to create a 3D point cloud of the environment. A time of flight laser works by sending a laser beam to the object and measuring the time for the beam to return to the sensor [65]. Phased-based scanning emits a constant beam of laser energy and measures the phase shift of the returning energy which it uses to calculate the distance [65]. From the distance measurements, the x, y, z position of a measured point can be calculated by recording the yaw and pitch of the TLS system. TLS systems can provide many data points quickly, with some reaching 700,000 points per second [47]. In addition, some laser scanners can achieve sub-millimetre accuracy.

TLS systems have successfully been used to map large areas [15, 72, 7, 47]. The TLS output results in dense point clouds of large areas that can then be processed for application, such as terrain traversability [47] and flatness analysis [7]. However, due to their high specifications, these sensors are often very costly and thus not suitable for many mobile robotic applications. In addition, there is a direct trade-off between increased CPU computational capacity and the large number of points required for high resolution. Some TLS laser scanners cannot continuously scan and so require a Stop and Go approach, which is a limiting factor for some applications. This is a particular issue when a mobile robot is moving at speeds greater than 0.5 ms^{-1} where the points begin to suffer from Doppler shifting due to the robot velocity combined with scan time. Krusi et al [47] overcame this limitation through software that corrected individual points based on known robot motion. Whilst TLS provides a robust method for 3D mapping and has been proven through a number of studies, the high cost of these sensors introduces commercial challenges that render it difficult to implement in a number of applications.

2.3.2 2D Laser Scanner

A 2D laser scanner (Figure 2-2b) uses similar technology to that of a 3D TLS but only provides a single scan line. Because of this, 2D laser scanners can have similar specifications to 3D TLS systems but are often substantially cheaper, at around US $1000-6000 [3] compared with around US $50,000 for an accurate 3D laser scanner. A 3D map using the 2D scans can be created through two methods: rotating the laser scanner and recording the rotation and scan information to be stitched together, or by moving the 2D laser scanner through the environment and recording position, orientation, and scan information. Due to the affordability of these laser scanners, they have been a popular choice of sensor in mobile robotics over the past 10-20 years, particularly for 2D SLAM [26, 40, 29, 8]. 2D laser scanners are a proven method for creating maps through algorithms such as OpenSlam's Gmapping [36, 35].

2.3.3 RGB-D Camera

An RGB-D camera (example Figure 2-2c) is a sensor that provides both RGB colour images and per-pixel depth information. RGB-D cameras can be relatively cheap 3D sensor options when compared with laser scanners. There are two main types of these cameras: structured light and time of flight [24]. In the structured light type, the sensor projects an infrared speckle pattern onto the target (for example PrimeSense [79]). The pattern is then captured by an infrared camera in the sensor and compared with reference patterns stored in the device. These known patterns can give an indication of the measured depth. The system then estimates the per-pixel depth based on the reference patterns. This type of sensor is often combined with stereo sensing. The second type uses time of flight sensors to measure the per-pixel depth and associates each generated depth estimate with the RGB pixel.

Henry et al. [41] used a PrimeSense [79] RGB-D camera to create dense 3D maps of an indoor environment. They found that accurate integration of the depth and the colour images can help to provide robust frame matching and loop closure, which is particularly useful for SLAM. Endres et al. [30] utilised an RGB-D sensor only to capture the 3D environment, without relying on odometry or external localisation at all.

Depth data can be correlated with the RGB camera, yielding an RGB image with a depth associated with each pixel. This is often represented as a depth cloud and converted into a point cloud. The alignment of RGB image with depth means that these sensors can be used for object recognition [10]. However, while they provide fast and cheap 3D point clouds of the environment, they often lack the accuracy, resolution, and range as compared with other technologies. In addition, the use of infrared light renders the cameras susceptible to noise and erroneous readings due to changes in lighting. Recent advancements in sensor technology allow the exposure to be controlled automatically and thus minimise measurement errors due to changes in lighting [45].

2.3.4 Optical Interferometry

Optical Interferometry uses the projection of structured laser light to study the surface of a target. It is often used for the analysis of small parts or objects and has incredible accuracy and resolution. However, Optical Interferometry can be highly sensitive to vibration and temperature, which renders it difficult to implement outside of a controlled lab environment. Gao et al. [33] investigated two methods for minimising the errors due to vibration when using interferometry to measure surfaces in process. The sensor system was able to provide a highly detailed surface analysis of the target object, but had a too small a field of view to be useful for large-area applications.

2.3.5 Photogrammetry

Photogrammetry uses a series of 2D images to form 3D objects using mathematical relationships and triangulation of the images. A number of methods can give insight into the 3D shape of objects, such as shape from shading, shape from texture, shape from specularity, shape from contour, and shape from 2D edge gradients [62]. This method is often low in cost and the instruments are quite portable. However, photogrammetry can fail to capture flat, recurring surfaces such as walls and ceilings with no texture. The photo-matching methods rely on features within the images to create the 3D models, and some practical applications may lack the required level of features. This could be the case with concrete grinding, where the walls and floor could appear relatively consistent and thus may lack significant features to build from.

2.3.6 mmWave Radar

Radar is becoming more popular in autonomous driving applications due to its robustness and cost-efficiency [67, 61]. mmWave radar is robust to lighting, temperature, dust and humidity changes, making it a reliable sensor in difficult environments. In addition, radar can penetrate several non-translucent materials, such as thin walls, fog, and snow [67]. Laser scanning is sensitive to dust and thus can give inaccurate results. Similarly, photogrammetry and RGB-D cameras are sensitive to lighting. Belaidi et al. [11] utilised a mmWave radar due to its robust features and created a 3D scan of the terrain for optimal path planning and traversability analysis.

2.3.7 Contact-based Measurement

In contrast to the non-contact measurement methods discussed above, stylus-type profilometers can provide 2D and 3D surface profiles. Diamond tip styluses are commonly used in industry for measuring quality and conformity of parts. The main advantage is the robustness of the process: the tip requires contact with the part for measurement, eliminating the distortions or occlusions that occur with optical measurement methods. However, there remains a systematic error between the real contact point and the measured contact point related to the radius of the stylus tip. Whilst these methods can measure with high accuracy, they require the stylus to be moved over the whole surface; for large surfaces, this could result in long scan times or arrays of styluses to measure a larger area at each pass.

2.3.8 3D Sensor Summary

Accurate 3D surface profiles can be achieved using a number of different technologies. Contact-based approaches can achieve high-accuracy and high-resolution 3D profiles in a controlled environment, but are challenging to apply to large dynamic environments. Optical Interferometry can also achieve high accuracies and incredible resolution, however, can be sensitive to vibrations and temperature. 3D TLS systems do not suffer from such sensitivity issues and can be deployed in a variety of environments; however, these systems are often very costly and thus unfeasible for some applications. Cheaper options such as 2D laser scanners and RGB-D cameras require additional work, such as moving the sensor through the environment, to create an accurate 3D representation.

2.4 3D Scanning Methods

Prior knowledge of the 3D geometry of a floor surface does not provide sufficient information for improved navigation: the floor surface must be related to a global map that the mobile robot can access and relate to. This can be achieved either through having a 3D map of the environment, or by registering the points of the 3D surface to a 2D map of the environment. 2D SLAM is a well-understood area of research in mobile robotics and makes use of a number of well-understood techniques [26, 40, 29, 8]. It can provide a robot with an accurate 2D map for localisation within the environment. 3D SLAM can be achieved through similar approaches, although it is more complicated and thus requires exponential levels of computation [23]. 3D SLAM would be required to create a full 3D map for use by a mobile robot. Methods that utilise cheaper sensors and cheaper computation would be favourable over 3D SLAM. A thorough assessment of the development of SLAM algorithms over the past two decades can be found in the work of Cadena et al. [19].

In the literature, a number of approaches have been used to create a 3D map of the environment. These fall into three main areas: stationary scanning, Stop and Go, and continuous mapping. In stationary scanning, the measurement device is used whilst stationary and often in a single position. In Stop and go scanning, the scanner is moved through the environment and stops at predefined points to perform a scan. Finally, in continuous scanning, the sensor moves through the environment continuously capturing data and does not have to be stopped to perform a measurement. These different methodologies are discussed in detail below.

2.4.1 Stationary Scanning

A stationary sensor can capture a large amount of information and, due to it being stationary, removes a number of sources of error that occur in other methods, such as vibration and localisation. However, stationary sensors are limited in what they can see, often by line of sight, and in complex environments they can fail to capture all of the available information. Grzelka et al. [37] investigated the use of a stationary 3D scanner for analysing the surface roughness of a concrete slab. The system was able to provide enough information to estimate the surface roughness of the areas. Olofsson et al. [56] utilised a 3D TLS to create 3D scans of tree stems and record height using the RANSAC method.

Mobile Laser Scanning (MLS) presents an alternative solution to traditional static terrestrial laser scanning that can increase efficiency and provide robust 3D representations of the environment. MLS does have some limitations: the mobile system must be accurately localised in the environment, bumps whilst in motion could cause errors, and the 3D resolution is often dictated by the velocity of the mobile robot and the sampling rate. Zlot et al. [81] suggest that a stationary 3D scanning system, whilst providing a large number of accurate points, relies too heavily on accurate surveyed sensor positions and expensive, time-consuming, systems.

2.4.2 Stop and Go

A common method for moving a sensor through an area to capture the total environment is Stop and Go. Stop and Go methods refer to moving the sensor in the environment and stopping to perform a measurement. The method can benefit from many of the advantages that stationary scanning provides, whilst also providing a more efficient solution. Stop and Go provides accurately referenced measurements without the vibration effects, velocity-shifted measurements, or localisation update errors that limit the use of some sensor technologies. The sensors used are often full 3D sensors and can take some time to perform the measurement.

Lin et al. [51] utilised a Stop and Go method to improve the efficiency of a traditional static 3D TLS. They concluded that the Stop and Go approach can provide a more flexible and faster mapping mode compared with TLS, while also providing the stability and high sampling density seen in TLS. Putz et al. [57] utilised a 3D laser scanner to scan uneven terrain for improved path planning using a Stop and Go approach; this method was used to help minimise measurement errors and ensure that accurate point clouds were created to be stitched together later. This was effective and provided detailed point clouds; however, as found by Lin et al. [51], this method resulted in increased scan time and this could make it impractical for some applications. Chow et al. [22] investigated fusing an Inertial Measurement Unit (IMU) with an RGB-D camera for assisting the localisation of a Stop and Go scanning solution.

2.4.3 Continuous Sensing

A faster method for thoroughly scanning an environment involves moving the sensor through the environment continuously. The sensor captures data whilst the platform moves, and the data are stitched together to form a 3D representation of the environment. In this methodology, the sensor can be either 2D or 3D as discussed previously. Wang et al. [73] investigated replacing expensive 3D laser scanners with two low-cost 2D laser scanners to create 3D representations of plant structures in indoor environments. The platform supplied accurate referencing and calibration which enabled each 2D profile to be stitched into a full 3D point cloud. The Wang et al. method proved to be a successful approach to reduce costs whilst providing sufficient accurate information for application.

Zlot et al. [81] utilised a similar approach to avoid using expensive 3D TLS. They utilised three 2D laser scanners to create a 3D map of a tunnel. One of the laser scanners was placed at 25-degrees to the vertical on a rotating platform with an industrial grade MEMS IMU. This laser scanner created a 3D map of the environment and was used for SLAM and localisation within the tunnel environment. The two other laser scanners were mounted vertically so that a 360-degree view of the tunnel was created. The sensor information was collected and stitched together to form an accurate 3D representation of the tunnel. This method maintained an even distribution of points along the scan line, which would not be the case with a single rotating 2D laser scanner.

A key issue with continuous mapping is accurate localisation. Each data collection frame must be referenced to a global frame so that they can be accurately stitched together. This was achieved by Zlot et al. [81] through 3D SLAM from a rotating 2D laser scanner. In contrast, Wen et al. [76] utilised 2D SLAM for positioning using a 2D laser scanner, significantly reducing computational complexity. Banica et al. [9] used two sets of laser-based imaging systems, spatially correlated through the use of proximity sensors, odometry, and geolocation. Both methods were successful in localising the robot, but varied greatly in the computation required; this could be a significant limitation for some applications.

A common problem that must be considered in continuous sensing is Doppler-shifted measurements due to the motion of the robot. Droeschel et al. [27] utilised a 3D range finder to approach the problem of simultaneous localisation and mapping. The modelling approach used surface elements to provide efficient and accurate registration of points. A Hokuyo laser scanner was used on a rotating platform with an IMU that compensated for motion during scan acquisition; the mapping was performed continuously. Krusi et al. [47] also recognised the importance of removing distortion from laser data captured during movement, particularly when using a sensor with a long measurement time. Their approach involved correcting measurements from a 3D TLS based on velocity and position information of the moving robot platform. This helped create accurate 3D point clouds of the environment that could then be used for terrain traversability and path planning.

2.4.4 Sensor Fusion

As discussed above, many of the sensors have both advantages and disadvantages. Some of these disadvantages can be overcome, or their errors can be minimised, through the use of sensor fusion. In sensor fusion, two or more sensors gather similar information which is then fused together, minimising errors. Wen et al. [76] fused a 2D laser scanner with an RGB-D camera to produce a 3D indoor map. Sensor fusion was used to help improve a previously identified difficulty of insufficient overlapping frames. Wen et al. [76] utilised a fusion-iterative closest point method to align frames consecutively. The 2D laser scanner was used for localisation alongside odometry data. From the localisation, the RGB-D camera data could be aligned and stitched together, creating a 3D map.

Chen et al. [21] fused an IMU with Visual SLAM information using an Extended Kalman Filter (EKF) to estimate the pose of a robot in indoor environments. From this known pose, 2D laser scans were stitched together to create a 3D map in real time. This solution is limited in that it relies heavily on visual odometry, which can fail if there is a lack of features in the environment. The fusion of the IMU can help overcome this limitation, as can the fusing of additional sensors, such as wheel odometry. Chow et al. [22] similarly investigated fusing an IMU with an RGB-D camera for assisting the localisation of a Stop and Go scanning solution.

2.5 Data Processing

Once the surface information has been gathered, it must be processed. While the processing requirements are largely dependent on the particular application, there are a few general methods that can be applied. The data processing in this study will be restricted to determining high points, low points, general slope, and flatness. An aspect of data processing that is widely discussed by researchers and is particularly application-specific is how the data are represented and stored. Storing the entire data can result in huge datasets that are not manageable and may not be efficiently processed. Various data representation techniques used in the field of point clouds, surfaces, and 3D environment are reviewed and analysed below.

Once the data is ready for processing, a number of different methods can be applied. The driving factors are the available computational capacity and the end result. Data processing can be used to aid decision making, such as those relating to terrain traversability and surface grinding.

2.5.1 Surface Representation

Processing large point clouds of a surface is possible, but is computationally expensive due to the large number of data points. A number of methods have been implemented to reduce the computation required for processing large point clouds, relating in particular to terrain traversability and better representation of surfaces. These aspects are often used for planning in uneven and rough terrain.

Endres et al. [30] overcame the limitations of pure point cloud representation by using a 3D occupancy grid approach. The grid utilised the Octree-based mapping framework Octomap [42]. This tree structure provides an efficient way to represent the voxels of a 3D environment, and inherently allows the map of that environment to be stored at multiple resolutions. Common 2.5D approaches are memory-efficient; however, Endres et al. [30] found that they were insufficient for storing a complete environment map and can miss some data. The Octree structure retains the benefits of graph-based data representation where data can be accessed and stored efficiently. Putz et al. [57] built upon previous research into graph-based surface representation to create a triangular mesh of an environment. A triangular mesh representation is suggested to provide a flexible solution that can be used for multiple purposes such as environment representation [38], visualisation of human robot interaction, and ground-based path planning.

Krusi et al. [47] used raw point cloud measurements for surface representation and utilised a graph-based submap approach to greatly reduce computation, particularly for navigation of large 3D environments. The number of points in a submap was restricted before being merged into the graph structure. This reduced computation whilst maintaining a high level of resolution and accuracy. The graph-based approach provides a rapid method for searching a total environment for an efficient path to a goal, and additionally provides a computationally efficient way of updating the map, simply by updating the current submap. This reduces the need to compare the point cloud with the enitre environment representation. The graph approach for data representation is said to be a highly scalable system [16]. The submap approach also reduces the effect of localisation drift on the global map.

A common surface representation in the recent literature involves the use of multi-level resolution surfaces; the surface is represented by a number of different resolution surfaces, in particular high resolution near the robot and decreasing resolution with distance from the robot. This is beneficial for a number of applications. In 3D localisation utilising feature matching, low-resolution maps can be used to quickly reduce the possible locations of the robot. The high-resolution maps can then be used to refine the robot's location [27]. This reduces computation as the large number of points is only used in a small area. If the large number of points is used for the entire point cloud area, this would require expensive computation in order to match the features correctly.

Droeschel et al. [27] used a surface description approach. First, a 3D laser scan is converted into points in a map. These points are then stored in a multi-resolution grid structure of the map. Cell size increases with distance from the robot, and so reduces the large number of cells traditionally required. Each grid cell is then represented by a surfel that summarises the 3D points in the cell. The surfel provides an indication of the position and orientation of the points at each grid cell position. Stoyanov et al. [68] used the Normal Distribution Transform to create similar surface descriptions. This provided an easy method for distinguishing between floor, walls, and obstacles from the 3D map of the environment, which could then be used for navigation and path planning (Figure 2-3).

Figure 2-3: Surfel surface representation of a 3D room [68]

2.5.2 Processing for Surface Analysis

In addition to storing and representing the captured 3D information, it must be processed for the specific application. For concrete grinding, the information must be processed to evaluate the floor, particularly its overall flatness. Useful information for concrete grinding include determination of the high and low areas of the floor and any rolling undulations in the floor. These undulations can have a number of different frequencies, and can be seen as a slight change in floor height over a short distance (e.g. 1 m), or even a change in floor height over a long distance (e.g. 10 m). The approaches discussed below are Straight Edge, Waviness Index, and Wavelet Filter.

2.5.3 Straight Edge Method

The Straight Edge Method is the simplest and oldest method of analysing surface flatness [18]. It involves placing a 3 m long straight edge on the floor at random locations. The deviation of the floor from the straight edge is measured, and if it is less than a tolerance value then the floor is determined to be in specification. This method is simple and cost-effective, but is prone to errors and only measures the surface waviness at 3 m wavelength.

2.5.4 Waviness Index

The Waviness Index (WI) is a quantitative analysis for determining floor flatness for meeting building specifications [46]. Measurements are conducted at 1 ft (30 cm) intervals along floor survey lines, and deviations are calculated from the midpoints of imaginary chords defined by pairs [6]. The WI is measured at five intervals (60, 120, 180, 240 and 300 cm) and so covers more frequencies of undulation than the straight edge approach. An advantage of WI is that it expresses the deviation from flatness as a measurement unit (inches or centimetres) and so is simple to comprehend; however, measurements are only performed along the 1D survey lines. The WI could therefore miss undulations of relatively short period (less than 60 cm) [14]. In addition, the WI is difficult to automate, which would limit its use in robotic applications.

2.5.5 Wavelet Transform

One method of surface analysis involves applying a Wavelet Transform to the data. A Wavelet Transform is a signal analysis method that is based on convolution of an input signal with the wavelet function at different locations of the signal at multiple scales. This means that a signal pattern of the wavelet function can be detected at any scale and any location. For a floor, this can enable the detection of different patterns and frequencies of undulations, ultimately providing a characterisation of the surface waviness.

Valero et al. [72] used a 2D Continuous Wavelet Transform (CWT) to process 3D TLS data for analysing the surface flatness of a concrete floor. The 2D CWT correlated well with results from the current state-of-the-art waviness characterisation method, the Waviness Index (WI). They found that the 2D CWT was able to automatically identify several areas of deviation from flatness, across a range of wavelengths, meaning that the method could identify short peaks in the floor as well as gentle undulations across it. Previous manual methods, such as WI, struggle to achieve this range of analysis. Bosche et al [14] found similar results when applying a 1D CWT to two different floors scanned by a terrestrial laser scanner. When combining all five undulation periods tested, they found a strong correlation between the methods, with an R-squared of 0.84, although this was not as strong as Valero et al.'s [72] value of R-squared of 0.96. These differences in correlation could be explained by the sampling methods used in the WI approach, as the measurement sampling can lead to inaccurate or even failed detection of undulations at specific periods, similar to aliasing. Additionally, Bosche et al. [14] found that due to the CWT output, it was possible to identify both concave and convex undulations in the surface.

Alhasan et al. [6] evaluated the performance of two algorithms for processing 3D Stationary Terrestrial Laser Scanned point clouds into surface maps that characterise roughness. The advantage of these algorithms is that they can provide a 2D analysis of the surface rather than traditional analysis along survey lines (1D).

Tang et al. [70] discussed three methods for analysing surface flatness, following a similar three step approach: set up a reference frame; smooth noise and calculate deviations between points and reference; and identify regions deviating from the reference by more than a threshold value. The data was collected by a stationary 3D laser scanner and was able to reveal deviations from flatness in the surface; however, further research into the effects of incident angle and other methodology factors is required.

2.6 Modelling Methods

Once a floor profile has been captured, the surface must be analysed to identify the optimum grinding specifications. This can be achieved by using a mathematical model.

2.6.1 Grinding model

Grinding is a process that involves many different factors. Research into grinding is vast, and many findings do not apply to concrete grinding, in particular floor grinding. However, some researchers have been successful in mathematically modelling the grinding action [69, 5, 52, 80, 49, 54, 66, 25]. Due to the complexity of the grinding process and the high number of variable factors, many models aim to reduce the complexity of the process to capture what they consider to be important aspects. A common approach to analysing the grinding process is the single grit approach [66, 60]. It is suggested that by analysing the performance and forces involved in a single grit of the grinding process, the model is greatly simplified. Once the single grit approach can be validated, the model can easily be extended to accommodate the entire grind wheel.

Single Grit Analysis

In general, grinding involves the sequential motion of individual grits cutting at a micron-level depth, which leads to a macroscopic-level removal of material [66]. Therefore, the grinding performance of a grinding wheel can be described by the behaviour of an individual grit. As a result, overall grinding performance is largely influenced by the number of active cutting edges (individual grits), and so the estimation of these active cutting edges is an important aspect. An exact determination is not possible due to a number of stochastic variables such as the distribution of the grits and their position, number, and shape.

Ma et al [52] investigated the dynamic behaviour of grinding, in particular the dynamic behaviour between the workpiece and the grinding wheel. Zhang et al. [80] analysed the cutting force and frictional forces of a single grain with respect to the stress state of the grain, and then verified the results using scratch tests. Grain shape is considered to have significant influence and Lee et al. [48] classified each shape into four categories: conical, spherical, rounded table, and four pyramid (Figure 2-4). Wang et al. [74] found that when the cut depth was large, the grains could be regarded as a conical shape, which has become commonly used in the literature. Zhang et al. [80] used a conical shape to distinguish grains that were active compared with grains that were not (Figure 2-5).

Figure 2-4: Grind grain shape [80]

Figure 2-5: Active grits based on depth of protrusion [80]

Stochastic Properties

Grinding is a stochastic process, in that it is impossible to ascertain exactly which grits are in exactly what cutting phase at all times. However, probabilistic approaches can help predict the cutting action and grit behaviour of the grinding. Chang and Wang [20] introduced a stochastic grit distribution function to describe the random grit distribution in the rotating wheel. The dynamic grinding force was modelled as a convolution of a single grit force and the grit density function.

Setti et al [66] used the Rabinowicz abrasive wear model to estimate the active grits, allowing for the effect of depth of cut, grit size, and workpiece hardness to be considered simultaneously. The simulation of grinding is a complicated process; factors such as grits stochastic distribution, undefined geometry, and unknown number of cutting edges make the process difficult to analyse quantitatively.

A particular challenge of concrete grinding is that as the concrete is ground, the concrete dust (removed material) begins to act as an additional abrasive. This makes accurate modelling of the concrete grinding process very difficult. Weihs et al. [75] addressed this problem by acknowledging that discrete simulation methods such as finite models cannot be applied due to the abrasive nature of the concrete material. Their approach was to subdivide both the material and the diamond into Delaunay tessellations. The resulting micropart connections could then be interpreted as predetermined breaking points, resulting in improved analysis of the surface geometry and cutting edge interaction.

Similarly, Raabe et al. [59] proposed a geometrical simulation model that describes the forces affecting the workpiece as well as the chip removal rate and the wear rate of the diamond in process parameters. The model treats both the material and diamond grain as tessellations of microparts connected by predetermined breaking points. The process was then iteratively simulated, with the forces calculated by interpreting the collisions of pairs of workpiece and grain microparts as force impacts.

Grinding Action

Grinding can be related to traditional milling in that the feed of the material and the speed of the grinding wheel (or head) greatly affect the rate of material removal. However, grinding differs from traditional milling in the method of material removal. Traditional milling relies on chip formation due to the tool edge to remove material at a macro scale. Grinding involves the systematic removal of material at a micro scale resulting, in a change in material at the macro scale. The grinding process is often broken down into different phases of chip formation. Rasim et al. [60] utilised a previously developed single grain scratching method which enabled the observation of chip formation in situ with a high-speed camera. This provided insight into the determination of specific grain engagement depths and transition points for the material (hardened steel in this case). In the study, Rasim et al. [60] aimed to develop a quantitative chip formation model that took the grain shape as well as cutting speed and lubrication into account. Grain shape in both the direction of motion and the transverse direction has significant influence on chip formation.

Throughout the grinding process it is common for grains to be considered as sliding, ploughing, and cutting. Zhang et al. [80] divided grains into just two categories, cutting and ploughing, on the basis of experimental results revealing the boundary between sliding and ploughing to be fuzzy and hard to distinguish. Durgumahanti et al. [28] categorised grinding forces into cutting, chip formation force, frictional force, and ploughing force. As the cutting edges of abrasives come into contact with a workpiece, elastic deformation occurs. As they traverse further into the workpiece, this deformation continues. This phase of grinding is purely frictional.

Tang et al. [69] divided the chip formation energy used in model calculations into static energy and dynamic energy, which is mainly influenced by shear strain, shear strain rate, and heat in the removal process. This study was performed on metal, so cannot be directly correlated to concrete.

2.6.2 Physics Model

In order to determine the position of the grinding cutting edges with respect to a global coordinate system, the robot position must be known. On a perfectly flat 3D surface, the robot position is easy to determine from the robot's speed and orientation. This can lead to changes in x and y position and thus provides the new position of the robot. However, it is not always possible to operate on a perfectly flat 3D surface, and on an uneven surface the robot's position must be defined in terms of x, y, and z , as well as roll, pitch, and yaw. This is difficult to accurately determine for a grinding robot due to the changing points of contact.

2.6.3 Surface Dynamics

As a robot moves along a surface of the floor, the robot will move with 6 Degrees of Freedom (DOF). This is x, y, and z, and roll, pitch, and yaw. This is important for grinding because, unlike a CNC or milling machine, the 6 DOF of the cutting tool cannot be perfectly controlled. For a mobile robot grinding application, the cutting tool (diamond tool) will move with the surface, and therefore the diamond tool cutting position and orientation will be determined partly by the surface dynamics. The mobile robot dynamics along the surface can be captured in two ways: using a dynamic model of the surface, or through applying a physics engine to the model.

2.6.4 Model Validation

Often, a model simulation on its own is not sufficient; the model must be tested against real-world results to validate any assumptions and equations used. A common method of validation is to run a similar real-world test and find the error of the model compared with the real-world measurements [59, 54]. Li et al. [50] used confocal topography to confirm surface simulations, which provided an accurate method for analysing the scratches in the workpiece due to grinding. Guo et al. [39] compared the developed model predictions to experimental results using a stylus to measure the surface roughness.

2.7 Opportunities for Improvement

This review has revealed a number of opportunities to improve upon the existing methods and approaches and utilise known information for this particular application. The design of an automated concrete grinding machine requires several areas of development: the autonomous system, capture of the floor profile, and use of the known floor profile for optimised grind control. There are a number of findings in the literature that can be applied directly. ROS is a commonly used research platform due to its open-source nature and modular, rapid development structure. In addition, floor profiles for a number of different applications have been previously captured with some degree of success. Finally, models of the grinding process have been proposed, particularly from a grinding wheel perspective, and a number of the key concepts can be considered in the development of a floor surface grind model.

Common sensors used for creating accurate floor surface representations are expensive 3D terrestrial laser scanners, and thus are unfeasible for a low-cost application. There is therefore an advantage in identifying a fast and cheap method of scanning the floor, which can then be used as prior knowledge for a second task, such as automated grinding. Other applications could use this same methodology, such as cleaning and navigation of uneven terrain. In addition, identifying the limitations of some sensors for this type of application can benefit the research field.

Further advantage lies in using fundamental concepts to devise a model representing how a grinding machine removes material on a concrete floor based on the known floor profile. Little research has been performed in this area, although some concepts and approaches from related areas may be used. An opportunity exists to develop a basic grinding model that considers the floor profile as an input, along with traditional factors such as grinding head speed and machine cut speed, to predict the post-grind surface of the floor.

2.8 Chapter Summary

This literature review has extensively covered current research in the areas that are relevant to this project. Analysis of a number of robotic platforms has helped confirm design decisions for the research platform. Research platforms with differential drive provide the advantage of utilising wheel odometry to assist with localisation. In addition, the open-source Robotic Operating System (ROS) can provide a fast and modular framework to develop from. Discussion of a number of 3D scanning technologies has illustrated the variety of sensors available and their limitations, which can help aid decision making regarding sensor selection.

Researchers have used a number of different methodologies to create a 3D map of an environment. These methodologies fall into the categories of stationary, Stop and Go, and continuous scanning methods, with the further option of sensor fusion. Continuous sensing can provide an efficient method for mapping a large area, such as a floor to be ground, and errors can be minimised through the use of sensor fusion. Following 3D data collection, the information must be processed and represented in a useful way. The state of the art of floor flatness analysis pivots on the application of a Continuous Wavelet Transform to the floor, providing insight in two dimensions across a number of wavelengths. Extensive research into the grinding process has highlighted the complexity of the process due to a number of stochastic variables; however, assumptions and predictive approaches can help overcome some of these complexities.

A number of opportunities for further research have been identified. This project will consider the floor surface capture process and sensor limitations, and research into a basic grinding model that acts on the captured floor surface.

Chapter 3

Floor Surface Capture Platform

3.1 Chapter Overview

This chapter describes the development of a robotic research platform for testing the floor surface scanning and modelling capability. The research platform is designed to be of similar size and shape to a grinding machine to give insight into system dynamics. The development of the platform is discussed in terms of the mechanical, electrical, and software systems used. The algorithm for capturing the floor profile and for planning the coverage path is discussed. Initial tests of the floor surface capture system provides insight into further development and challenges. A number of the challenges are addressed and the methodology for testing as well as improving the system is analysed.

3.2 Research Platform Development

The floor surface capture system aims to utilise cheap and accessible sensors. This can be achieved through using one sensor to localise the robot and a second sensor to capture floor surface data. The sensor can be moved through the environment and the resulting captured data stitched into a 3D profile. In order to move a sensor through the environment to capture the required data, a moving platform is required. This platform must be relatively robust, and provide adequate support for the sensors to be mounted on it. Continuous scanning of an environment also requires accurate localisation of the robot. The robotic platform must therefore be capable of localising whilst capturing the floor surface profile data.

3.2.1 System Requirements

The research platform is designed to be a replica of the target grinding machine for implementation. Accordingly, the system does not carry excessive weight from the grinding motors whilst still providing insights into the dynamics of the size of the system. This allows the robotic platform to be easily portable and enable fast development of control, as well as efficient testing of systems such as floor scanning.

3.2.2 Mechanical System

The robotic platform (Figure 3-2) is a differential drive robot with two drive motors at the rear of the machine. The front of the platform is supported by a castor wheel. The robot has two levels: one providing a base for a horizontal laser scanner for SLAM, and the other holding components for control and communication and providing a mount for a second sensor to scan the floor. Structurally, the system is of similar dimensions to a typical grinding machine.

An adjustable sensor mount (Figure 3-1) was developed to provide the means for measuring a floor profile at various angles and with different sensors. The mount was designed to be relatively universal and sufficiently strong to hold a variety of floor scanners in position as the robot moves around the room. The mount was designed to hold components weighing at least 6 kgs and could easily be adjusted 180-degrees in pitch and then locked in place. It was made out of 3 mm steel bent into shape. Figure 3-1 shows the adjustable mount with an Intel RealSense RGB-D camera on the underside and an IMU on the top. The camera is mounted on a second adjustable frame that can be manually tuned to ensure that the camera is level.

The motors are mounted directly into the supports of the frame using 4x M5 bolts. This mounting is sufficient for the test platform, but it would need to be strengthened for the final machine due to additional loading from the weight of the machine and grinding motors.

3.2.3 Electrical System

The electrical system consists of power distribution and communication connections. A power distribution board was designed to provide power to each component, protected by fuses and controlled through relays. The system power architecture is shown in Figure 3-3.

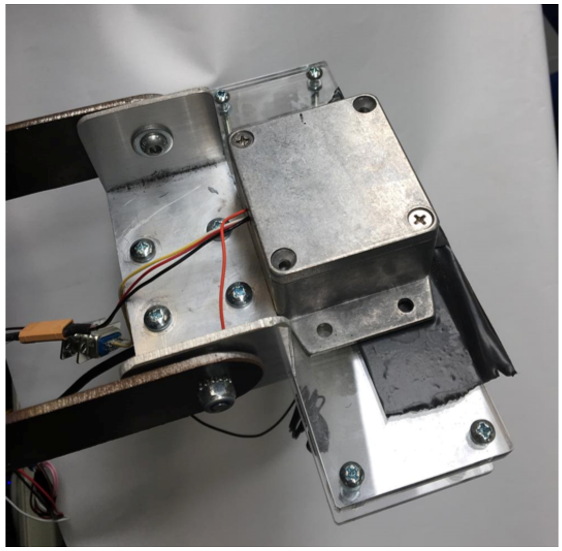


Figure 3-1: Floor sensor adjustable mount

The main power demands are 24V, 12V, and 5V. The 24V system is limited to a maximum of 50 A from the battery and is protected by fuses. Each 12V and 5V component has a fuse to restrict current , ranging from 0.5 A fuses to 5A fuses. Additional ports were supplied for future expansion.

A schematic of the power distribution board was designed using Circuit Studio. The board takes in 24V and provides 12V via a 24V to 12V converter. The 12V is then converted to 5V to power a microcontroller board (Arduino Uno) for relay control. There are 6 relays on the distribution board, 4 x 24V and 2 x 12V, which can be turned on and off from the Arduino. Each output port is protected by a fuse to help keep components safe. For a commercialised product, more-reliable control would be desirable; this could be achieved by a dedicated USB-controlled relay board or a PLC relay board. This component can easily be swapped out to achieve the required functionality at a later time.

3.2.4 Software System

The robot uses the Robotic Operating System (ROS) framework for internal communication and control [58]. ROS is an open-source system that allows for the creation of many nodes that can communicate efficiently through the use of topics and services.

The ROS system begins with a core, which provides the base communication framework. Nodes can be added to the system, which can communicate through the roscore using top-

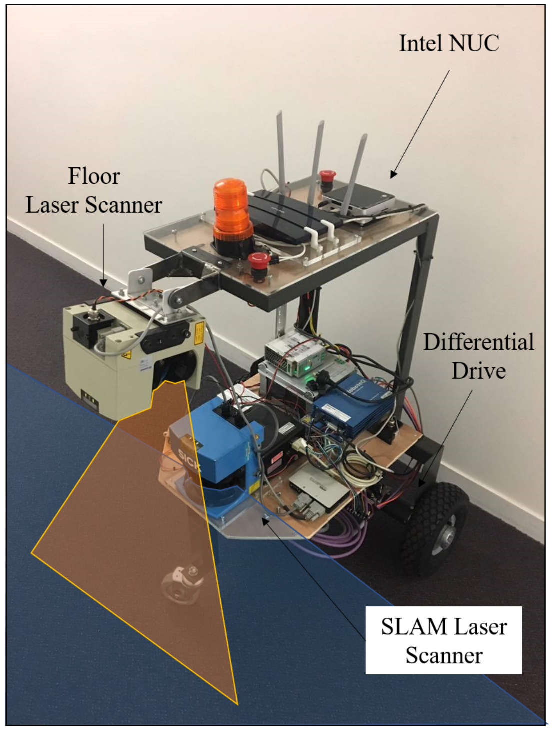


Figure 3-2: Diagram of the mobile robot platform highlighting the laser scanner position and orientation

ics and services. Any node can publish or subscribe to any topic or service, providing a highly modular system. Due to the open-source nature of ROS, a community has provided a number of existing solutions to common problems, such as Adaptive Monte Carlo Localisation (AMCL), Gmapping, and SLAM. This results in an efficient and proven framework. ROS was selected as the software framework because of its open-source nature, the modularity provided by nodes, and the ability to accelerate development using existing solutions. The ROS system for the research platform requires a number of components; the general architecture is shown in Figure 3-4.

3.2.5 Floor Profile Creation

A ROS integrated package was developed to capture laser scans of a floor profile and assemble these into a point cloud that could then be analysed and used as prior knowledge. First, the raw laser scan data is filtered so that only the floor in front of the robot, spanning a 60-degree angle, is captured (Figure 3-2). These laser scans are transformed through the robot relative to the robot's base_link in the global space. As the robot moves through

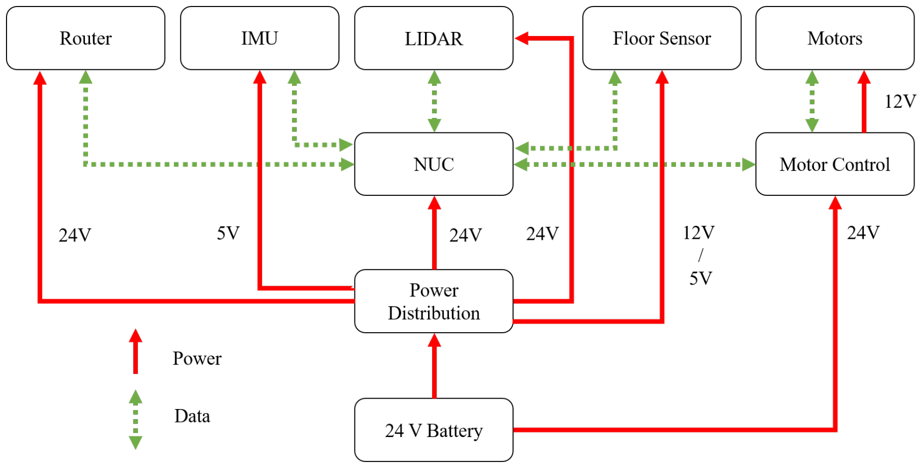


Figure 3-3: Electrical power and data connection block diagram

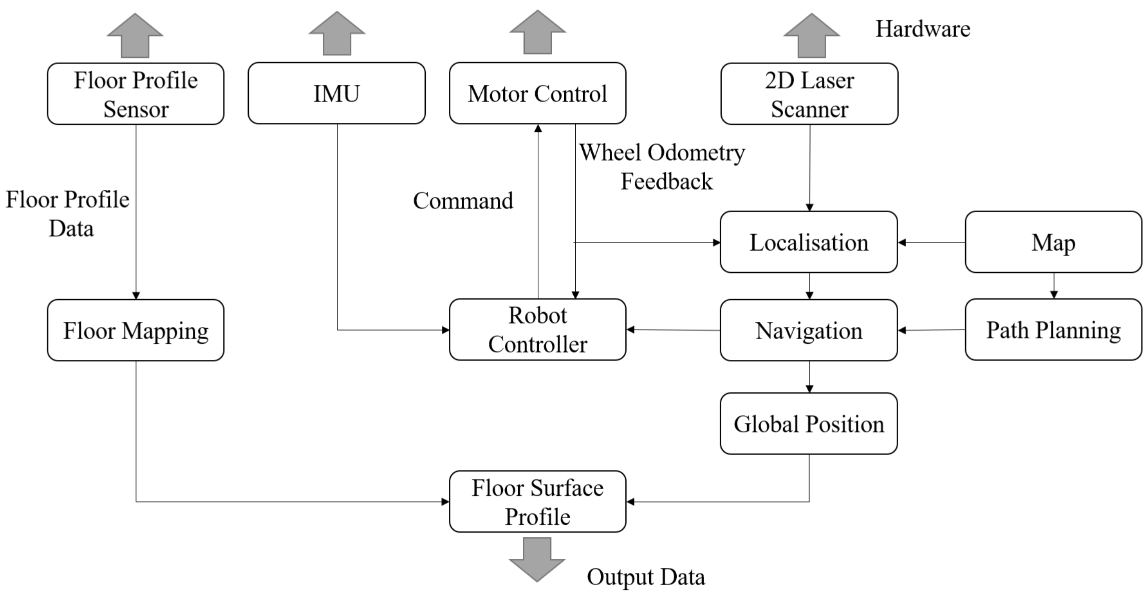


Figure 3-4: ROS system architecture

the environment, the base_link transform moves through the global coordinate system. This in turn moves the location of the laser scan and thus the laser scan data. Each laser scan provides a line scan of the floor profile at a point in the 3D global coordinate system. Assembling many of these single scan lines together therefore forms a series of lines and thus a surface of the floor profile. The assembled scans are captured by the laser_assembler package. A keyboard-controlled node calls the laser_assembler services to start and stop collecting data. Once the laser_assembler service is called to stop assembling, a point cloud of the assembled scans is published to the assembled_floor_scan topic. This topic can then be saved to a .pcd file for analysis. This is performed in real-time; however, the point cloud can only be viewed and analysed once the full scan process has been completed. Alternative profile creation algorithms will be required for different scanning methods, such as RGB-D; and this will be discussed in Section 3.6.5.

3.2.6 Path Planning Program

A program designed to automatically create a coverage path for a set area on a map was created using OpenCV [17] and ROS integration. The program can operate in two modes. Each mode takes in a ROS map description consisting of a .yaml configuration file and a .pmg image of the map. The first mode allows the user to input desired points for the robot to move to. These are displayed for the user on the map and the robot can then move to each point in the order in which they were placed (Figure 3-5). A point is added by left-clicking on the map; a previous point can be removed using the middle mouse button. Points can then be saved to a file by right-clicking. The points are saved to a text file relative to the global coordinate system of the map (robot-defined).

The second program mode allows the user to define an area on the map, and then automatically creates a path to cover this area. The program was developed with the intended application in mind, in that the machine drives in straight lines at a set, but variable, distance apart. This creates the desired degree of grind overlap for the grinding process. Accordingly, the angle of the robot's parallel paths, and the separation between each pass, can be set by the user. The coverage path is achieved through a series of parallel lines and intersections. A large number of parallel lines are drawn on the map, starting at the desired offset from the first position, at the user-defined angle, and at the user-defined spacing. The program then calculates the intersection of each of these lines, starting with the boundary lines defined by the user. Each point is stored in a list of the points required to cover the area. This method works well, however, does require error checking and correction. For example, the order of points is not considered by the initial program although this is crucial for successful coverage path generation.

Accordingly, a few steps were taken to consider the order of points.

- The first point of the path is chosen based on the closest point to the first point clicked on the map by the user.

- Each line created by the path generation is compared with the user-defined path angles.

- If a line is on a diagonal (beyond some pre-determined threshold), then the points must not be in the correct order and are therefore swapped.

A later version of this program was implemented in RQT. RQT provides an easy interface with the ROS system and can provide a platform with many different plugins running at a time, including diagnostics and data plotting. This may be useful in later development where many different aspects of the robot may need to be monitored at one time. A selection of paths created by the coverage generation program are shown in Figures 3-5 and 3-6. The interface allows the user to vary the spacing, angle, and offset of the generated path, and the path updates in real time to allow for fine control.

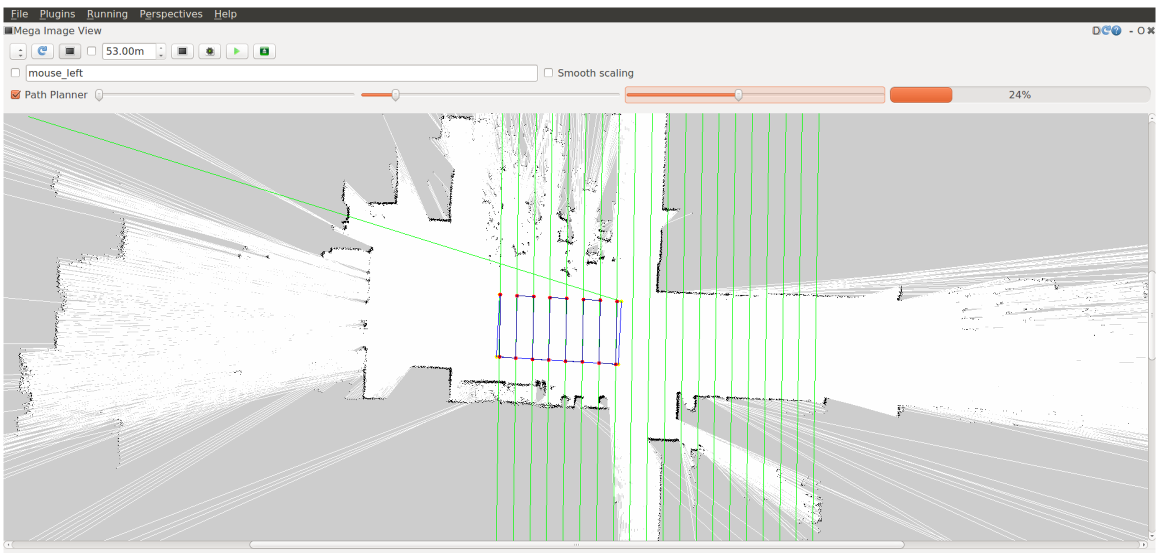


Figure 3-5: Created coverage path with vertical machine direction and small grind overlap

The goal positions created by either the user or the program are saved to a text file that can then be read by the waypoint management node for controlling the robot. Each waypoint is saved as a global x, y, z coordinate, as well as yaw for the orientation of the robot. For the 2D robot, the z coordinate is arbitrary. The waypoints generated from the program are required to be converted from the image coordinate frame into the robot's coordinate frame.

The robot's coordinate frame is determined when the robot is creating the initial map. 0,0 on the robot's frame is taken to be the lower left pixel in the map [34], whereas 0,0 on the image coordinate frame is the top left-hand corner. In addition, yaw is considered to be

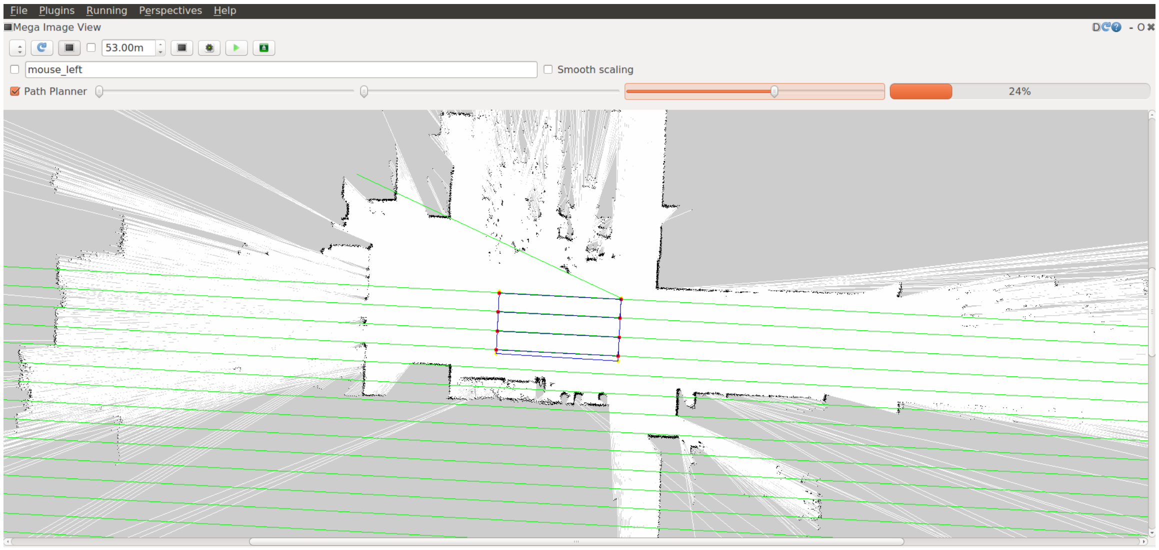


Figure 3-6: Created coverage path with horizontalmachine direction and large grind overlap

counter-clockwise rotation. Therefore, each coordinate needs to be remapped and converted from pixels to metres. When the position is sent to the robot, the robot knows both the desired global position and its current position, so it can accurately plan a path from its current position to the goal. The orientation is determined such that the robot achieves its goal in x, y and then rotates in position to face the next goal.

3.3 Hardware Selection

3.3.1 Localisation Sensor

A sensor must be used to help aid localisation to overcome the inherent accumulation of errors from wheel odometry. A number of different sensor technologies can be used, each offering different advantages and disadvantages. A selection of these technologies has been described in the literature review.

Due to its affordability yet relatively good accuracy, range and resolution, a SICK LMS291 was selected for localisation of the robot. This scanner provides a 2D laser scan of the environment, and can produce 50 mm accuracy up to 80 m or 35 mm accuracy up to 8 m [3]. The laser scanner can easily be integrated into the ROS framework with an existing ROS package. The sensor data can be used by Gmapping [35, 36] to create a 2D map of the environment or by AMCL [71] to localise the robot in an already created map. AMCL uses a probabilistic approach to match the laser scans to likely positions in the map.

3.3.2 Floor Sensor

A sensor is required to capture the floor surface profile information. A number of technologies can be used for this task. Selected technologies are summarised in Table 3.1.

A 2D laser scanner was selected for initial floor scanning, this is once again due to both accessibility and affordability. While 3D laser scanners have been used to perform accurate sensing of an environment, including the floor, they are very expensive and thus make them not feasible for some applications. The 2D laser scanner used for initial testing was a SICK LMS291 laser scanner, which is relatively cheap at around US$6000. The SICK LMS291 has an aperture range of 180-degrees, with an angular resolution of 0.25-degrees. At a range of up to 80 m the accuracy is ± 50 mm, reducing to ± 35 mm at a range of up to 8 m [3]. Additionally, a Hokuyo URG 2D laser scanner was used for further testing, due to its short-range design. The Hokuyo laser scanner has a detectable range of 20 mm to 5600 mm, with a field of view of 240-degrees at a resoltuion of 0.36-degrees [63]. However, despite having been designed for short-range use, the accuracy of this laser scanner is only ± 30 mm. The cost of the Hokuyo laser scanner is around US$1080, substantially less than the other sensors.

An RGB-D camera was selected as a secondary sensor for testing and comparison. The RGB-D camera used was a D435 Intel RealSense camera, which uses active IR stereo to produce a depth image alongside the RGB data from a 2 MP camera. [45].

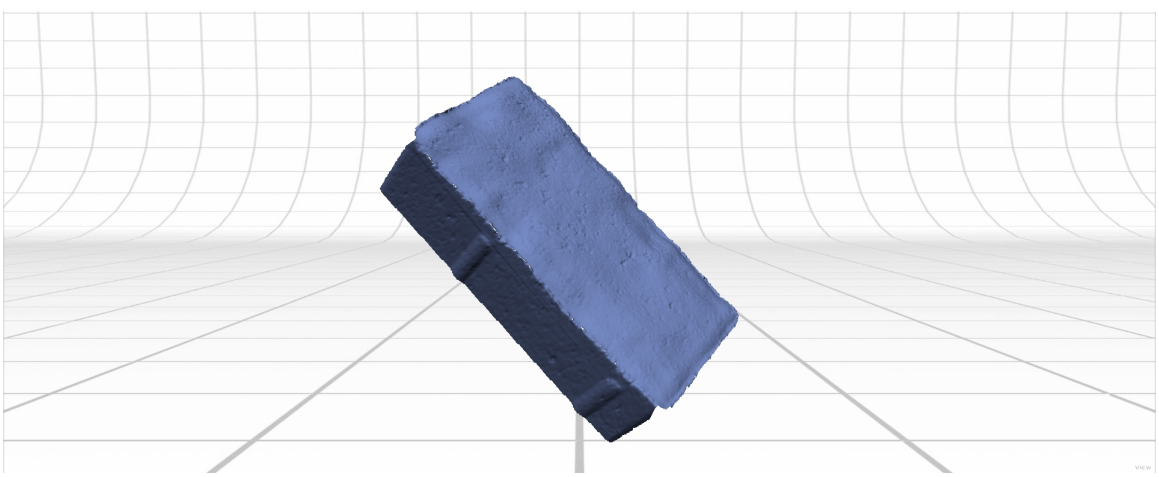


Figure 3-7: NextEngine scan of brick [31]

Table 3.1: Sensor Specifications

Sensor	Range	Accuracy	Resolution	Price
SICK LMS291	8m or up to 80m	± 35 mm and 50mm	0.25 degrees	US$6000
Hokuyo URG	20mm to 5600mm	± 30 mm	0.36 degrees	US$1080
Intel D435	10m	not stated	640 x 480 pixels	US$180
NextEngine 3D	200mm	± 0.30 mm	3.50	US$2995

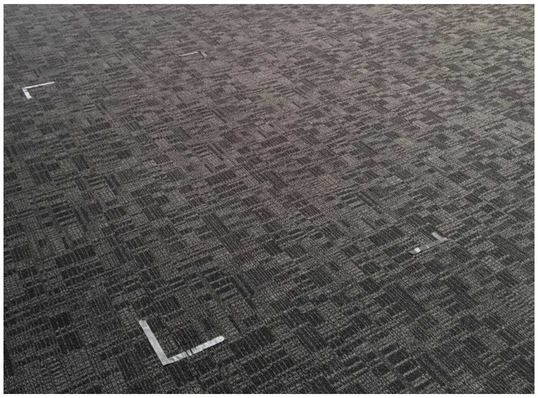
Optical Interferometry can also provide detailed scans of a surface; however, often have long scan times or requires a textured surface for good performance. For example, a multi-laser-based scanner, the NextEngine 360 [31], performs very well with masonry (Figure 3-7). However, the sensor can take up to 2 minutes to perform a scan, rendering it unsuitable for this application. This sensor provides incredible accuracy, with up to ± 100 microns for a macro model and up to ± 300 micron for models with a wider field of view.

3.4 Initial Testing

The platform's ability to capture a floor surface profile was initially tested using two 2D laser scanners. One laser scanner (mounted vertically) was used to capture the floor profile, while the other laser scanner (mounted horizontally) was used to localise the robot within the environment. The initial testing methodology and results were presented in the Conference Paper 'Floor Surface Scanning using a Mobile Robot and Laser Scanner' (Appendix A) [77].

3.4.1 Experiment Methodology

The scanning experiments were set in an area of 2 x 2 m^2 marked out with black electrical tape (Figures 3-8a, 3-8b, and 3-8c). This tape has low reflectivity and high absorbency, resulting in a poor laser scan measurement that helps to identify the boundaries of the scanned area in the final assembled point cloud. The robot was positioned outside the lower left-hand corner of the square and then followed a coverage path (Figure 3-9). This path provided sufficient space for the robot to perform a turn and record scans of the surface. The robot moved at a relatively slow velocity of 0.1 ms^{-1}. At the beginning of each test, all laser scans were recorded in a ROS bag file for later analysis if required, and the real-time laser



(a) Carpeted floor

(b) Asphalt floor

(c) Coated asphalt floor

Figure 3-8: Test surfaces used for mapping

scan to point cloud conversion begun. This point cloud creation process involved capturing every laser scan and associated transform and placing them in a 3D coordinate system. The assembly of laser scans was then converted to a single point cloud of the floor, which was then saved as a .pcd file for analysis. In each test it took around six minutes to complete the coverage path.

The robotic platform was used to map three different surfaces: carpet flooring (Figure 3-8a), outdoor asphalt pavement (Figure 3-8b), and a coated asphalt floor (workshop floor) (Figure 3-8c). These surfaces were chosen to provide a representative sample of various types of indoor flooring. The test surfaces were expected to give insight into how well the laser scanning could identify areas of interest for the different surfaces. Each surface was mapped three times and cross-analysed to determine accuracy. A contour plot from the resulting point cloud was created and used to identify high and low areas of the floor.

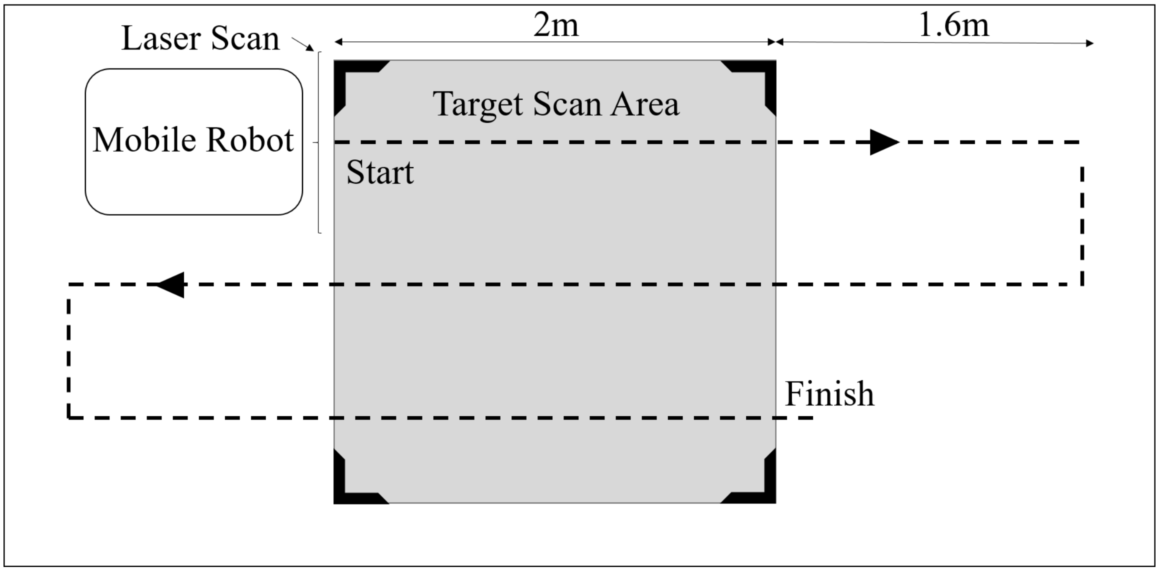


Figure 3-9: Coverage path for scan area

3.4.2 Measurement Methods

The captured point cloud of each floor surface was saved as a .pcd file. MATLAB was used for processing, which involved clipping the scanned area to the target size of 2 m x 2 m. The 'black tape' outliers were removed by applying a threshold to the point cloud data set, and a Gaussian 5 x 5 filter was then applied to the data to smooth the resulting surface and reduce noise. The point cloud was then presented as a contour plot, indicating high and low areas throughout the 2 m x 2 m area. The surfaces were inspected by touch and visually for any deviations in flatness at key areas. These areas were noted and compared to the resulting point cloud and contour plot.

3.4.3 Initial Floor Capture Results

The mobile robot system was able to successfully locate itself and use this information to create a surface profile of each floor. The odometry information provided sufficient pose and orientation estimation to capture the general surface profile of each floor. Odometry errors were observed consistently in all tests. Additionally, systematic errors from the laser scanner were observed in all tests, illustrated by a continuous low measurement near the centre of the scan.

3.4.4 Carpeted Floor

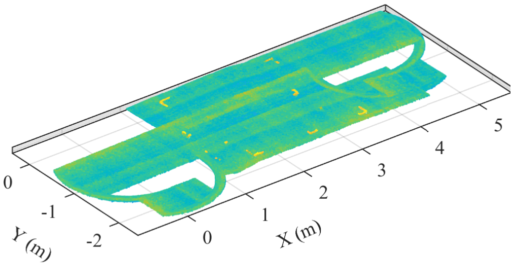
The carpeted floor was successfully mapped despite being anticipated to be a difficult surface for consistent performance due to the fibre orientations of the carpet. The laser scan provided a reasonably thick surface measurement of 0.1 m. The carpet was difficult to inspect visually and appeared to be relatively flat. The contour plot (Figure 3-10c) shows a relatively flat area (with the systematic centre scan error) and a slight high area towards the bottom of the target area.

3.4.5 Workshop Floor

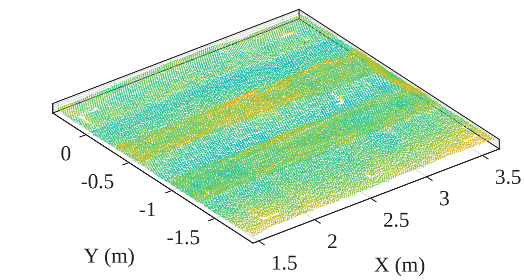
The workshop floor is an example of an indoor surface covered with dust, cracks and pits. This type of surface could be hard to map; however, from the results it is clear that the scanning system could successfully create a consistent point cloud of the coated asphalt (workshop) floor. Despite the consistent low centre measurement due to a systematic error, a high area was identified by the surface scanning system at the middle right of the target area (Figure 3-11c). This area was confirmed by visual inspection as a large step change in the floor. There are two-coin sized dents (30 mm diameter) in the floor of the workshop that the mapping system was unable to detect. However, the system did detect a general slope along the y axis.

3.4.6 Asphalt

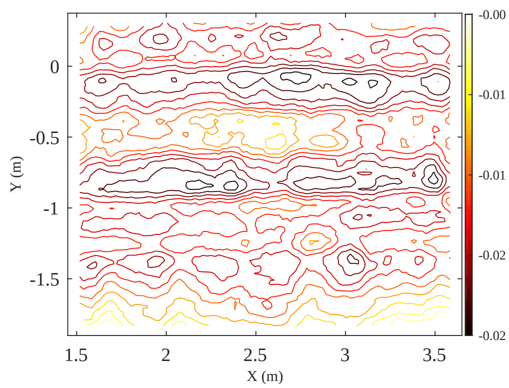
Asphalt is another example of a difficult surface to map, as the colour and texture may vary due to weathering and wear and tear. This surface was also successfully mapped, demonstrating the strength of the developed system. Even though the entire surface was on a gradual slope, and no IMU data was available, the surface profile suggested a high point to the right and a low point to the left of the start position. A ridge in the surface was detected similarly to the coated asphalt floor, but upon inspection this high region was due to a rougher area of asphalt. The contour plot (Figure 3-12c) illustrates the general slope of the surface, with some deviations of the slope due to surface roughness and the systematic errors.



(a) Carpet floor raw data



(b) Clipped carpet floor scan showing target area

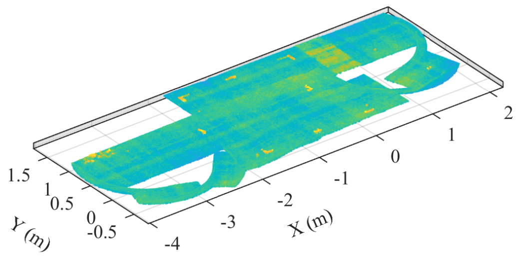


(c) Contour plot of carpet floor

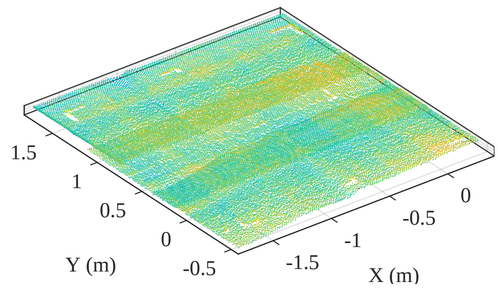
Figure 3-10: Results for carpeted surface

3.5 Initial Challenges

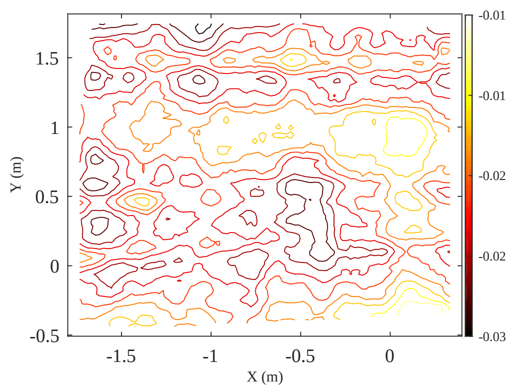
Although initial testing did prove to be successful, there are a number of improvements that can be made and challenges that can be overcome. The initial development and testing identified challenges that include: localisation, sensor accuracy, and 2D limitations. These challenges are discussed in the following sections.



(a) Coated asphalt floor raw data



(b) Clipped coated asphalt floor scan showing target area

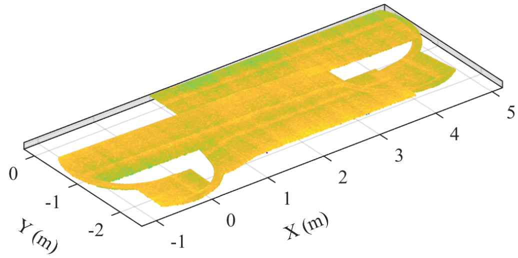


(c) Contour plot of coated asphalt floor

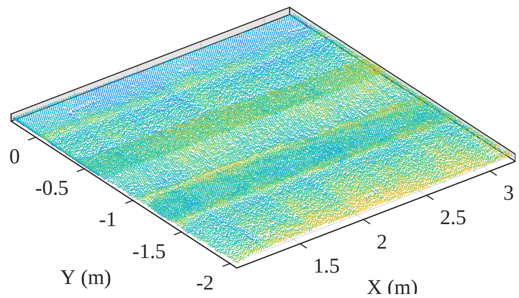
Figure 3-11: Results for coated asphalt surface

3.5.1 Localisation

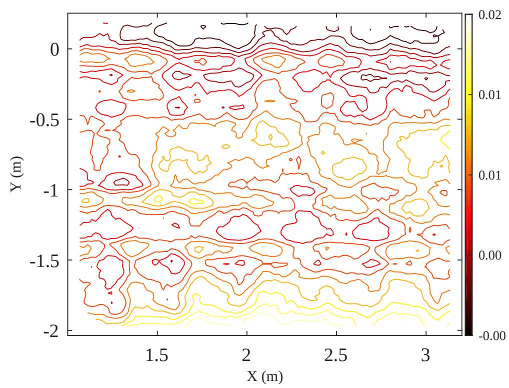
A particular challenge for the robot platform was accurate localisation. Based on research conducted by Thrun et al. [71], the robot can use the horizontal laser scan for localisation through the Adaptive Monte Carlo Localisation (AMCL) ROS node. This node provides a laser scan matching and probabilistic approach for localising the robot from the 2D laser scans. The probability of the robot position in a number of locations is calculated from the combined laser scan matching and wheel odometry information. The position with the highest probability is updated as the robot's current position in the map. This method



(a) Asphalt floor raw data



(b) Clipped asphalt floor scan showing target area



(c) Contour plot of asphalt floor

Figure 3-12: Results for asphalt surface

works well, although due to odometry errors and drift, the robot will jump to the calculated 'correct' position every time the AMCL node updates. These jumps are small and manageable in some applications, but for this particular application they are not desirable as this will result in a shift in the floor profile. The transformed laser scans will have gaps when the position jump occurs, and this could result in inaccurate floor profile estimation.

This challenge can be overcome through a couple of different techniques. First, the jumping action can be minimised by calibrating the odometry to minimise errors and therefore minimise the possible jump in position. In practice this can be difficult due to a number of hard-to-control dynamic factors that can contribute to odometry errors, such as uneven floor, varying tyre pressure over time, axle alignment, wheel point of contact, and tyre slip. However, a similar process to that of Borenstein et al. [13] can be used to estimate the correct wheel radius and wheel separation parameters for odometry tuning. In addition, this can help identify any alignment issues with the robot that can then be allowed for through the wheel odometry calculation. An alternative solution for overcoming the jumping is to gather the transforms as the robot moves, and once AMCL updates the robot's position, realign the previous positions to fit the known positions of the robot. This could be computationally expensive but could provide a consistent method of scanning the floor.

3.5.2 2D limitations

The ROS system utilised is largely built around the assumption of a flat 2D surface; in particular, the robot is set up to have the base_link attached to the 2D planar floor, and the laser scanners can only capture data in 2D. However, the surface the robot moves along is 3D, and involves 6 dimensions of robot pose and orientation (x, y, z, roll, pitch, yaw). This means that these 2D assumptions can result in inaccurate readings of the floor profile. As seen from the results, the system is able to identify high and low areas of a floor however, the heights and depths of these high and low areas cannot be accurately quantified. This needs to be overcome for accurate analysis of floor profiles. A number of 3D solutions to this problem exist, but these are often computationally expensive. In contrast, a computationally inexpensive solution similar to that implementated by Wen et al. [76] could be utilised. In this approach, a 2D sensor provides the location of the robot in the 2D plane, and this location is then extended into 3D using additional sensors such as an IMU.

3.6 Further Development

A number of improvements were identified from the initial floor mapping process and experiment results. The following system and process changes aim to achieve these improvements.

3.6.1 Sensor Accuracy

The laser scanner suffered from systematic sensor errors which could be attributed to its longer-range design. Therefore, a Hokuyo short-range sensor and an RGB-D camera will be tested with an aim to overcome some of these accuracy and systematic error challenges.

Comparison experiments are to be performed using these additional sensors. The short-range laser scanner requires correction to convert the raw laser scan data from polar coordinates to Cartesian coordinates. The system methodology will have to be adjusted for the larger field of view of the RGB-D camera and will need to accommodate a sensor calibration step.

3.6.2 Map Creation

Other software approaches for 3D mapping are to be trialled, in particular Google Cartographer 3D and 3D Octomapping. These approaches could help to create accurate point clouds from sensor information that can then be manipulated and applied to the 2.5D grinding simulation and other terrain navigation applications. This will be particularly important for managing the large point clouds created by the RGB-D RealSense camera, as the raw point cloud data will quickly become inefficient to manage.

3.6.3 2D to 3D Extrapolation

The current implementation utilises 2D approaches for capturing a 3D floor profile. Whilst the methodology gives insight into the general flow of the floor, it lacks the accuracy to be useful in application. This accuracy can be improved by extending the 2D approaches into an adapted 3D system. AMCL localisation can provide accurate positioning through the use of a 2D laser scanner in a 2D plane, however, this localisation does not consider changes in z height and the orientation of the robot in terms of roll and pitch. These additional considerations are necessary for accurate mapping of the 3D environment.

Such mapping has typically been achieved through the use of expensive 3D sensors such as 3D laser scanners, but this is not feasible for this application. In addition, an IMU is often utilised to provide 6 DOF information on robot pose and orientation [27, 81, 22]. The target application surface can be considered as a relatively flat floor with simple-shaped obstacles (flat walls), and so the use of a 2D laser scanner for information on x and y position is a relatively cheap and effective solution. It is therefore proposed that the current 2D system be extended into 3D. This will be implemented by utilising an IMU for inertial information, AMCL for x and y position information, and estimated changes in z based on filtered IMU data and odometry information (Equation 3.1, below).

A node was created to fuse IMU and odometry information into an updated odometry frame. This was published as the Odom to base_link transform, which was then used by

AMCL for localisation. AMCL does not take the z, roll, or pitch components of the frames into consideration, and can therefore successfully update the x, y, and yaw position and orientation information, whilst the roll, pitch, and global z height can be adjusted. This introduces modularity and can provide the ability to create other methods for calculating the z and 6 DOF information, such as visual odometry and point cloud registration.

$$
\begin{aligned}
\delta_{pos} &= \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \\
r &= \sqrt{(x_0)^2 + (z_0)^2} \\
\Delta z &= \tan(pitch) * \delta_{pos} \\
z &= r * \cos(pitch + \pi/2 + \theta_0) + \Delta z \\
x &= r * \sin(pitch + \pi/2 + \theta_0)
\end{aligned}
\tag{3.1}
$$

An extreme situation was tested to verify this approach, involving capture of the floor profile of a slope in the form of a 10-degree ramp down from a relatively flat floor (Figure 3-14a). Limitations of the previous system resulted in the floor profile being incorrectly captured, particularly due to the measurement performed being the distance from the sensor to the floor. As seen in Figure 3-13, when the robot is on a constant slope the measured distance to the ground is the same as when measuring on a flat floor. This, combined with 2D localisation assumptions and no consideration of changes in global z height, results in the system incorrectly capturing a flat floor. To overcome this, the pitch of the robot and the global z height must be considered. To capture the floor, the robot drove forwards at a speed of 0.1 ms^{-1} and recorded the resulting point cloud information. The captured point cloud using the improved capture system is shown in Figure 3-14c; for comparison, Figure 3-14b shows the point cloud captured with no z compensation or consideration of slope. There is significant improvement in the floor capture capability, with the slope of the ramp continuing to be captured even when the robot is fully on the ramp. This suggests that the z height compensation helps to extrapolate the 2D system into a full 3D capture.

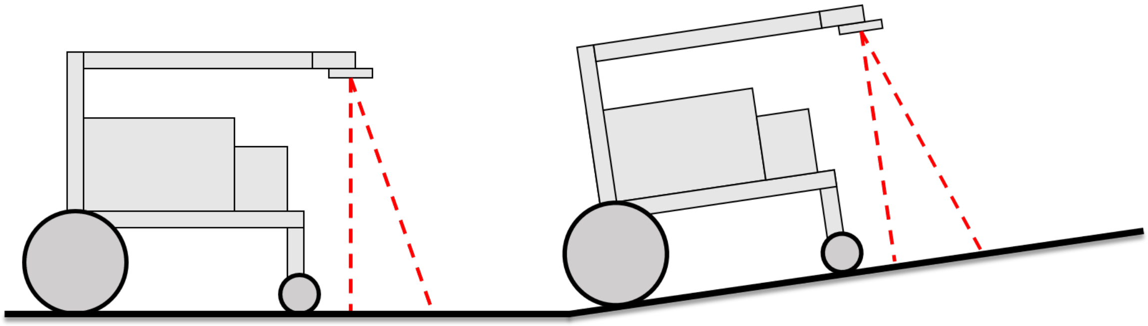
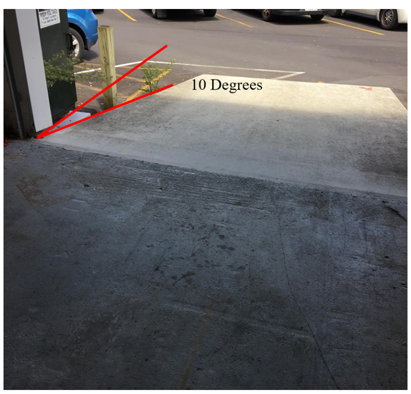
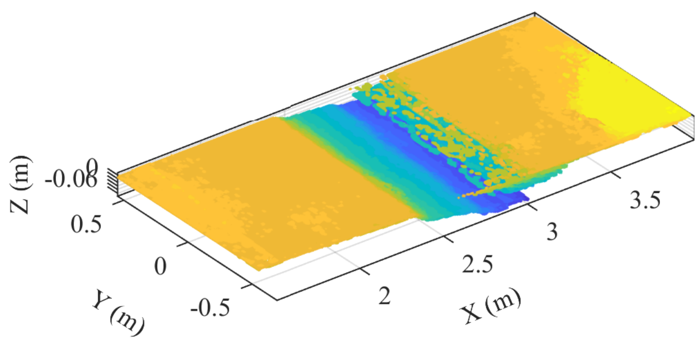


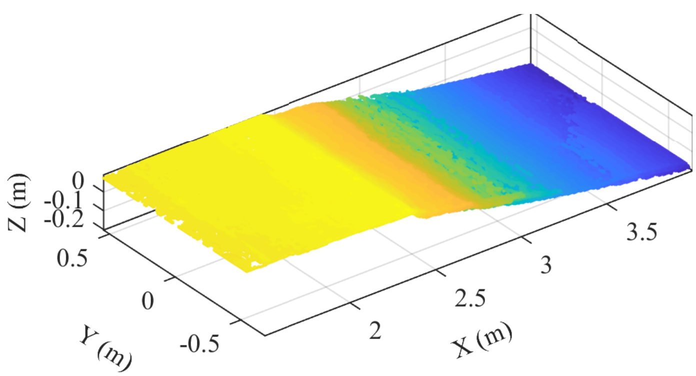
Figure 3-13: Diagram of robot floor measurement on a slope



(a) Ramp down from flat floor



(b) Point cloud captured using initial system



(c) Point cloud captured using improved system

Figure 3-14: 3D extrapolation verification test

3.6.4 RealSense Camera Testing

The Intel RealSense D435 camera is an RGB-D camera and thus provides both a colour image and a depth image of the environment. This information can be used to create a 3D map/model, which can be used to extract surface floor profile information. The camera can be used in two ways: it can be mounted to view the environment from a horizontal position and the floor profile can be extracted through processing, or it can be mounted to view only the ground, potentially increasing accuracy and reducing occlusion errors. Occlusion errors are common when objects are hidden due to the angle of the sensor scan line. The beam sent out by a sensor cannot bend around a corner, so the view is limited to the unobstructed line of sight. In addition, RGB-D cameras can be sensitive to lighting and so must be calibrated.

3.6.5 Floor Profile Creation with RGB-D Camera

The floor profile creation program used for the laser scanner cannot be applied to the RGB-D camera information due to the different methods of storing information. The RGB-D camera provides a point cloud of the depth cloud captured. The point cloud resolution is 640 x 480 points and therefore contains 307,200 points. Thus, to store every point and assemble them correctly requires expensive computation and is inefficient. Further, the camera provides a new point cloud approximately every 0.1 seconds, resulting in around 3,000,000 points per second to process. A common solution to this problem is to use the Octomap Octotree point cloud storage system [42]. In this system, raw point cloud data are down-sampled into 3D voxel grids that are then stored in a tree structure, providing an efficient method for accessing and processing the information. The Octomap attempts to match the new point cloud to previous scans based on the 3D voxel grid. This creates a 3D map stored in an efficient graph-based tree called an Octotree. This has been used by other researchers to create full 3D maps [30, 42], but in this application will be used to produce only a 3D profile of the floor surface.

Due to computational limitations, the real-time processing of the point cloud stream into an Octomap is restricted to voxel grid sizes of around 8 mm. This eliminates some information about the floor profile, which needs to be considered for real-time applications. For testing, the point cloud data is recorded into a bag file and can be processed at a later time. The processing of the recorded point cloud data results in less computational limitations and the resulting achievable voxel grid size can be as small as 2 mm; Octomap point clouds based on this voxel grid size were used during testing. The raw point cloud bag file from a 6 minute test can be as large as 30-40 GB; for comparison, a saved point cloud from an Octomap with a 2 mm voxel grid size is only 20-30 MB, highlighting the substantial reduction in the amount of data to process.

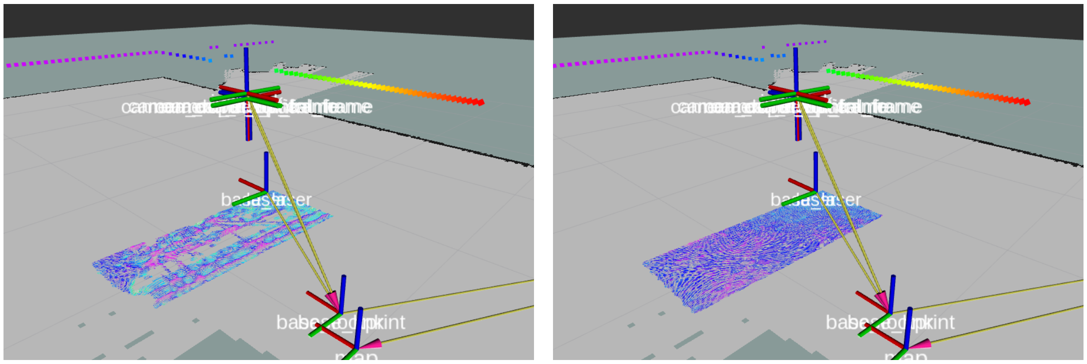
3.7 Improved Experiment Methodology

The experiment methodology for the test comparing the RGB-D camera and the Hokuyo short-range laser scanner is similar to that of the initial tests, described in Section 3.4. In addition to the process outlined in Section 3.4, a calibration step was included prior to testing to improve floor sensor accuracy and to calibrate the odometry scalar variables. Black tape was used to outline the 2 m x 2 m target area due to its different level of reflection giving rise to laser scanner measurements that successfully highlight the edges of the zone. Although the tape does not show up well on the RGB-D point cloud, the two point clouds and target areas can be accurately compared due to localisation in the global coordinate system.

3.7.1 Sensor Calibration

In addition to calibration of the odometry and AMCL, the floor sensor itself must be calibrated prior to testing. The RealSense camera is highly susceptible to infrared light, particularly differences in lighting conditions. This can introduce errors or even result in no measurement. Figures 3-15a and 3-15b highlight the difference between no auto-exposure and an auto-exposed (calibrated) depth cloud. The performance of the RGB-D camera is greatly improved with auto-exposure, capturing a greater region of the view and containing fewer artefacts (missed measurement areas). Errors due to changes in lighting conditions can result in a patterned floor result or even parts of the floor unmeasured (Figure 3-15a). These errors are overcome through calibrating the RealSense camera using the auto-exposure function for 2-5 seconds.

The Hokuyo short range laser scanner produces a laser scan result at angles spanning 270-degrees. The raw laser scan data must be converted from polar coordinates to Cartesian coordinates to accurately capture the floor profile. If this conversion is not performed, the laser scan has an obvious skew, which can be visualised when looking at a flat floor (Figure



(a) Screenshot of uncalibrated RealSense camera (b) Screenshot of calibrated RealSense camera

Figure 3-15: RealSense lighting calibration

3-16). The uncorrected laser scan of the flat floor has a bent shape (bottom curve of Figure 3-16), due in particular to the mapping of coordinates and changes in angle of incidence. This can be corrected (Figure 3-16) through remapping the laser scan data points to the correct position relative to the angle of the scan (Equation 3.2). In Figure 3-16, the z height is indicated by colour, with purple corresponding to the highest z value and red to the lowest. This results in a significantly improved measurement of the floor, although the scan still contains noise and measurement errors.

$$scan_{filtered}.ranges[i] = scan_{filtered}.ranges[i] + -0.03 * exp(-fabs(\sin(angle))) \tag{3.2}$$

3.7.2 Measurement Methods

The final surface point clouds were saved as a .pcd file, which was then imported into MATLAB for analysis. The point clouds were converted into a mesh grid and then smoothed using a Gaussian filter. This helps to reduce noise from erroneous measurements and provides better insight into the surface trends. The resulting point clouds of the surface were compared visually through the aid of a contour plot. The resulting surfaces were checked for deviations and any significant deviations in the floor were inspected manually by visual inspection, touch, and the straight edge approach. This helps identify if the system is correctly detecting high and low areas and not just capturing noise from the sensors. Results from the comparison testing can be found in Chapter 5; Experiments and Results.

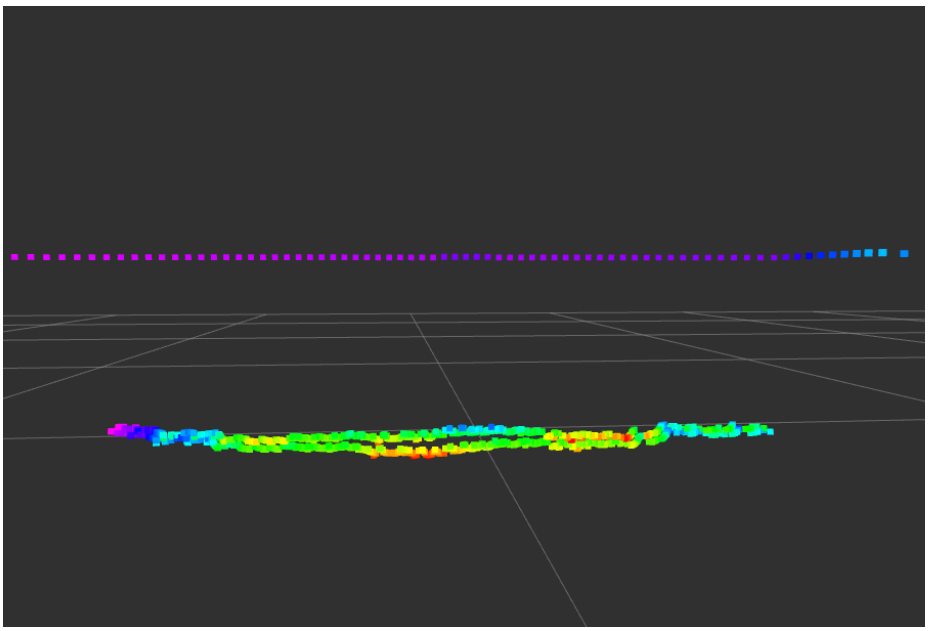


Figure 3-16: Corrected laser scan (above) shown with raw laser scan (below)

Chapter 4

Grinding Model from Floor Profile

4.1 Chapter Overview

After capturing the floor surface profile, the next phase of the project is to represent the floor surface as a mathematical model and understand how to implement the cutting action. A basic grinding model can help to provide understanding of how a grinder performs on an uneven surface, and this understanding can lead to design and control improvements. This chapter details the development of a basic grinding model and discusses a number of assumptions that are required to implement the model.

4.2 Important Aspects of Grinding

For a mathematical model to be accurate with respect to the real world, all of the significant factors in the problem need to be considered and approximated. For grinding, a number of different factors need to be considered, and many of which cannot yet be quantitatively determined. The purpose of the grinding model is to help identify the depth of the cut on the floor and where the cut is performed. In addition, how the cut varies across the floor and along the grinding head profile needs to be quantified for an accurate model.

Important factors are:

- Planetary head rotational velocity (back electromotive force (EMF), PID control)
- Forward speed of machine
- Grind head floor pressure

- Grind depth
- Force from floor to grinder and resulting motion
- Tool cutting profile
- Diamond tool wear (Comet Tails) and glazing
- Wet vs dry floor
- Torque of drive grinding motor
- Hardness of aggregate and concrete
- Aggregate location
- Diamond distribution in tool

As discussed in the literature review, common approaches for modelling the grinding action, particularly for wheel grinding applications, involve the estimation of the contact length, chip length, and chip thickness. Due to the horizontal grinding action used in the floor grinding process, these variables are difficult to quantify due to the following reasons;

- Contact length is dependent on diamond tool wear, tool 3D position and shape, floor surface profile, and the action of the grinding head. Accordingly, it is very difficult to accurately determine the contact length of a diamond tool tip with the floor at any point in time.

- Due to the particle nature of concrete, the 'chip' formed when grinding is in the form of dust. The dust is difficult to quantify, and so the chip thickness variable is difficult to quantify. This removal mechanism is the reason why traditional grinding models [52, 69, 80] cannot be easily applied in this situation. In addition, the abrasive nature of the concrete dust aids the grinding process, introducing another stochastic variable into the grind depth equation.

Therefore, an alternative approach to traditional grinding models is required. For this new grinding model, the purpose is to estimate material removal based on grinding head speed, machine speed, and floor profile. A number of the unknown factors, such as contact length, stochastic variables, and 3D cut depth, can thus be considered within a cut profile variable determined through experimentation.

4.3 Model Development

4.3.1 Concrete Grinding Process

The concrete floor grinding process involves the systematic removal of material at the micro scale to create a change in the floor at the macro scale. A typical grinding head is modelled from the Husqvarna 820 Concrete Grinding Machine [43]. The grinding head has a maximum cutting width of 820 mm and is formed by 3 planetary discs (Figure 4-1). The grinding head rotates at a maximum of 60 rpm, and the planetary discs rotate at up to 1100 rpm. This results in a donut shaped coverage area (Figure 4-1). Each disc can hold 3-8 diamond tools, which perform the cutting action. As the machine moves along the floor, the pressure from the weight of the grinding head, motors, and machine engages the diamond cutting tools with the floor and results in material removal in the form of concrete dust. The dust is extracted using a vacuum system.

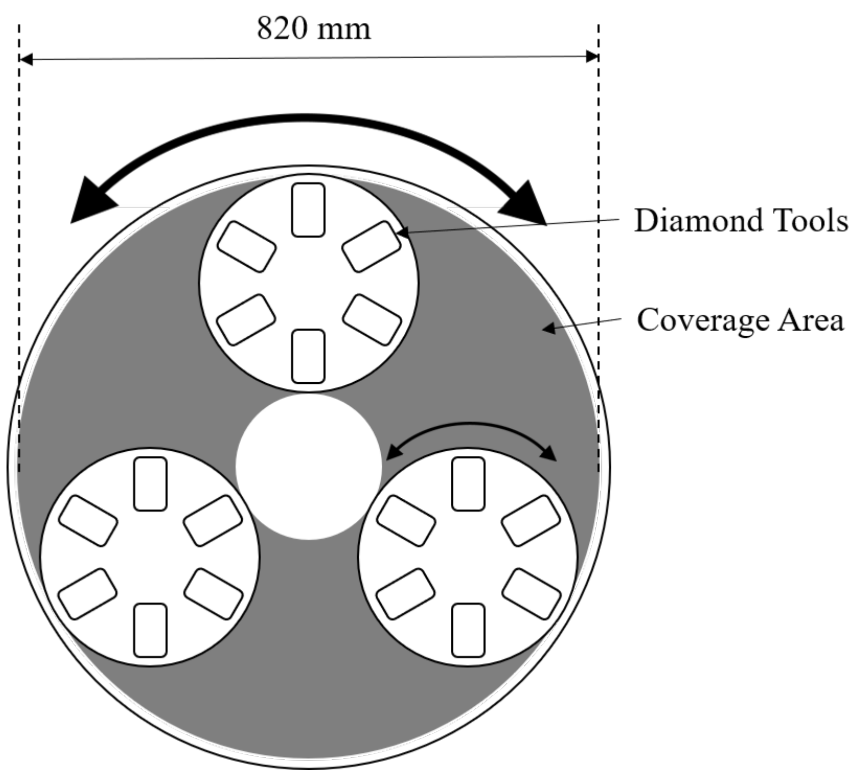


Figure 4-1: Diagram of grinding head

A number of additional factors contribute to the rate of floor material removal by the diamond tools. The concrete dust generated is abrasive in nature and so aids the grinding process. The hardness of the floor material changes due to both the natural properties of concrete and the process in which the floor was laid. Floor hardness determines the type of diamond cutting tool used; typically hard, medium, or soft. The type of diamond cutting tool is determined by the hardness of the metal matrix bonding the diamonds together. This matrix wears with material removal, and correct matrix selection can greatly improve productivity and reduce excessive tool wear. In addition, the diamonds can undergo a process that is commonly called 'comet tailing', which results in a reduced cutting edge and often requires operator intervention to provide efficient grinding once again. Balancing of matrix hardness with concrete hardness is required to ensure that the matrix does not wear too quickly or too slowly. Wearing too fast can result in the diamonds not engaging fully and in tool wastage. Wearing tool slow can underexpose the diamonds, resulting in poor cutting performance and reducing the exposed cut edge; this has a significant impact on the productivity of the grinding process. These factors are difficult to capture in a basic model, and so must be considered in alternative ways.

4.3.2 Initial Process

The initial approach was to adapt traditional grinding models to the floor grinding application. This involved estimating the depth of cut of the tool with respect to time and thus calculating the amount of material removed. However, a number of considerations had to be taken into account. The grinding head of the machine is not a controlled system: it floats above the surface and reacts to the forces of grinding. This differs from traditional wheel grinding, where the grinding head and tool are fixed in a known and controlled position.

Therefore, the first attempt to model the cutting action of the grinding head followed a similar approach to that used by Setti et al. [66]. First, the tool position and kinematic path with respect to time was calculated. If the position of the diamond cutting edges in the tool are known, then it follows that the floor area currently being ground is also known. As the floor profile can be determined through the capture method described previously, the interaction of the kinematic path of the diamonds with the varying floor heights could give insight into the location and rate of material removal. The diamond edges are stochastically distributed on the tool face and so the kinematic path of the cutting edges can only be estimated and assumed to be correct using predictive models [66, 20]. Knowledge of the tool position can therefore provide insight into diamond distribution and position relative to the global frame.

A Mathworks Simulink model (Figure 4-2) was created to simulate tool position with respect to time. The model takes the grinding head rotational velocity, disc head rotational velocity, robot velocity, and a clock of 0.01 time steps as input parameters and outputs the tool position in x and y coordinates. Tool position with respect to time is found using Equation 4.1. In this model, the workpiece surface was modelled in a 2.5D array where the value of the grid indicates z height. The initial model simply removes 0.05 units of height at the tool position at each time step, giving insight into how the tool moves with respect to time.

$$
\begin{aligned}
\theta_g &= \theta_{gi} + \omega_g * t \\
\theta_d &= \theta_{di} + \omega_d * t \\
pos_x &= m_x + r_h * \cos\theta_g + r_w * \cos\theta_d \\
pos_y &= m_y + r_h * \sin\theta_g + r_w * \sin\theta_d
\end{aligned}
\tag{4.1}
$$

Where m_x and m_y are machine position in x and y (metres), θ_g and θ_d are the grinding head and grinding disc rotational positions, respectively, and r_h and r_w are the radius of the grinding head and the grinding disc, respectively. θ_g and θ_d are determined by the initial positions (θ_{gi} and θ_{di}) multiplied by ω_g and ω_d, the respective rotational velocities of the grinding head and grinding disc in radians per second.

The velocity of the grinding tool and diamond edges can give insight into the forces involved and thus the rate of material removal. Depending on the direction of the grinding head relative to the grinding discs, the rotational velocities, ω_g and ω_d, can combine constructively or destructively. The velocity of the tooth can be given by Equation 4.2.

$$
\begin{aligned}
v_{dx} &= r_w * \cos\theta_d * \omega_d \\
v_{dy} &= r_w * \sin\theta_d * \omega_d \\
v_{gx} &= r_h * \cos\theta_g * \omega_d \\
v_{gy} &= r_h * \sin\theta_g * \omega_d
\end{aligned}
\tag{4.2}
$$

Where v_{dx} and v_{dy} are the velocities of the grinding disc in x and y, and v_{gx} and v_{gy} are the velocities of the grinding head in x and y. These two velocities either add to or subtract from each other depending on the direction of rotation. The direction of rotation is taken as positive if clockwise, and negative if counter-clockwise. The velocity of a tooth on the grinding disc can thus be determined by Equation 4.3.

$$
\begin{aligned}
v_{tx} &= direction * (v_{dx} + v_{gx} + m_x) \\
v_{ty} &= direction * (v_{dy} + v_{gy} + m_y)
\end{aligned}
\tag{4.3}
$$

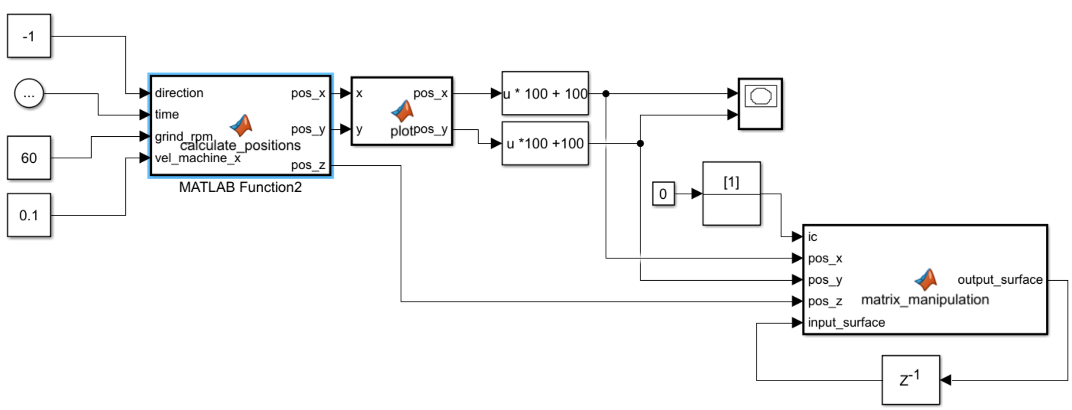


Figure 4-2: Simulink model of tool positions with respect to time

4.3.3 Initial Model Testing and Results

Results for the initial model show similar tool path patterns to those seen on the floor during the floor grinding process. An image of the scratch pattern of a floor grinding process is shown in Figure 4-3a. A comparison of the calculated tool position in a 5-second simulation

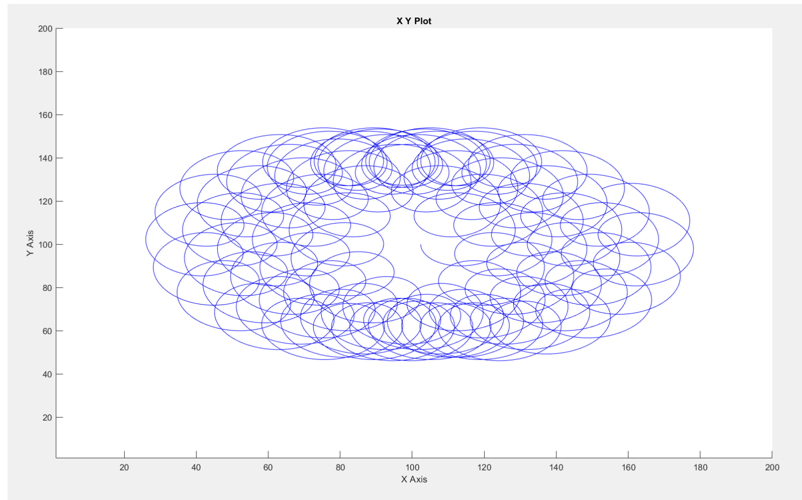
Table 4.1: Time to Complete Simulation

Test	1	2	3	4	5	Average
Single grit	4.17	3.19	2.88	3.30	3.17	3.34
Multiple Grits	143.13	148.73	142.94	147.14	145.03	145.39
Single-grit Surface	36.78	39.69	37.98	37.56	37.84	37.97
Multiple-Grit Surface	175.89	177.80	169.91	170.22	171.14	172.99

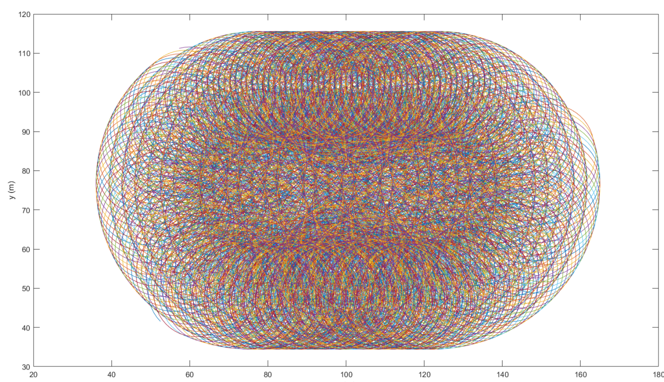
with a time step of 0.001-seconds is shown in Figures 4-3b and 4-3c. Figure 4-3b shows the path of a single point (grit) on a grinding disc, whilst Figure 4-3c shows the path of 6 diamond tool points on three discs (18 paths in total). The simulation parameters used were based on estimated real-world data: 6 diamond tools per disc; 3 discs; and a constant machine speed of 0.1 ms^{-1}. It can be seen that the resulting scratch pattern is similar, and it can therefore be assumed that the initial model is able to estimate the motion of the diamond cutting edges in the 2D x and y plane. Interestingly, the scratch pattern of a grinding machine often looks like a series of large-radius scratches; as can be seen in Figure 4-3c, this large-radius path is in fact generated by the sequential paths of the multiple tools. The simulation took an average of 3.34 seconds to complete over a series of five runs (Table 4.1) under the same computer conditions, with accelerated simulation computation. With additional grits, 18 used in this test (3 discs each with 6 tools), the simulation time increases significantly, to an average of 145.39 seconds. This is probably due to the processing and storing of large vectors, whereas the single-grit analysis only needs to store single values.

The Simulink simulation was run again with a basic 2 m x 2 m surface manipulation block added into the model (Figure 4-4a). This can give insight into the increased computational requirements of surface manipulation. The surface manipulation involves removing 0.05 m of material at each tool position at each time step. The results are tabulated in Table 4.1, and an example output is shown in Figure 4-4b. There is a significant increase in computation time when large arrays, due to either multiple grits or surface processing, are manipulated or fed back into the system. With a single-grit analysis the simulation can run in real-time with small time steps (0.001 seconds), as the average time taken for a 5-second simulation was only 3.34 seconds. However, this is not possible for the simulations that process arrays, where substantial increases in the time required rule out the possibility of real-time processing. This could be a limiting factor in model development.

(a) Scratch pattern produced by grinding machine

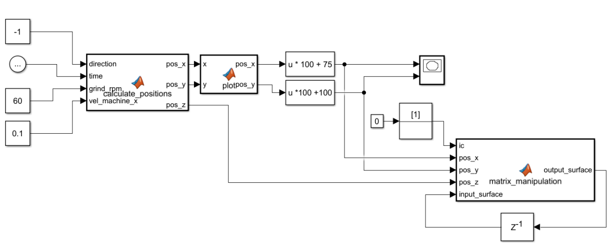


(b) Single grit scratch pattern 5-second simulation

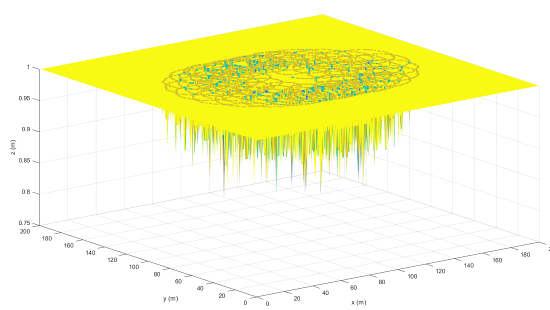


(c) Simulated multiple-grit scratch pattern 5-second simulation

Figure 4-3: Scratch pattern results



(a) Simulink model



(b) Example surface result for single grit

Figure 4-4: Simulink simulation with surface manipulation

4.3.4 Contact Modelling

The results demonstrate that it is possible to calculate the position of the diamond tools in an x and y plane given the initial positions and the rotational velocities of the grinding head and planetary grinding discs. However, for a model to accurately capture interactions with the floor, the contact between the diamond cutting edges and the floor surface must be captured. Furthermore, the 6 Degrees of Freedom (DOF) dynamics of the machine are required for full analysis of the grinding motion relative to the floor surface profile. The machine dynamics and diamond edge contact information will help to extrapolate the model from the x and y plane into the 3D interactions of the machine and the floor surface. Modelling the dynamics of the machine can be achieved through three typical approaches, discussed below.

Impulse

Impulse modelling involves analysing the contact between two objects at a one-time event when they collide. Before collision, each object has a certain momentum (mass * velocity) or rotational equivalent. Depending on whether the collision is elastic or inelastic, the magnitude and direction of each object's momentum will change. These changes in momentum can then give insight into the forces involved and the resulting velocity (position). This approach works well in models where object collisions are infrequent.

Force-Based

A force-based approach involves the understanding of the net forces. Each object has a normal force based on a force law. A penalty force can then be applied, which can allow objects to overlap (helping to estimate deformation of intersection) when they collide. The calculation of force-based collisions can require multiple simulation time steps and are thus computationally more expensive than impulse-based approaches.

Motion Constraint

In a motion-constraint model it is assumed that objects are always in contact, so collision dynamics are not required. This means that computation around the normal forces and resulting interactions are not required, and so simplifies the modelling process. However, many applications cannot assume the objects are always in contact.

4.3.5 Grinding Head Position

An important consideration for tool position estimation in 6 DOF is how the grinding head sits on the floor surface. On a flat floor, the grinding head can be considered to also be flat and grinding of each part of the floor to be relatively consistent. However, on an uneven floor the grinding head will only grind some areas, and may have an orientation change in 6 DOF. This 6 DOF information is required for accurate modelling of tool position.

Solution of this challenging problem has often been achieved through using physics engines resulting in additional computational overhead. Due to the various points of contact, the grinding head can sit on a number of available planes. Simplifying the process to a 2D situation allows it to be modelled such that a line must be touching the highest point, and is then rotated about that point of contact until it contacts the floor again (Figure 4-5). The centre of gravity of the machine must be taken into consideration as some solutions will result in an unstable position. Extrapolating this problem to 3D can be achieved in a similar approach through large amounts of computation. First, the highest point in the known target area can be found. This point must be in contact with the grinding head. From this known point, at least two other points of contact are then estimated and checked for position stability and machine kinematics. This problem becomes exponentially complex when considering the stochastic protrusion of the diamond edges in the tool, as it is these points of contact that will determine the grinding head 6 DOF orientation.

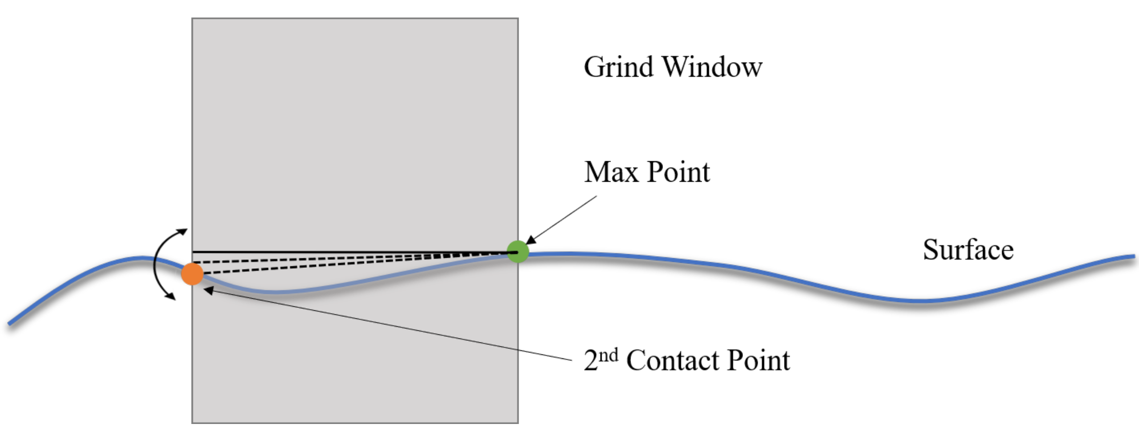


Figure 4-5: 2D Diagram of grinding head surface contact

While this approach is manageable in theory, in actual application it requires a number of checks to ensure that unstable conditions or natural limits are not reached. This results in a large amount of computation, and combined with additional processes results in a computationally expensive process.

4.3.6 Critical Analysis of Initial Methodology

The approach discussed above provides a robust analysis of a number of variables; however, in order to incorporate a number of additional variables such as grinding head dynamics, varying machine speed, tool wear, and tool global position in 6 DOF, the grinding model quickly becomes computationally inefficient. As seen in the preliminary simulation results, the manipulation of large arrays results in exponential increases in computation, and this reduces the practicality of the model. Large arrays fed-back into the system are difficult to avoid because the current cutting conditions depend largely on the current state of the floor, and so the floor must be updated by changing the floor surface array and providing the new array as input to provide an accurate basis for calculations. In addition, the model assumes that every diamond cutting edge is in the cutting phase, whereas in reality, only diamonds with sufficient interaction with the floor surface will be removing significant amounts of material. Some diamonds may be in the rubbing phase, ploughing phase, or not in contact with the surface at all. Many of these variables rely on predictive models of stochastic distribution and therefore, despite the extensive computation, may fail to estimate the correct grind profile. A simplified alternative grinding model will therefore be used for the second approach.

4.3.7 Initial Process Conclusion

Analysis of the initial approach showed that for the model to be effective, it must consider a number of additional variables, and these variables are too computationally inefficient to quantify at this time. In particular, accurate tool position, diamond cutting tool shape, cut depth, and resulting wear would be required to estimate material removal at any given time. In addition, in order for tool position to be known with respect to the floor surface, machine position on the surface in 6 DOF is required. From the machine's 6 DOF, and considering the resultant forces involved, the global tool position can be estimated. This would require implementation of a physics engine and a computational simulation, which was considered out of scope for this project. A simplified model will therefore aim to provide insight into the grinding process by estimating the material removal based on the known floor profile and grinding head position.

4.4 Simplified Global Flatness Model

To achieve global flatness, a control method that utilises the known floor heights and deviations must be devised. For a simplified model, the grinding action was reduced through a number of assumptions and simulation methods. The floor profile was discretized into a 2.5D grid of heights, and these heights were used to determine the amount of material required to be removed to achieve global flatness at each point. The grinding head was represented by a moving window of points, initially defined by a rectangle and then extended to be represented as any shape, in particular a circle and a donut. This representation provides the ability to adjust the grinding head profile according to experimental data and remove the need for estimation of diamond and aggregate position and height. The model was created through MATLAB and can be directly integrated with floor scan processing. This could enable the model to be integrated into ROS for the robot control.

4.4.1 Assumptions

A number of assumptions were made for the initial model, based on an analysis of the grinding action of a machine and following thorough discussion with an industry expert.

- The grinder grinds all parts of the floor equally at a particular moment in time. This is not true in real-world scenarios, as deviations in the floor cause variations in the contact between the grinding head and the floor. These variations in contact cause variations in the amount of material removed, and so, given the grinder position, it is very difficult to quantify the amount of material removed at each point under the grinder. This assumption reduces the amount of computation required and therefore reduces algorithm development time.

- The grinding head remains flat with respect to true ground. In reality, the grinding head can pivot in pitch, and so moves with the floor to some degree. This causes variations in the cutting depth and cutting action. In addition, it can cause issues such as scalloping at the bottom of undulations. Scalloping occurs when the grinding head is shifted in terms of pitch and begins to aggressively remove material from the base of the undulation due to the angle of the grinding head.

- While the model assumes that the area under the grinding head is ground at all times, the grinder only grinds at the diamond tool tips. The position of the tool tips can be determined and so this assumption can be overcome in the future with additional computation.
- The grinding machine will naturally remove high-frequency peaks in the floor, and so the algorithm focuses on low-frequency, long undulations in the floor. This assumption is related to the contact area of the grinding head. Due to the contact dynamics, the grinding head will, to a point, naturally remove small, high-frequency peaks in the floor.
- The cutting action remains relatively constant throughout the process. In reality, the diamonds of the cutting tool constantly wear and replace, and thus the cutting action, depth, and dynamics is constantly changing in a stochastic way. This makes it difficult to determine exactly how much material is being removed at any one moment; however, we can assume an approximation and stochastic models can be created to help verify this assumption. These additional models are out of scope for this research.

4.4.2 Simplified Model Development

The initial grinding model simplifies many aspects of the grinding action. The grinding area is simplified to a rectangle of 82 x 82 units, where each unit is 0.01 m. The surface is simplified to the large undulations in the floor: the focus of the grinding model. This is achieved by reducing the resolution of the floor down to grid sizes of 0.01 m and by smoothing the surface, which removes any surface roughness that will naturally be removed during the grinding process. The surface is represented as an array of z heights, similar to that of a point cloud, and similar to a 2.5D surface representation. This surface representation can be taken from the point cloud created from a floor profile capture process such as that described previously. An example surface created in MATLAB, using Equation 4.4, is shown in Figure 4-6. The deviation along this example surface is 45 mm. The grinding area array, called the grind window, follows the path of the grinding head and moves along the surface 2.5D array. The moving grind window array can therefore provide the current surface of the floor under the grinding head and use this information for control.

$$
\begin{aligned}
a &= base\_height \\
b &= a + \sigma_{floor}^2 \\
rand\_num &= (b-a) * rand(6,6) + a \\
x_t &= linspace(1, length, 6) \\
y_t &= linspace(1, length, 6) \\
[xq1, yq1] &= meshgrid(1:1:length) \\
surface &= griddata(x_t, y_t, rand\_num, xq1, yq1, 'cubic')
\end{aligned}
\tag{4.4}
$$

Where a is the base_height of the created floor (usually 1), and b is the base_height of the floor plus the desired floor variation. The MATLAB functions rand(), linspace(), meshgrid(), and griddata() are used to create a randomly generated floor of any size that can be used for further processing. Floors of 500 units were used in the initial testing.

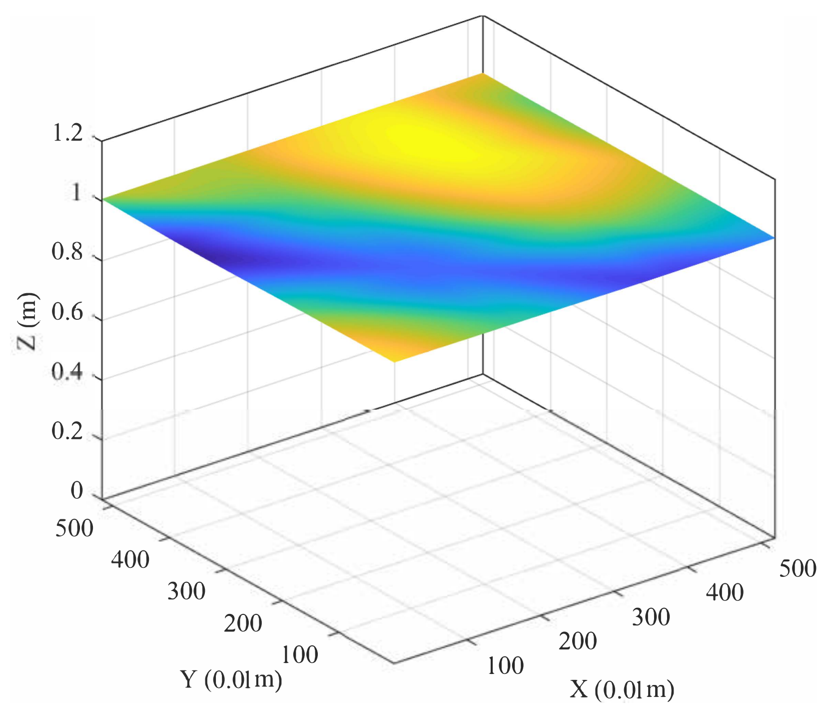


Figure 4-6: Example of created surface to be ground

The position of the robot is stored in a matrix. The next position, $h_{x_{i+1},y_{i+1}}$ is given by the current position h_{x_i,y_i} and the chosen path. The position of the grinding head is simplified to 2D (x and y) and is assumed to sit on the surface of the floor at that point. The orientation of the robot is ignored and only the direction of the grinding head along the path is considered. This removes the need for additional computation to take the dynamics of the robot itself into account. The robot moves between set positions along the coverage path at varying speed, which is pre-determined by the time required to grind the surface down to the target height or by a set value such as a constant machine speed.

To apply the model, the time required at each point in the array to grind to a specified height must first be estimated (Equation 4.5). This height is determined by the current grit of diamond tool being used, and thus indicates the desired level of material removal across the surface.

$$\begin{gathered} target\_depth = min(input) - cut\_depth \\ pos_t = (h_{x,y} - target\_depth)/cut\_rate \end{gathered} \tag{4.5}$$

Where *input* is the input floor surface, stored in a 2.5D matrix; *cut_depth* is an estimated variable that captures the depth of cut across the grinding head in a unit of time; and $h_{x,y}$ is a coordinate value of the 2.5D floor height matrix.

There are some real-world limitations: first, the machine cannot accelerate and decelerate too quickly. In addition, the amount of material removed affects the overall aesthetics of the floor, in particular the amount of exposed aggregate. The maximum amount of material that can be removed is therefore limited to the desired end-aesthetic of the floor. The cutting rate is a variable that has great influence on the amount of material removed, and without experimentation is difficult to accurately determine. From discussions with Jason Torbet, an expert in concrete grinding, the cut rate was assumed to be 3 mm/time step.

The time to grind is then applied to the model by providing an indicator of the speed of the machine required to remove the desired amount of material at the point. However, as the grinding head has a certain area, we cannot focus on a single 0.01 grid cell at a given time. Grinding at one point within the grinding head (grind window) will affect the other points within the grinding area, and this must be considered for accurate control of the machine. This was achieved by averaging the required time across the grind window area.

The method used to calculate the average can be altered, for example to better simulate aggressive grinding. The average time is then used to control the speed of the machine. Based on this speed, the machine is then moved step by step along the surface. At each step, the amount of material removed is determined by the cutting rate divided by the speed of the machine (Equation 4.6).

$$\delta h_{x,y} = cut\_rate/v_x \tag{4.6}$$

Where $\delta h_{x,y}$ gives the material removed and v_x is the velocity of the machine in the forward direction.

In addition, a variety of masks can be applied to alter where and how much material is removed. The addition of masks provides modularity and the ability to capture variables that are difficult to simulate, such as the 3D grind profile of the grinding head. At this step, diamond tool wear could be considered in future improvements by adjusting the cut_rate.

The new height of the grind_window is found by removing the calculated change in material across the grind window (Equation 4.7). At this stage the material removed is consistent across the grind window; however, this methodology provides a means for the grind window shape and profile to be easily adjusted through the use of masks.

$$h_{x_1-x_2,y_1-y_2} = h_{x_1-x_2,y_1-y_2} - \delta h_{x,y} \tag{4.7}$$

Where $h_{x_1-x_2,y_1-y_2}$ specifies a 2.5D sub-matrix of the grind window position relative to the global floor matrix, h. The value of the matrix corresponds to the height of the floor at that position. $x_1 - x_2$ and $y_1 - y_2$ specify the bounds of the grind window, which allows the window to move across the global floor matrix at each time step.

4.4.3 Consideration of End Condition

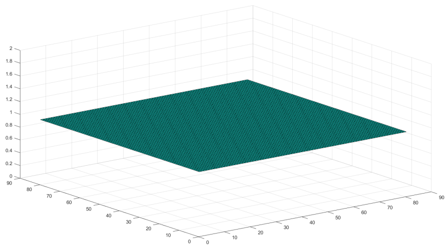
Throughout the grinding process, turning the grinder around to perform the next pass is difficult. To ensure maximum coverage of the area (to minimise the area of touch-up grinding), the operator performs a precise series of turning manoeuvres. This ensures that all of the turning area is ground flat. It is computationally unnecessary to implement this in the model as the aim is to remove the high areas across the floor, and it can be assumed that the turning areas will be sufficiently ground flat through the operator's turning pattern and the first outer cut. Therefore, to accommodate this assumption in an efficient manner, special end-condition grind depths were used in the model. At the start and end of a pass, the grind depth was assumed to be a rectangle to cover the area required. This ensures that the whole area is ground and modelled in an effective way.

4.4.4 Adjustable Grinding Head Profile

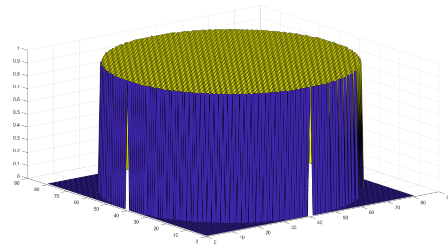
The initial model simplified the grinding head profile to a moving rectangular window relative to the global 2.5D floor surface matrix. However, in 2D the grind profile is donut-shaped (Figure 4-1), and is relatively unknown in terms of the third dimension (depth of cut). This shape and unknown third dimension (depth) can be considered in the model by applying a mask to the rectangular window. The mask provides the ability to remove points not included in the shape (set to 0), and to scale points relative to experimental data to obtain the estimated depth of cut across the profile. This provides modularity to the model, allowing it to be adjusted to suit floor material conditions as well as changing grinding head configurations.

4.4.5 Grinding Path Overlap

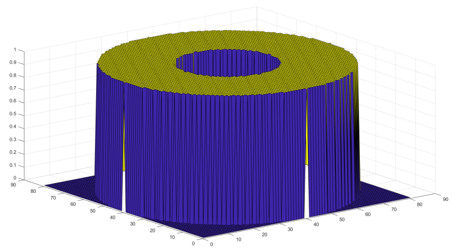
To give a smooth final result, the grinding process involves overlapping passes. With the assumed rectangular grinding head, all parts of the floor under the head get cut at the same rate using a single path, which is not the case in reality. Implementing an overlap can lead to incorrect results, as the overlapped areas receive significantly more grinding. Depending on the direction of rotation of the diamonds, the grinder can cut more on the inside of the profile or more on the outside [43]. This can be applied to the simulation through the use of a mask on the grind window. The mask can take any shape as input, for example a complete filled circle or a more accurate donut shape. Figures 4-7a, 4-7b and 4-7c illustrates some examples of cutting masks that can be applied to the grinding process to analyse varying performance and ideals. Figure 4-7d shows a donut mask with an exponential profile applied in the x direction, accounting for variation in the depth of cut across the grinding area. The 3D depth can be confirmed through experimentation to produce an accurate analysis.



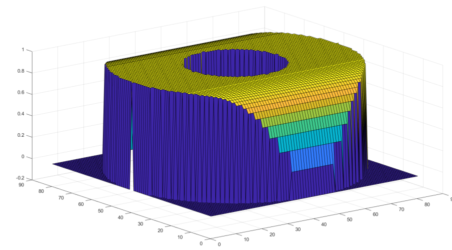
(a) Simple rectangular profile



(b) Circular mask



(c) Donut mask



(d) Donut mask with exponential profile

Figure 4-7: Test surfaces used for mapping

4.5 Overcoming Issues from Assumptions

4.5.1 Grinding Area

A key assumption is that the grinding head grinds each part of the target head area at each time step, whereas, in reality only parts of this area are ground due to the rotational velocity and the tool position. Tool position can be modelled with respect to time; this can then be used to estimate the amount of material removed with respect to time, in addition to estimating which parts of the target area are actually ground. The depth of the grind is not consistent across the grind profile, and this would need to be considered. While this is a relatively unknown area of research, it can be estimated using experimental data.

In addition, the grinding head does not maintain even contact with the surface and has a large surface area, so cannot grind the surface of deep grooves. A surface can only be ground if a diamond cutting edge can come into contact with it; this naturally removes high points in the surface. However, capturing this behaviour in a model is difficult and computationally expensive. The pose and orientation of the robot may be captured using a dynamic model of the machine or through the use of a physics engine, but both of these are computationally expensive solutions and also themselves entail a number of further assumptions. The behaviour of the grinding head can be approximated in two ways: through assuming that the head follows the surface in such a manner where it sits at a tangent to the curve along a single axis; or by assuming that the head is controlled and is always perfectly flat, thus only grinding those parts of the surface that it can reach from this assumed plane. There is potential for future development of the machine to control the head in a manner that allows more precise control, rather than the semi-floating-head approach seen in today's machines. Such development could increase both productivity and produce better results. The floating head assumption and the controlled head assumption was therefore modelled and tested for comparison.

4.5.2 Floating Head Assumption

The grinding head can pivot on the yaw axis only, in that it can only move up and down along one axis. The perpendicular axis is fixed to the chassis of the machine, and so is limited to the motion of the machine. If the machine is on a slope, the grinding head will therefore be on the same slope (assuming a rigid body and stiff connections). As the machine is operating on a fairly flat floor, we can assume that the angle along the roll axis is negligible and can be ignored. We thus must only consider the grinding head pose in the x, y, and z and the orientation in the pitch axis. This can be achieved by analysing the surface gradients in the direction of robot motion under the grinding area.

This was implemented in the simulation by finding the maximum point along each y axis of the grinding head, within the region of stability. This reduced the calculation for pose estimation from 6 dimensions (x, y, z, roll, pitch, yaw) to just 4 dimensions (x, y, z, yaw). From the maximum y value, the gradient to all other points in the grind window along the y direction was calculated. The point with the smallest slope was considered to be the second point of contact, and therefore, the slope estimates the pitch of the grinding head. From the slope, a mask that adjusts the cutting depth of the diamond based on the distance along the grinding head is created. This helps estimate the change in grinding due to the undulations of the floor surface.

4.5.3 Controlled Head Assumption

Under the assumption that the grinding head is controlled in such a manner that the head is always perfectly flat, the cutting area can be estimated with reasonable accuracy. The head will sit on the highest point in the grinding area, and so only areas within a certain limit will be ground. This limit is determined by the height of the diamonds in the cutting tool. The height of the diamond protruding from the surface provides the cutting depth, and so estimates the maximum depth that can be cut in that time step (Figure 4-8).

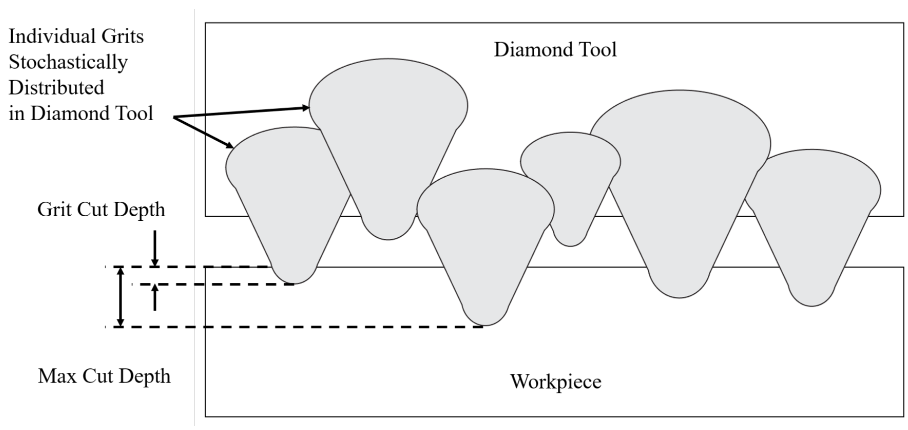


Figure 4-8: Diagram of diamond tool cutting depth

This was implemented in the model by applying a second mask to the grinding area. The highest point in the area was calculated and this, along with the diamond cutting depth, allowed calculation of the minimum height that could be cut. The mask then applied a check to determine which areas could or could not be cut with respect to the depth of the diamond. Applying this mask to the calculation results in the grinding of only those areas that can have diamond-surface interaction. This methodology provides a quick method of changing various parameters, by simply changing the mask and hence changing the grinding area/ depth, etc.

4.6 Testing Process

Three test floors were created using a developed MATLAB random surface generation program (Equation 4.4). Each surface had a maximum deviation of 20-50 mm, similar to measured floors seen in industry according to Jason Torbet. The simulated floors have a width of 500 units and a height of 500 units. The grid size is 0.0 1m. These randomised floor surfaces were used as input into the model. A control model of each floor was run with machine speed kept constant throughout.

The measured surface values were:

- Min initial
- Max initial
- Min output
- Max output
- Initial floor deviation
- Output floor deviation
- Percentage flatness improvement

These variables were used to evaluate the performance of the model with the different grind head assumptions and improvements made. Future work will include testing the best models against real-world grinding. Deviation across the floor will be measured by taking the maximum minus minimum points in the floor area, excluding the turning area. The turning area is excluded to isolate the performance of the control algorithm from the additional end condition assumptions. The grinding head profile used will be rectangular (rect.), circular (circle), and the more machine-like donut shape. The floating head assumption and the controlled head assumption will be simulated and compared. In addition, each test will be performed using a constant speed assumption and a controlled speed.

4.6.1 Measurement Methods

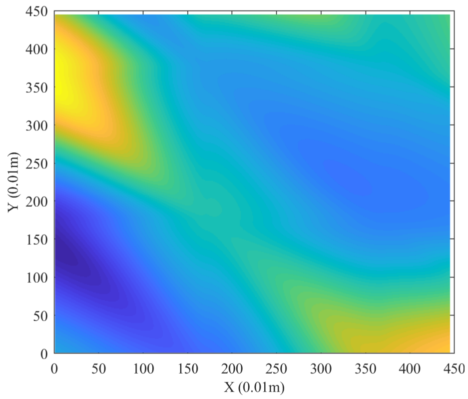
The initial testing was measured by calculating the deviation across the floor for both the input and the output surfaces. The constant speed model was compared with the improved grind shape models by calculating the percentage improvement between the surface deviations. This is calculated by finding the percent difference between the control and the different model variables, with 100% indicating the surface has no deviations, and 0% indicating the surface has not changed. In addition, the surfaces are compared visually for any small changes in how the surface was manipulated.

4.7 Initial Model Results

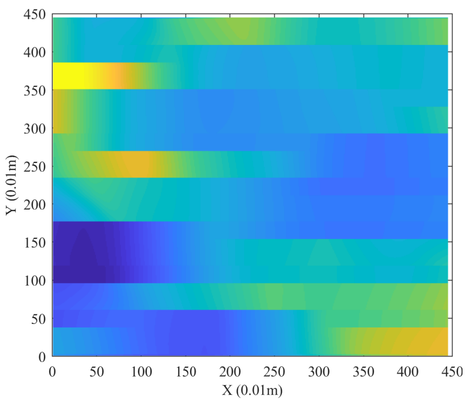
The model was tested initially with a series of randomly generated surfaces. The surfaces were input into the model and the model was run to highlight the changes in the floor. The minimum, maximum, and deviation of each floor surface for each grinding profile is tabulated (Table 4.2 for floating head assumption). The deviation results from the controlled head assumption were identical to those for the floating head assumption. Some exemplar surface outputs are shown in Figure 4-10. Despite the identical deviations, small differences in the contour plots of the simulation surfaces can be seen. It can also be seen that there are significant differences in the output depending on the grind profile used.

4.8 Initial Model Discussion

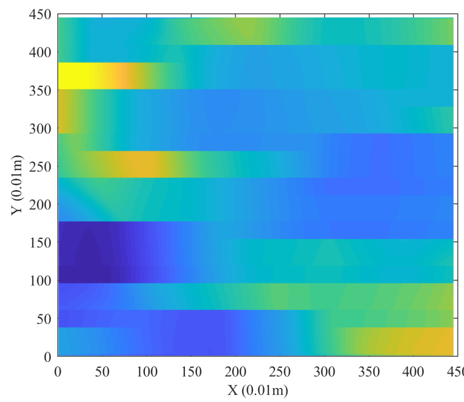
Interestingly, there was no significant difference between the floating head assumption and the controlled head assumption in terms of overall floor deviation. This is probably due to large undulations in the test floors. Visually, there was a small amount of difference, particularly noticeable in the contour plots. A bigger influence on the grinding model appears to be the grind profile, in both 2D and 3D. The depth of cut and the tolerances used for the estimation of which areas can be cut have a much greater influence on the simulation output. From these results, the floating head assumption will be used for comparison with experimental results.



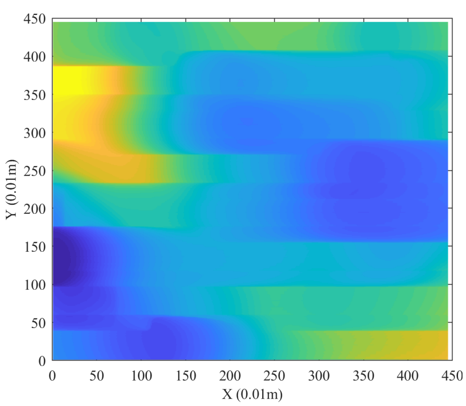
(a) Input surface test 1



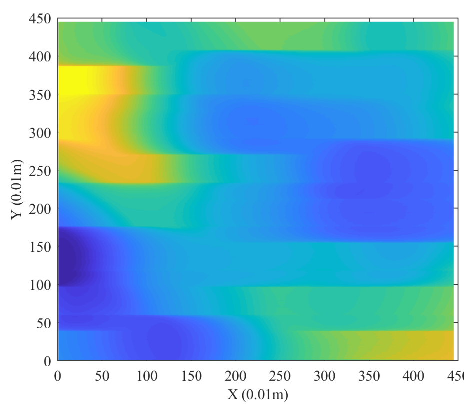
(b) Floating head rectangular grind profile



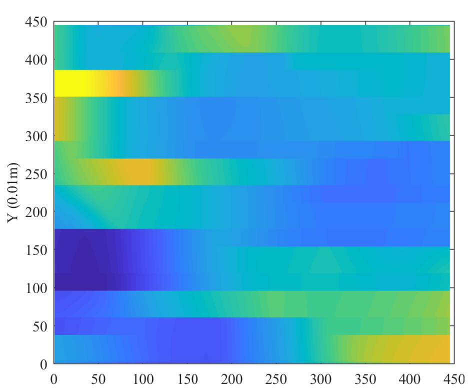
(c) Controlled head rectangular grind profile



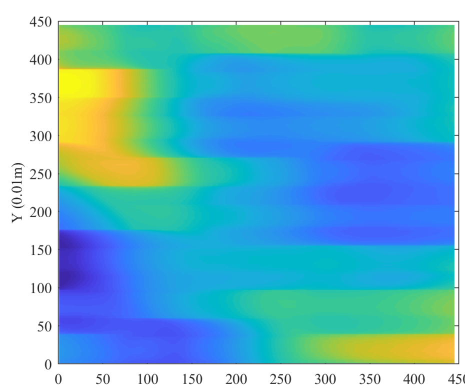
(d) Floating head circular grind profile



(e) Controlled head circular grind profile

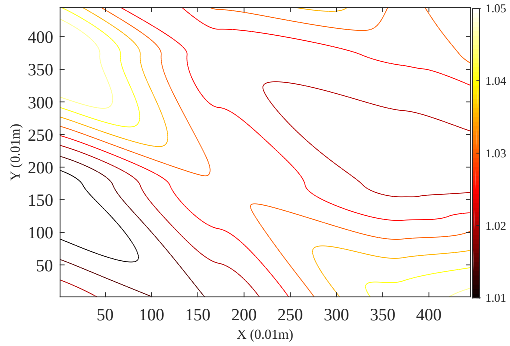


(f) Floating head donut grind profile

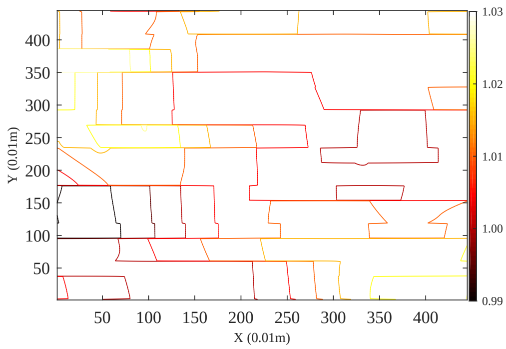


(g) Controlled head donut grind profile

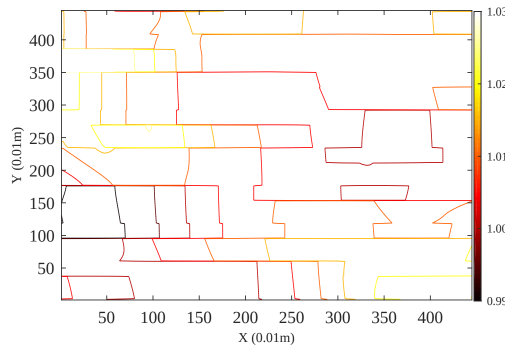
Figure 4-9: Test 1 example output



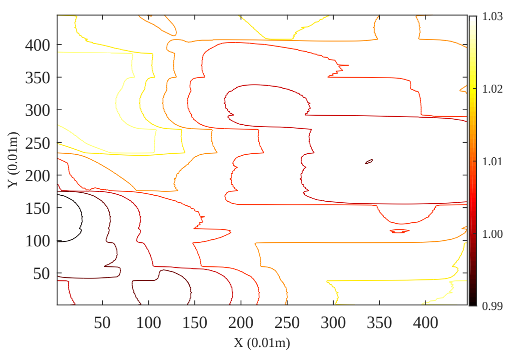
(a) Input contour test 1



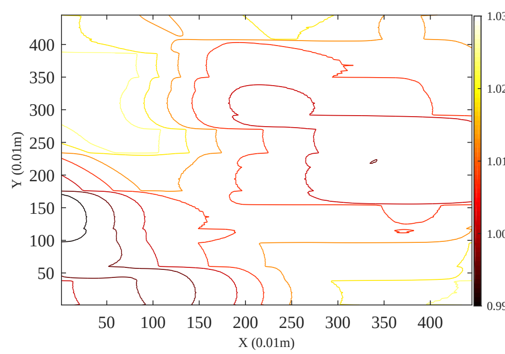
(b) Floating head rectangular grind profile contour



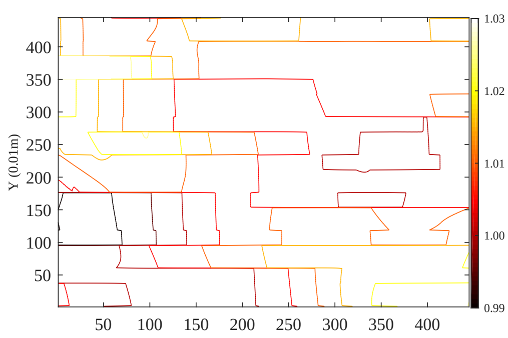
(c) Controlled head rectangular grind profile contour



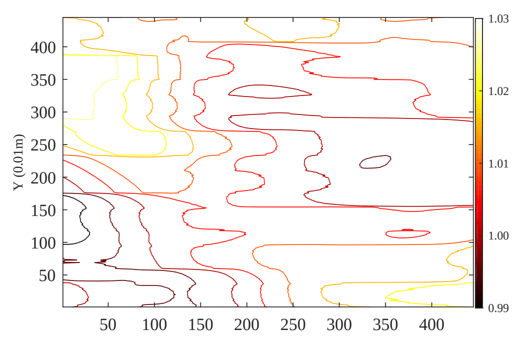
(d) Floating head circular grind profile contour



(e) Controlled head circular grind profile contour



(f) Floating head donut grind profile contour



(g) Controlled head donut grind profile contour

Figure 4-10: Test 1 example contour output

Table 4.2: Deviation across Simulated Floors using Floating Head Assumption

Surface	Test	Max (m)	Min (m)	Range	Percent Improvement %
1	Control	1.0476	1.0003	0.0473	-
	Rect.	1.0418	0.9944	0.0474	-0.2
	Rect. Speed	1.0343	0.9885	0.0458	3.1
	Circle	1.0427	0.9980	0.0447	5.5
	Circle Speed	1.0349	0.9935	0.0414	12.5
	Donut	1.0437	0.9976	0.0461	2.5
	Donut Speed	1.0357	0.9923	0.0434	8.2
2	Control	1.0440	1.0044	0.0396	-
	Rect.	1.0391	0.9967	0.0424	-7.0
	Rect. Speed	1.0333	0.9936	0.0397	-0.2
	Circle	1.0399	1.0004	0.0395	0.3
	Circle Speed	1.0340	0.9970	0.0370	6.6
	Donut	1.0397	1.0000	0.0397	-0.3
	Donut Speed	1.0344	0.9967	0.0377	4.8
3	Control	1.0470	1.0106	0.0364	-
	Rect.	1.0397	1.0022	0.0364	-3.0
	Rect. Speed	1.0322	0.9970	0.0352	3.3
	Circle	1.0410	1.0051	0.0359	1.4
	Circle Speed	1.0331	0.9985	0.0346	4.9
	Donut	1.0415	1.0045	0.0370	-1.6
	Donut Speed	1.0337	0.9981	0.0356	2.2

Chapter 5

Experiments and Results

5.1 Chapter Overview

This chapter presents the results from the series of final experiments performed for this research. The results from initial floor testing can be found in Section 3.4, and those for the initial grinding model development can be found in Section 4.5. In this chapter, the methodology of the final experiments is discussed, followed by the results of each test. The experiments consist of a comparison test between a laser scanner and an RGB-D camera for capturing the floor surface profile of two surfaces, and a validation test of the grinding model.

5.2 Testing of Improved Floor Surface Capture System

5.2.1 Experiment Methodology

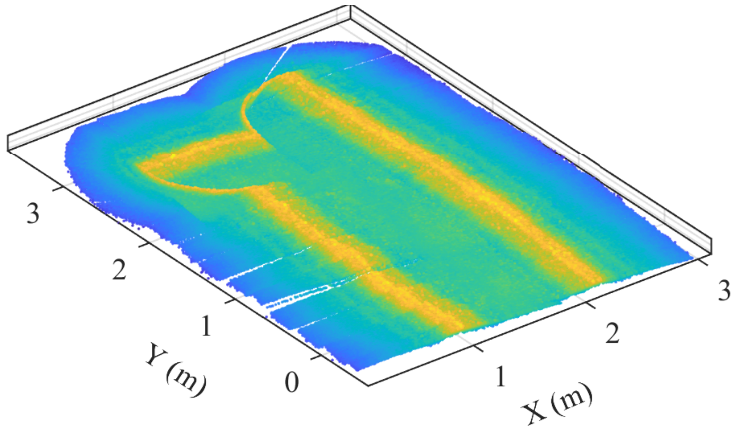
The improved system was tested using two sensors for comparison. One sensor was the D435 RealSense RGB-D camera, and the other the Hokuyo URG short-range laser scanner. These sensors were compared to identify limitations and the appropriate sensor for testing. The test was set up similar to those in Chapter 3: first, a 2 m by 2 m area was marked out using black electrical tape. A 2D map of the area was created using Gmapping. A coverage path for the target area was devised and the robot platform then followed this coverage path, capturing the floor surface profile. The laser scans were assembled using the laser_assembler package and then saved as a .pcd file. The RGB-D camera point cloud data was recorded in a bag file and assembled into an Octomap; the result was also saved as a .pcd file. The test was completed on two different surfaces: carpet and coated asphalt (workshop). AMCL was used to assist localisation throughout the test, and an IMU was used to assist with estimation of the global z height. The point cloud surfaces were processed using MATLAB, and contour plots of the results were used to identify high and low areas. Any identified areas of deviation were investigated visually, by touch, and with a straight edge.

5.2.2 Improved Floor Capture Results

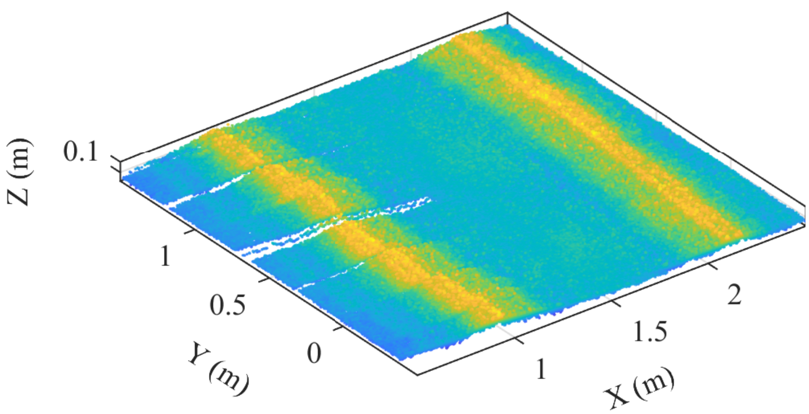
The results of the improved floor capture of the two target surfaces are shown in the Figures below. The floor profiles captured from the laser scan and the RGB-D camera are compared for both the carpet floor and the coated asphalt workshop floor. It can be seen that the laser scan seems to contain a higher level of noise, giving a floor thickness of around 0.02 m. In addition, the laser scan contains consistent high and low measurements regardless of the floor area measured, and are thus systematic errors. The laser scanner fails to detect a number of features indicated by the RGB-D camera, and subsequent inspection reveals that the RGB-D sensor seems to be accurate in its relative floor profile estimation. The changes in height registered by the RGB-D camera are not yet validated and do not seem to correlate with measured deviations via the straight edge. This could be due to the voxel grid size used in the testing.

Capture of Carpeted Floor

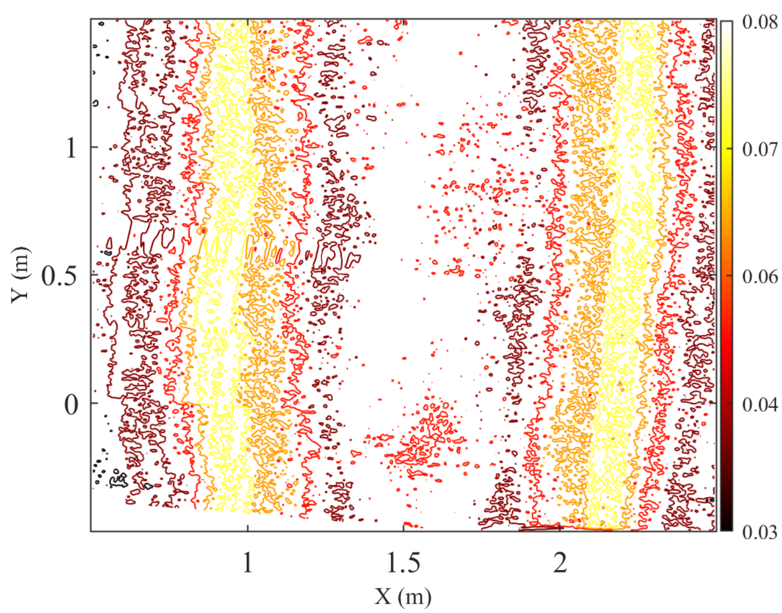
The carpeted floor was successfully mapped by both the laser scanner and the RGB-D camera; however, the laser scanner appears to not have captured some features of the floor as well as introduced systematic errors. The laser scan of this floor (Figure 5-1b) shows a relatively flat floor with high areas down the middle of each pass. These high areas are consistent with robot position, suggesting systematic error rather than measured deviations. In contrast, the floor captured by the RGB-D camera system (Figure 5-2b) shows again a relatively flat floor, but with small deviations in certain areas. Visually, the floor looks flat with no obvious deviations in flatness across the measured floor area. Upon closer inspection of the floor using a straight edge, the high areas captured by the RGB-D camera were confirmed.



(a) Laser scan of carpet floor raw output

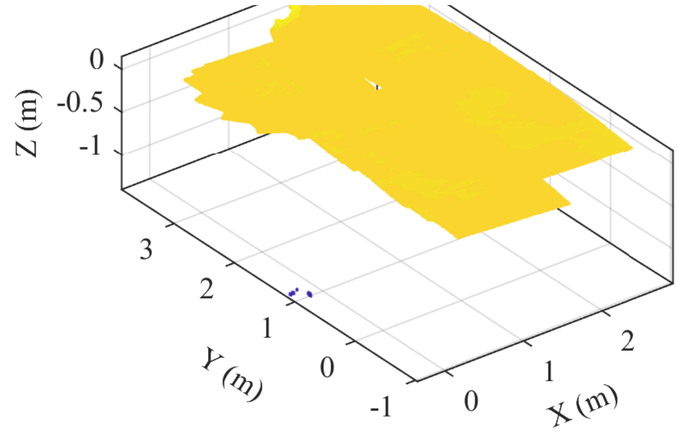


(b) Cropped surface of the laser scan of the carpet floor

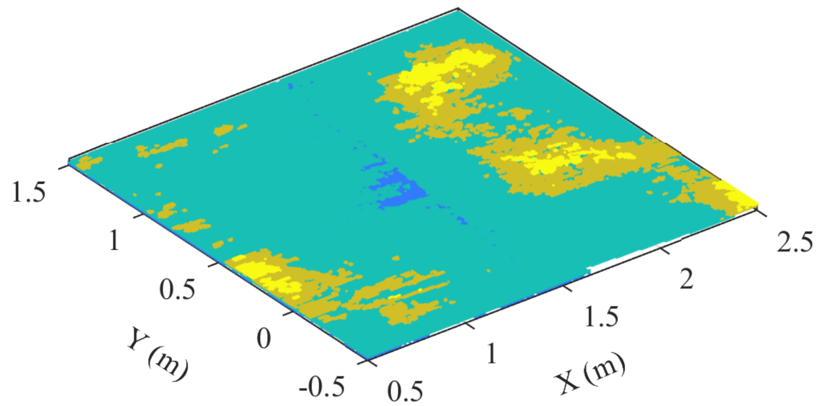


(c) Contour plot of the laser scan of the carpet floor

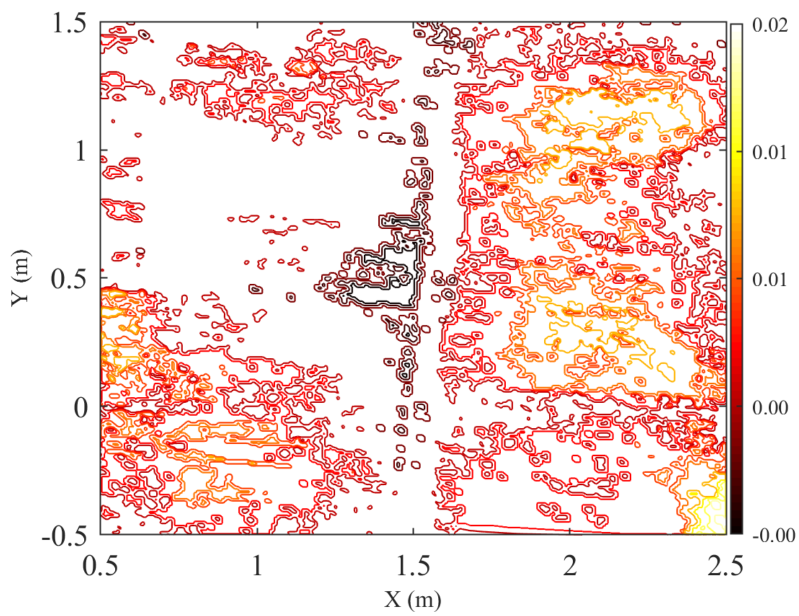
Figure 5-1: Laser scan results for carpet floor



(a) RGB-D capture of carpet floor raw output



(b) Cropped surface of the RGB-D capture of the carpet floor

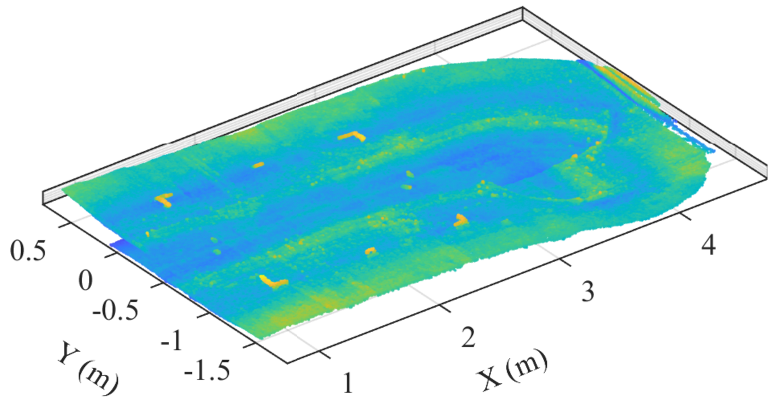


(c) Contour plot of the RGB-D capture of the carpet floor

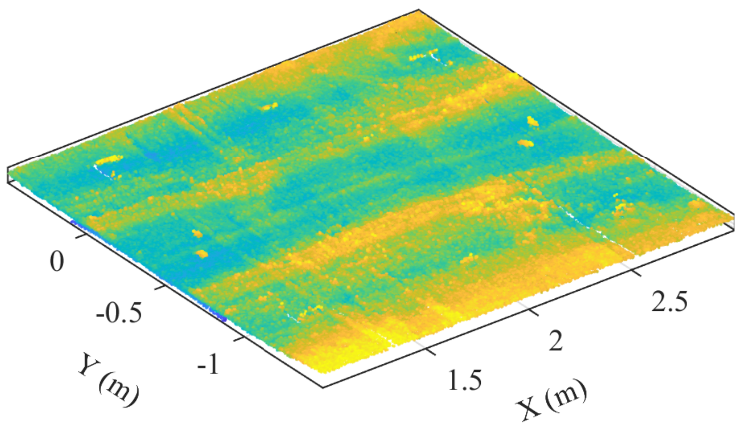
Figure 5-2: RGB-D capture results for carpet floor

Capture of Workshop Floor

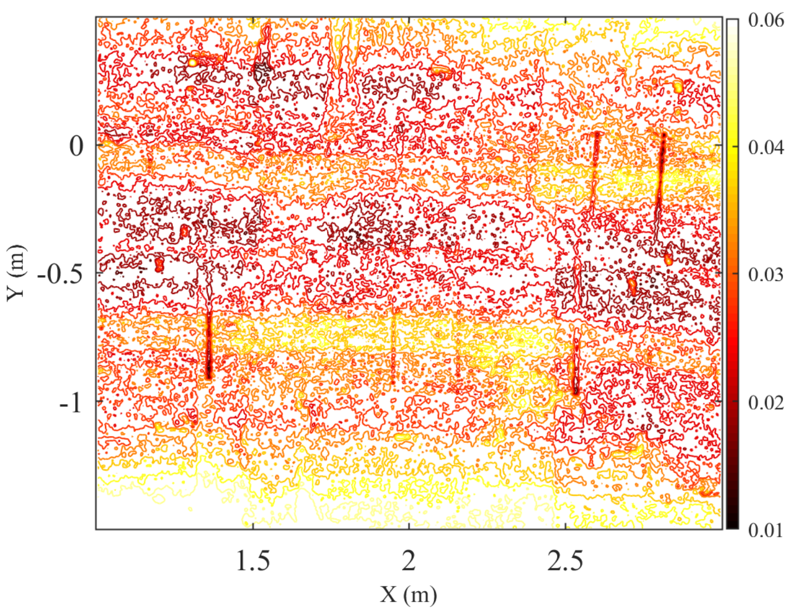
The workshop floor was captured more successfully by both systems. The laser scan continued to show the systematic high area in the middle of the scan that was observed in the carpet tests. Some deviations from flatness were detected, in particular at the sides of the target area. The RGB-D camera successfully captured a number of high areas that were confirmed both visually and with a straight edge. The accuracy of the size of the high areas is not known.



(a) Laser scan of workshop floor raw output

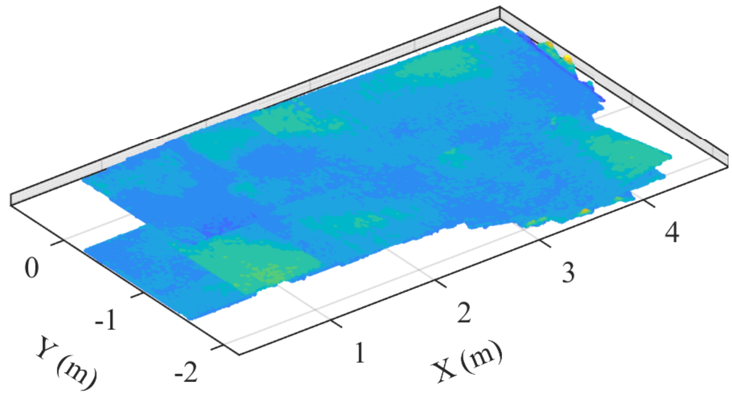


(b) Cropped surface of the laser scan of the workshop floor

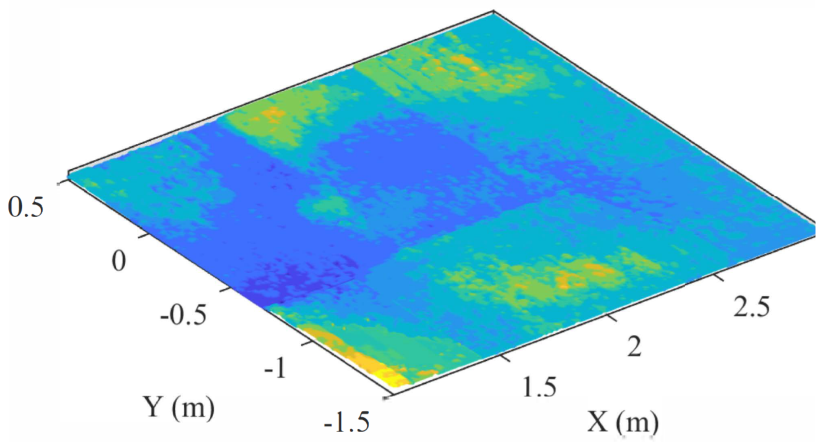


(c) Contour plot of the laser scan of the workshop floor

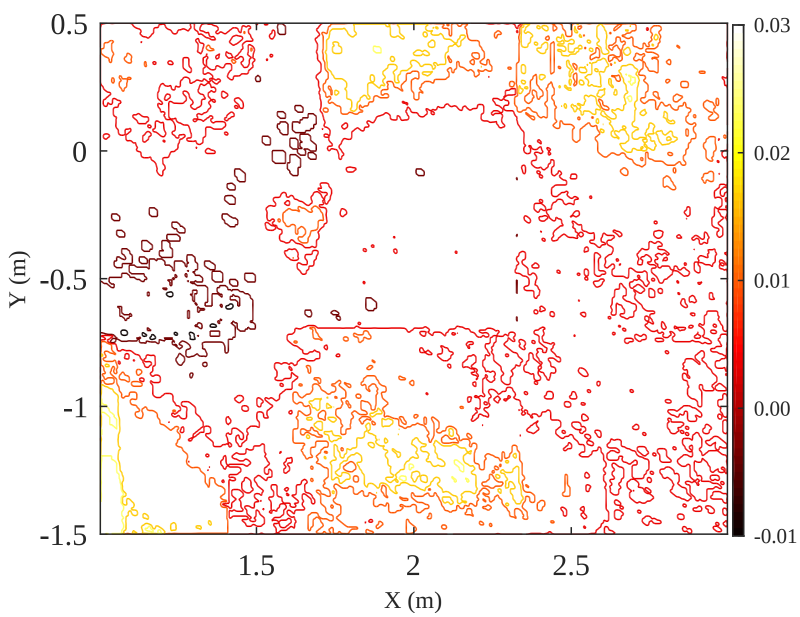
Figure 5-3: Laser scan results for workshop floor



(a) RGB-D capture of workshop floor raw output



(b) Cropped surface of the RGB-D capture of the workshop floor



(c) Contour plot of the RGB-D capture of theworkshop floor

Figure 5-4: RGB-D capture results for workshop floor

5.3 Validation of Grinding Model

The developed grinding model must be validated to allow it to be applied as a successful model. This can be achieved by comparing the model estimations with real-world experimental data. The following sections will discuss the research and development required for implementation to the robot platform. The experiment methodology will also be discussed, followed by the results of a grinding test compared with the results from the model.

5.3.1 Platform Implementation

To validate the model, the research platform was used to capture the floor profiles before and after a grinding operation. The sensor used was the RealSense D435 camera, which has a resolution of 640 x 480 pixels. This camera was selected due to its superior performance in detecting deviations from flatness in the sensor comparison tests performed previously. The camera was mounted on the platform at a height of 0.85 m, giving a floor view of 1.2 m by 0.6 m (Figure 5-5). The camera mount is adjustable, and prior to data capture was manually adjusted to be level in x and y. In addition, the wheel odometry parameters were tuned to minimise any localisation errors. In previous testing, localisation using tuned odometry and AMCL was successful, and so the same process was utilised here. The IMU provided robot pose and orientation information in 6 DOF, which was fused with other robot information to give an accurate odometry estimate. The IMU was used to allow for the floor slope, and a calculation using wheel odometry and IMU pitch estimated any deviations in the z height. This created a full 3D analysis of the floor.

5.3.2 Methodology

An experiment was devised to validate the performance of the grinding model. In this experiment, the research platform was used to perform an initial capture of the floor profile of a 2 m by 3 m floor area. The floor was a concrete floor in an old warehouse (Figure 5-6a). The scanning followed a similar format to the tests conducted in Section 3. First, the research platform created a 2D map of the area using SLAM Gmapping [35, 36] (Figure 5-6b). The research platform then localised itself in the known map using the horizontal 2D laser scan measurements and AMCL localisation. Next, the platform followed a coverage path over the target area, capturing 3D floor profile information with known global positions

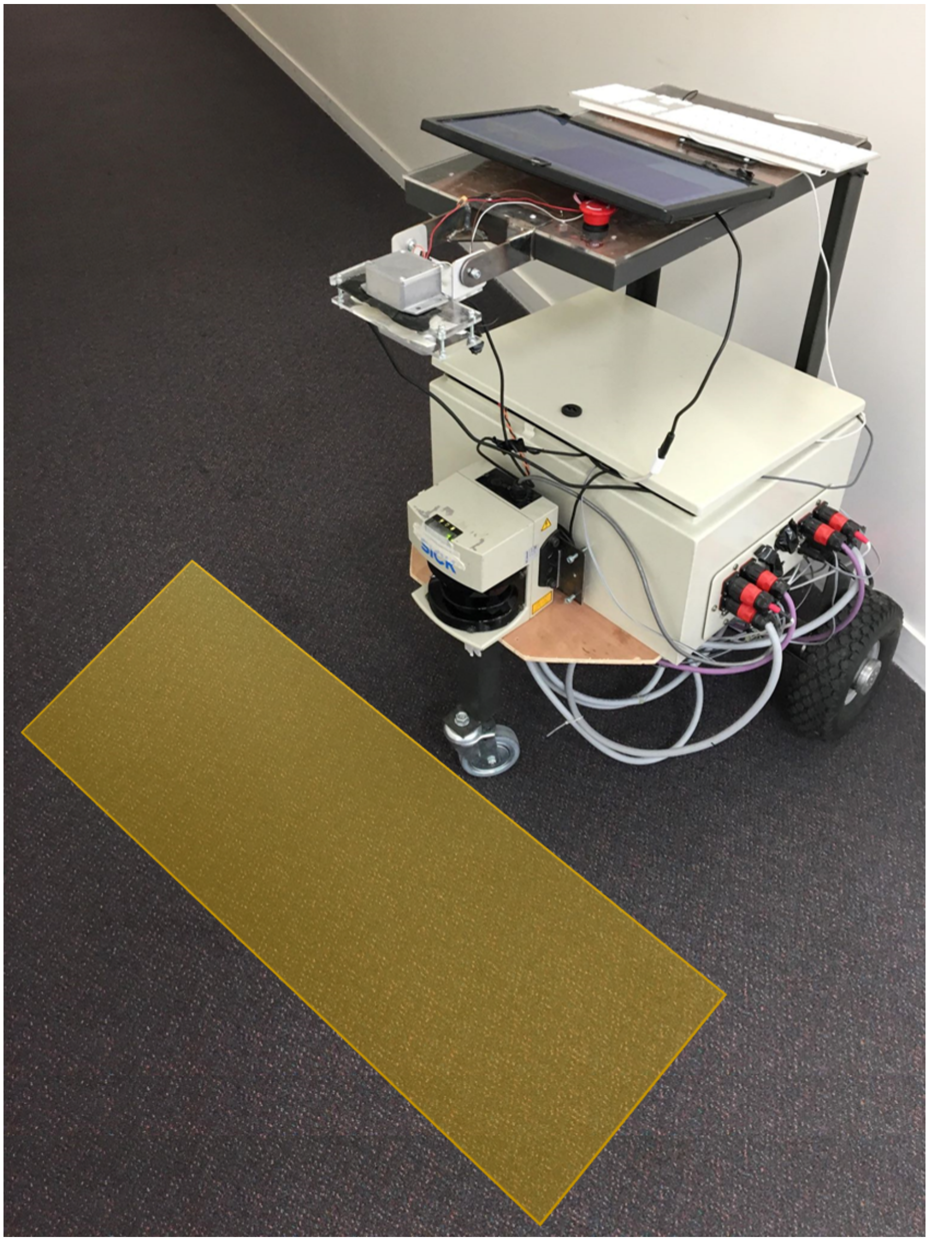
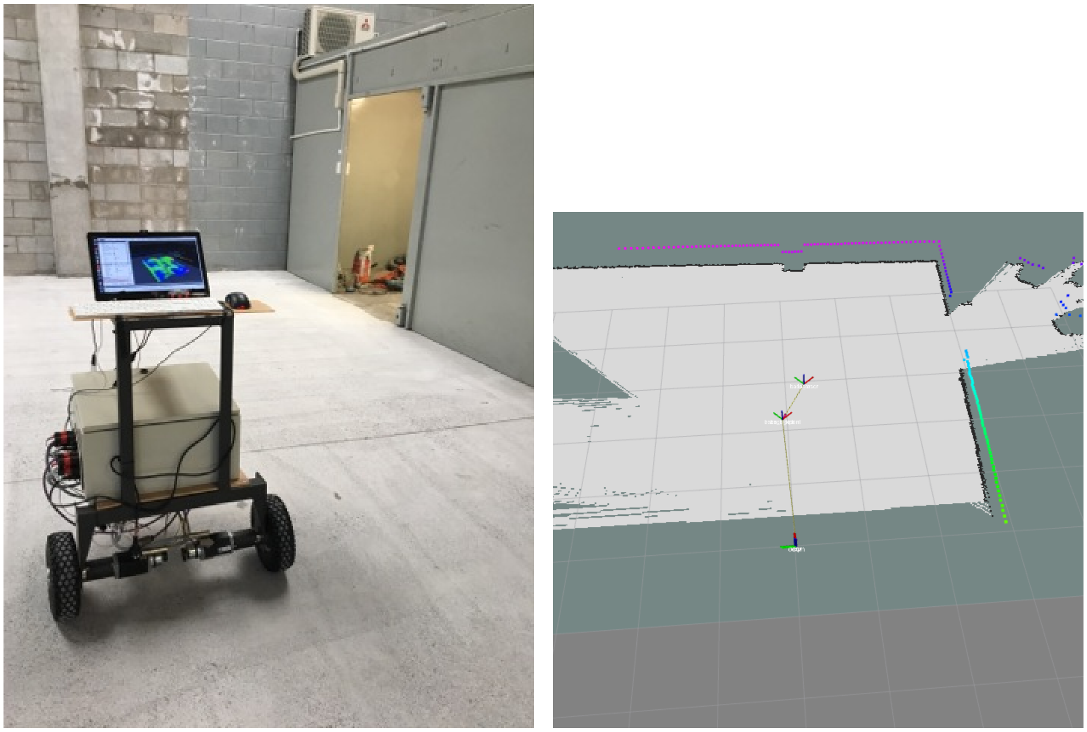


Figure 5-5: Robot platform for floor profile capture, with RealSense field of view highlighted

in the map. The speed of the robot platform was set to a slow speed of 0.1 ms^{-1}. It took the robot around 6 minutes to complete the coverage path.

The floor was then ground by an industry expert a constant speed, with the aim of removing 5 mm of material during this first cut. After the first grind was complete, the research platform was again used to scan and capture the floor surface profile using the same coverage path and system setup as described previously. The point cloud information was captured in a ROS bag file throughout the test for post-processing, in which Octomap



(a) Image of target floor area

(b) Map of target area created using SLAM Gmapping

Figure 5-6: Grinding model validation methodology

was used to assemble each point cloud into a map of the floor using a voxel grid size of 2 mm.

5.3.3 Measurement Methods

The floor surface profile was captured as a point cloud using Octomap and a voxel grid size of 2 mm. This point cloud was processed into a 2.5D grid surface, which was then input into the grinding model. The model was run with similar parameters to those used in the grinding experiment: the speed was constant and the depth of material to be removed was set at 5 mm.

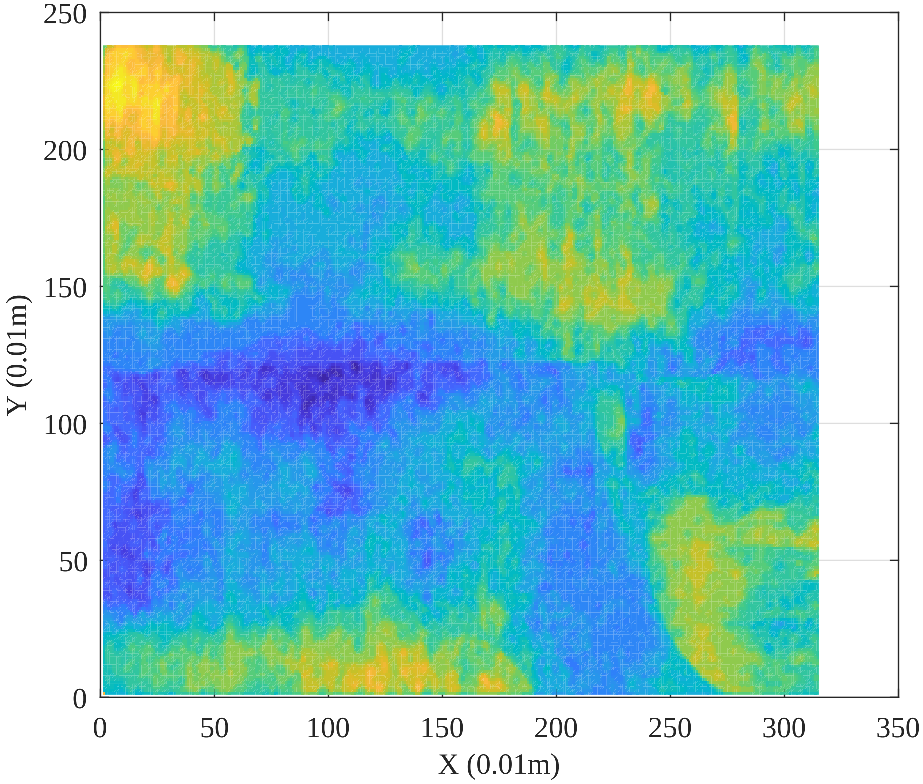
The results obtained from the developed grinding model were compared with the captured surface of the ground concrete floor. This comparison helps to validate the grinding model and identify areas of improvement. Due to resource constraints, the test was only performed on one surface, but multiple scans and multiple grind passes were used to capture as much data as possible. The grinding model was input with similar specifications and performance to the grinder used in the test. For the comparison, the real-world data was cropped and any erroneous shifts were corrected. Comparison was initially performed as a surface correlation. The real-world post-grind data was used as the baseline; an accurate grinding model would have a correlation close to 1. This comparison helps to validate the model and confirm the assumptions used in the basic grinding model. In addition, the change in deviation across the floor was compared.

For improved comparison of the pre-grind, post-grind, and simulated output surfaces, a 2D Continuous Wavelet Transform (CWT) was applied. This provided better comparison of the surface properties, in particular the location and size of deviations from flatness. A correlation of two surfaces does not take into account any shift or rotation in the two measured surfaces, and so is inherently not accurate in estimating the similarity of two surfaces if any shift is indeed present. Due to the collection process, small shifts in measurement are expected as the localisation is not 100% accurate. The slope of the surfaces can be compared; however, the 2D CWT provides an analysis of floor flatness at many different wavelengths. Analysis was carried out using the cwtft2() MATLAB function with a continuous function of wavelengths and a similar process to that of Valero et al. [72]. The MATLAB function returns the position and size of peaks in the floor; for each peak a boundary of the points connected to that peak was created. A fitted bounding ellipse was applied to each peak and its connected neighbours, and if the major or minor axis of the fitted ellipse was the same as the wavelength used to find the peak, then this area was considered to be a deviation from flatness. Each deviation from flatness was then outlined on the surface for visual comparison. The area of each ellipse gives insight into the size of the deviation. Larger-deviation wavelengths can be considered as large undulations in the floor and an overall smoother surface.

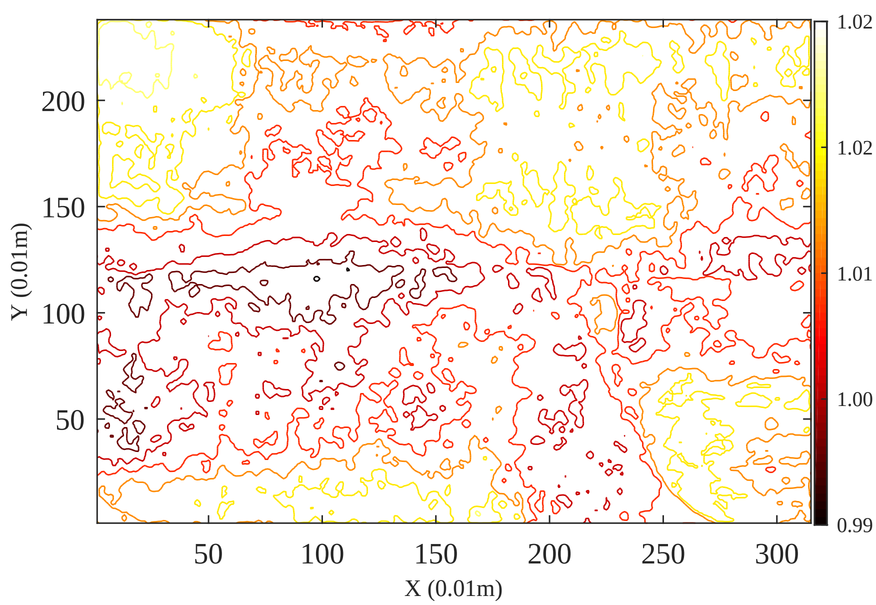
5.4 Results for Model Validation

5.4.1 Floor Profile Capture

The captured floor profiles pre-grind (Figure 5-7a) and post-grind (Figure 5-8a) are shown below. The deviation across the floor is tabulated in Table 5.1. Pre-grind, the average measured deviation along the floor is 0.0375 m and post-grind it is 0.0255 m; a significant reduction in floor deviation. Visually, the contour plots (Figure 5-7b and 5-8b) show a significant improvement in terms of the reduction of high areas, particularly on the upper half of the floor. The maximum and minimum heights measured for the pre-grind and post-grind floors are relative to the global robot frame, and so cannot be directly compared. A global frame would be required to enable comparison of the maximum and minimum heights. Accordingly, removal of material across the floor can currently only be estimated and compared relatively, not quantitatively, at each point.

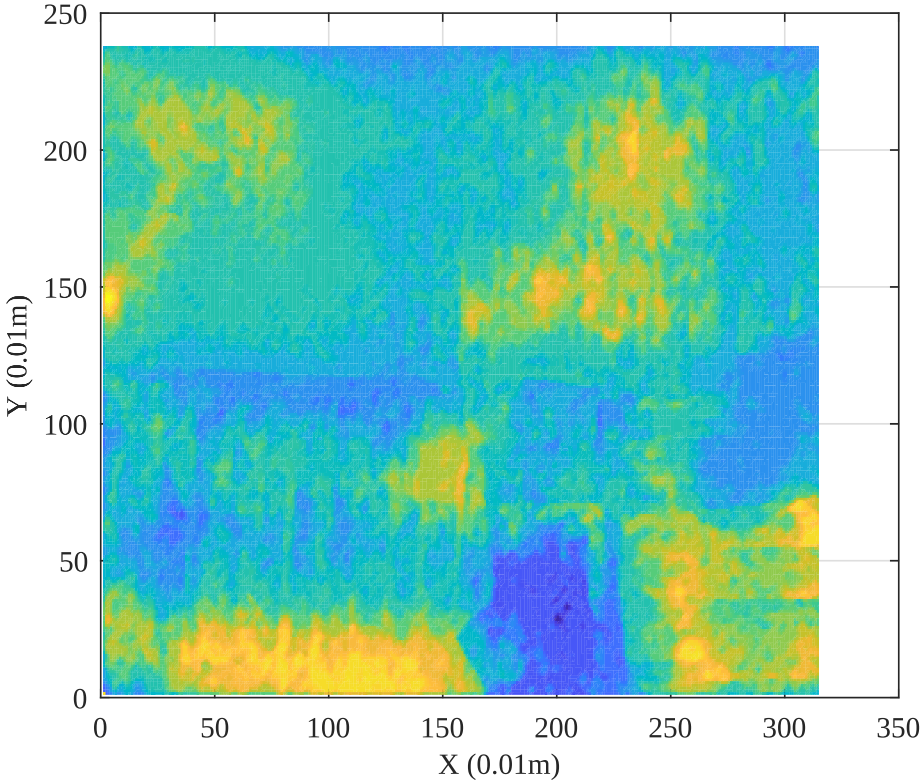


(a) Pre-grind cropped surface

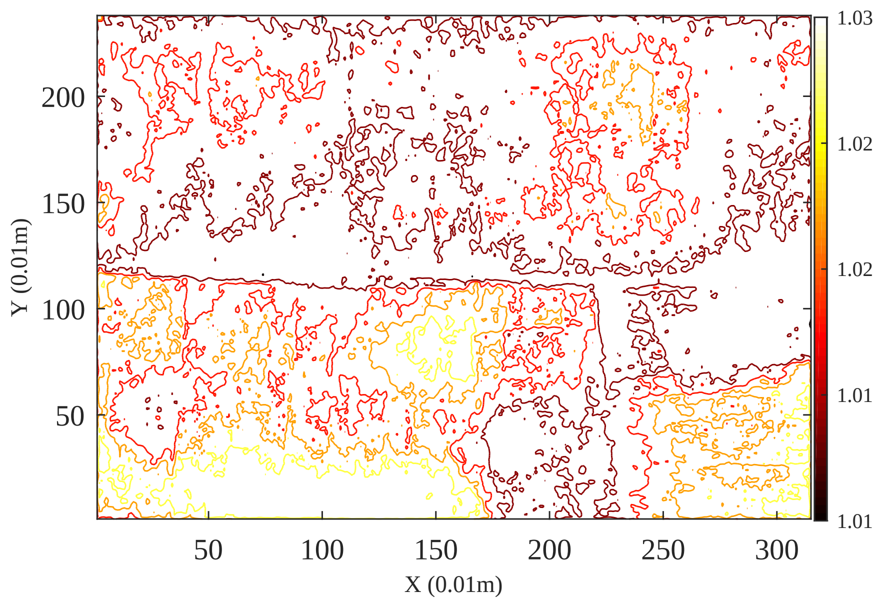


(b) Pre-grind contour plot

Figure 5-7: Pre-grind floor surface capture results



(a) Post-grind cropped surface



(b) Post-grind contour plot

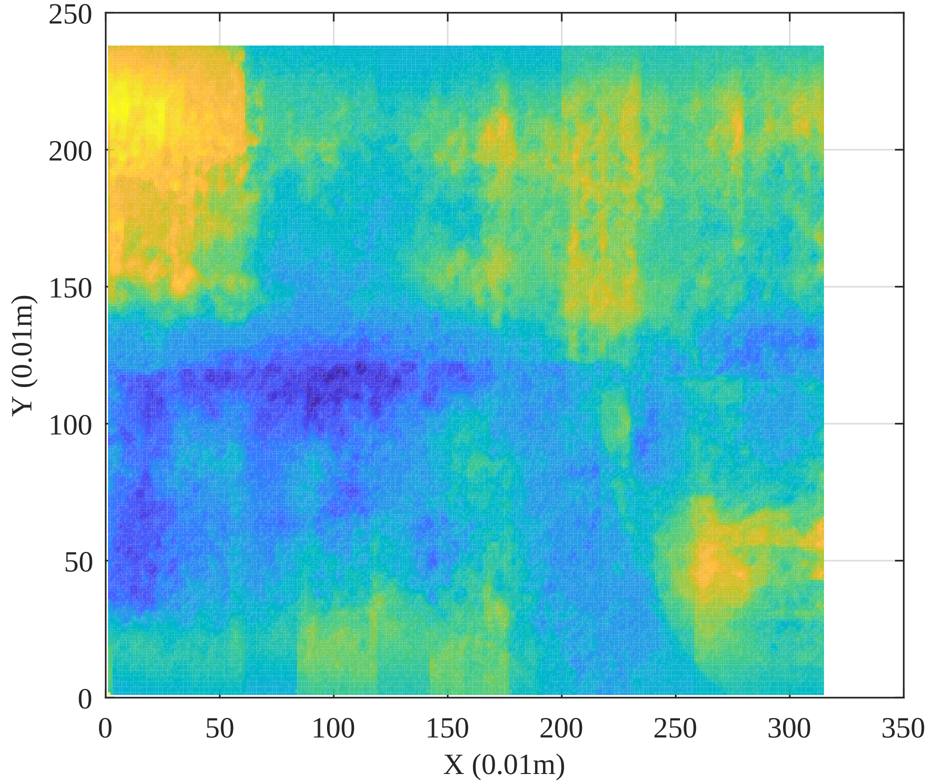
Figure 5-8: Post-grind floor surface capture results

Table 5.1: Deviation across Validation Floors for Pre-Grind, Post-Grind, and Simulation

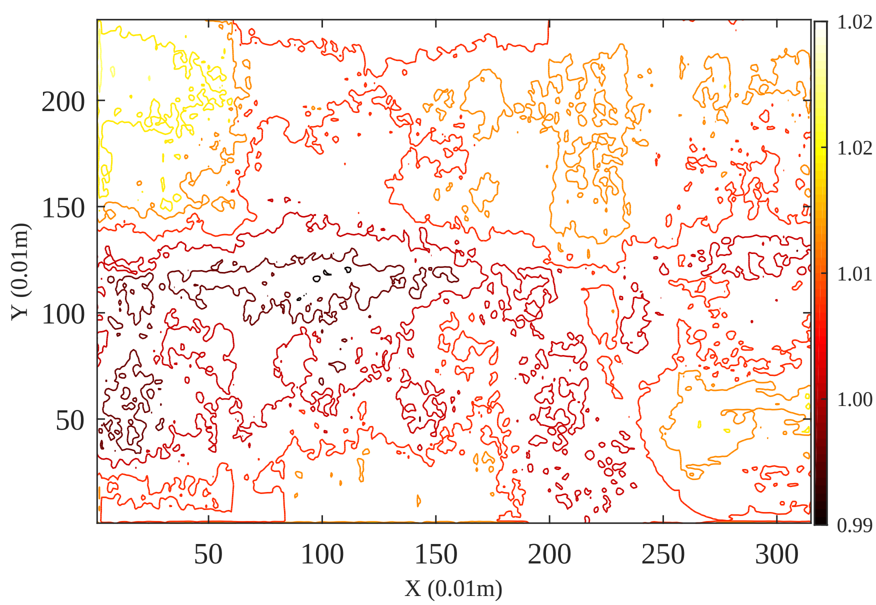
Surface	Max Height (m)	Min Height (m)	Range (m)
Pre-Grind	1.0269	0.9894	0.0375
Post-Grind	1.0223	0.9968	0.0255
Simulated-Grind Floating Head	1.0201	0.9894	0.0306

5.4.2 Grinding Model Simulation

The results for the grinding model simulation are shown on the following pages. The output of the simulation using the floating head assumption is shown in Figure 5-9a and a contour plot of the surface is given in Figure 5-9b. It can be seen that material has been removed, particularly around the high areas of the floor, and that the floor has become somewhat 'flatter'. This is supported by the average deviation across the floor being reduced to 0.0306 m for the simulation (Table 5.1). The simulation result will now be compared with the pre-grind and post-grind experimental results.



(a) Post-grind simulated surface; floating head



(b) Post-grind simulated contour plot; floating head

Figure 5-9: Simulated floating head floor surface results

5.4.3 Model Comparison

For comparison of the grinding model simulation results with the experimental data results, the surface was mapped onto a 0.01 m grid for input into the grinding program and subsequent analysis.

The correlation between the floating head simulated output surface and the measured output surface is only 48.1%; however, the measured difference between the slopes of the floors provides insight into the general flow of the floor. 65% of the slope difference between the simulated and the measured output surface falls within 1 standard deviation of the output floor surface 2.9-degrees. We can therefore state that the simulation floor can represent 65% of the measured floor fairly accurately, and so is an average representation of the real-life data. The surfaces were compared using Continuous Wavelet Analysis; this analysis is discussed in Chapter 6.

The correlation between the measured input surface and the measured post-grind surface is only 55%, indicating that there is a significant difference between the pre-grind and post-grind surfaces. This could also suggest small differences in floor alignment, which would result in a poor correlation.

Normalised difference plots were calculated by subtracting the normalised output surface from the normalised input surface. Normalising the surface removes differences caused by the global z height inconsistency. The plots highlight the areas where the model differs from the real-world experiments, and gives insight into how the model could be improved. The difference between the pre-grind and post-grind floor surfaces is shown in Figure 5-13 and that between the pre-grind and simulated output floors is shown in Figure 5-14. Difference can be seen in particular in the top left-hand corner of the figures, where the simulation failed to remove the amount of material removed by the grinding machine. In addition, the lines from the simulated grinding machine path can be seen in the profile. This is not observed in the experimental results, which have a much smoother surface result.

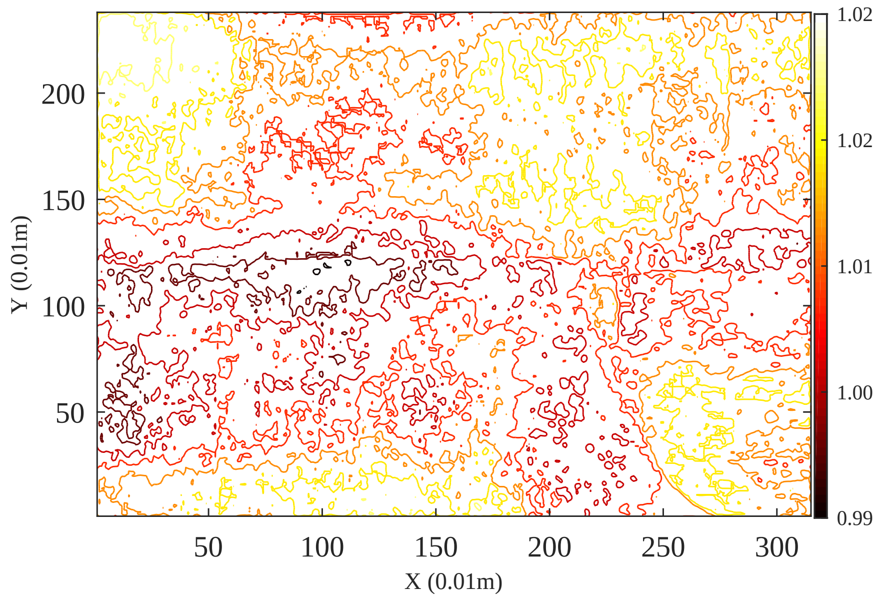


Figure 5-10: Floor surface pre-grind

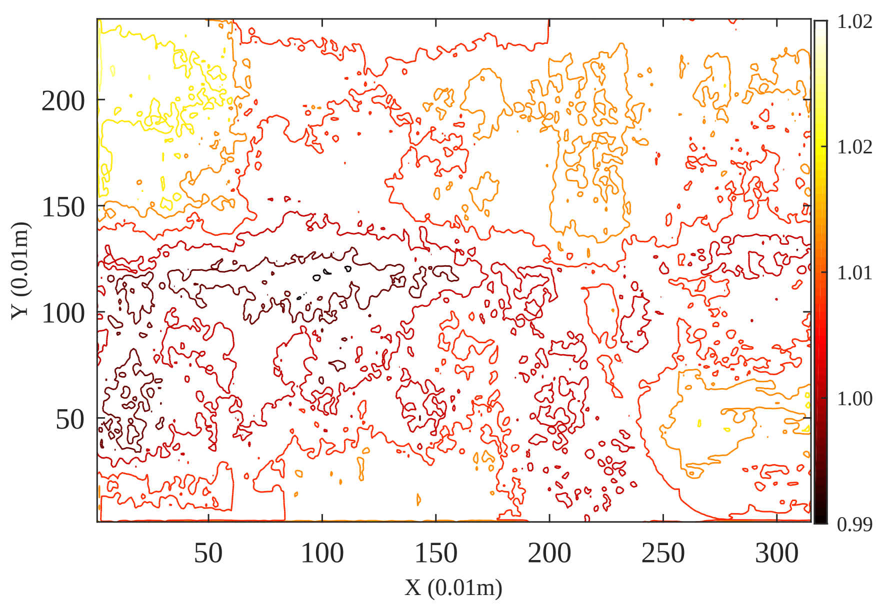


Figure 5-11: Simulated floor surface post-grind; floating head

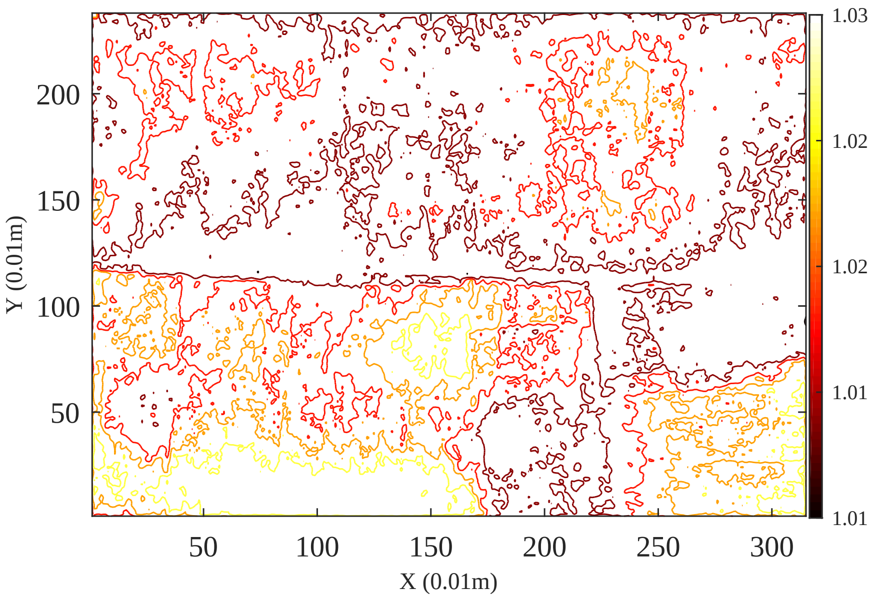


Figure 5-12: Experimental post-grind output for comparison

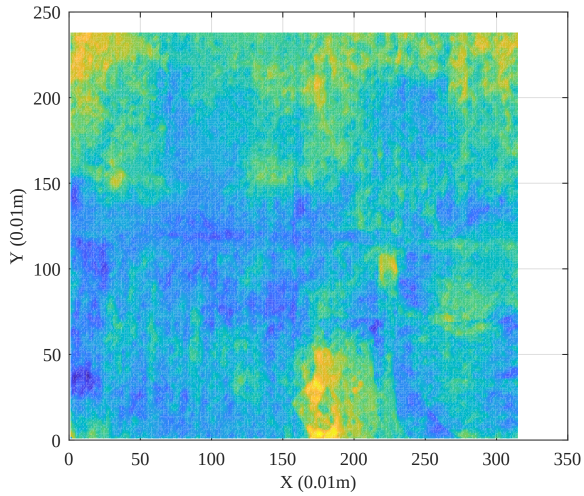


Figure 5-13: Normalised difference between pre-grind surface and post-grind surface

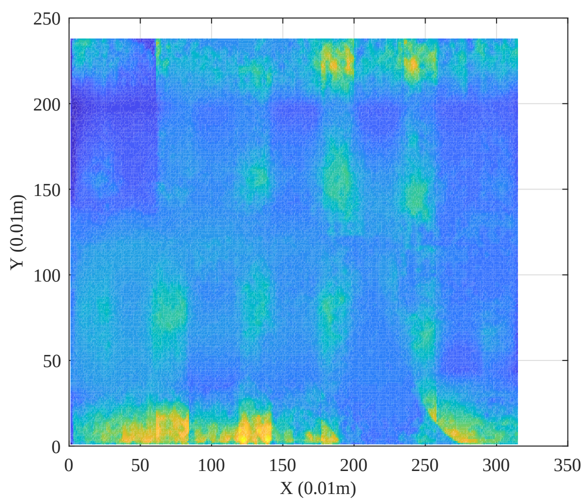


Figure 5-14: Normalised difference between pre-grind surface and simulated output surface

Chapter 6

Discussion

6.1 Chapter Overview

This chapter discusses the results of the experiments, including thorough discussion of the performance of the various sensors in capturing the floor profile, and the limitations and improvements. In addition, this chapter describes the outcomes of model validation and provides some suggestions as to the cause of the low correlation observed. Further improvements to the system and to the grinding model are also described.

6.2 Floor Profile Creation

The floor profile creation system was successful. The methodology was able to capture areas of interest in the target zone and provide an overview of the mapped floor. The RGB-D camera provides a larger field of view and this gives greater insight into the floor features; in particular, it is able to better capture both local and global areas of interest. The laser scanner has limitations surrounding the level of noise present in the measurements as well as material surface reflectivity and deflection errors. The laser scanner is sensitive to a change in material, which could introduce errors in some applications. The RGB-D camera also has some limitations; in particular, it must use auto-exposure in dynamic lighting conditions. If the camera is incorrectly exposed, the resulting point cloud and depth images can be poor. In contrast to the laser scanner, the RGB-D camera is less sensitive to different materials.

This system can be used for identifying points of interest in a large area, such as high points or low points. The scanning method does not provide sufficient accuracy for applications that require micrometre or millimetre resolution; however, it can produce a fast scan for a large area, and a second, high-resolution scan could then be applied to key areas. Further, the profile mapping system can autonomously provide a quick means of producing an overview of a large area. The floor capture method is able to identify areas of interest, although, the accuracy of the estimated deviation from flatness is not yet validated.

6.2.1 Sensor Comparison

Overall, the RGB-D sensor captured more features of the floor and suffered from less systematic errors. Some errors due to camera alignment were observed. This is because the camera is required to be initially perfectly level for the system to accurately create the floor profile. In addition, the RGB-D camera was able to detect and highlight high and low areas, however, the scale of these high and low areas is yet to be validated.

The short-range laser scanner continued to detect the black tape as a higher section of the floor, particularly in the workshop test. While the RGB-D camera did not detect the black tape to the same extent, the tape could be visually detected at some points throughout the test. The laser scanner measured consistent high areas that did not change with floor profile and robot position, and therefore can be considered to be systematic errors of the laser scanner itself. These could be overcome through a thorough calibration process. Due to Octomap point cloud stitching, this error as well as the surface thickness error is greatly reduced in the RGB-D sensor tests. The laser scanner produced a surface thickness of around 0.02 m when scanning the workshop floor. Due to the voxel grid approach, the RGB-D sensor produced a floor thickness of one voxel, 0.01 m.

6.2.2 Floor Capture Capability

The RGB-D camera was able to highlight areas of interest, particularly high and low areas of the floor. While some errors were observed, the overall floor deviations appear to have been estimated. In the carpet 2 m x 2 m test, the sensor detected a high area in the middle of the first pass (Figure 6-1). This area was analysed after the test using a straight edge and was confirmed to have a deviation from flatness of around 2 mm over 200 mm of floor (Figure 6-2).

Similar performance was observed in the workshop floor experiments. Areas in the floor that had visually detectable deviations were successfully detected by the RGB-D sensor

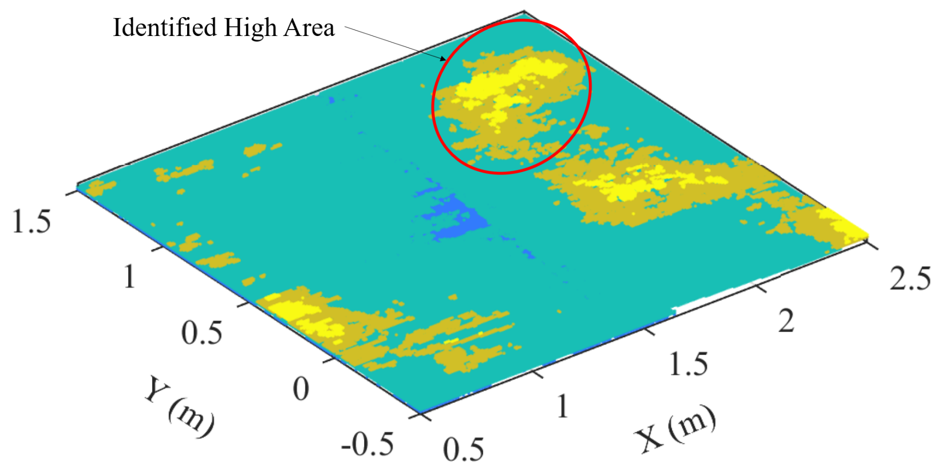


Figure 6-1: Captured high point in carpet floor

Figure 6-2: Investigation of corresponding high point in carpet floor using straight edge

capture process. There was a significant bump in the workshop floor marked in Figure 6-3 that was also successfully detected during the floor capture process (Figure 6-4). A second area of interest was investigated using a straight edge and was also confirmed to be a deviation from flatness (Figure 6-5). The accuracy of these detected high areas is yet to be validated.

6.2.3 Surface Thickness

In an ideal world with ideal sensors, the measured surface profile would have a thickness of a single measured point. However, it has been observed that the thickness of the point cloud

Figure 6-3: Workshop floor with significant bump highlighted

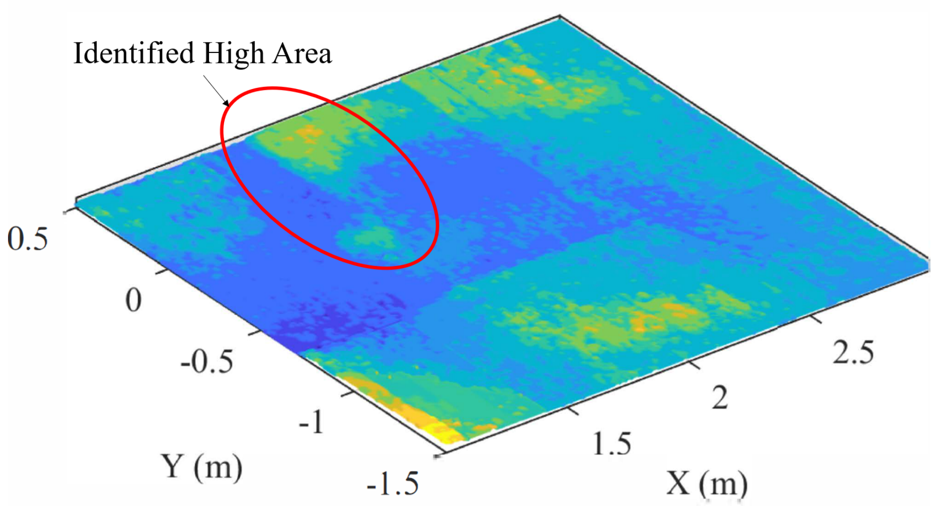


Figure 6-4: Capture of workshop floor high point

surface varies between floor types, with carpet producing the worst thickness. Such point cloud thickness is due to the accuracy of the laser scanner (± 1 mm) and the reflectance and diffraction of light by the surface. However, this thickness is not an impediment and it is still possible to identify areas of interest in the floor.

Figure 6-5: Investigation of high point in workshop floor using straight edge

6.2.4 Sources of Error

Some sources of error have been identified through analysis of the results and the robotic system. A key source of error is the floor sensor itself producing significant noise in measurements. The noise was observed in both of the sensors used, the laser scanner and the RGB-D camera. Octomap stitching of the point cloud data helps to reduce the effect of noise through collecting many frames of data. The frames of data are stitched together and the ray tracing method removes any erroneous measurements from previous frames. Due to the high frequency of frames (between 10 and 30 frames per second, depending on CPU load), and the slow movement of the scanning platform, this process successfully reduces the noise in the created floor profile.

The IMU is subject to erroneous measurements and drift over time. This can result in a slow change in the z height over time, or, as observed in some tests, in incorrect readings when stopping or starting suddenly. In the testing, this was overcome through careful control and slow acceleration / deceleration of the robot, but remains a source of error that must be considered and mitigated. Any errors in height estimation can lead directly to errors in the captured floor profile. In future work, this error will need to be overcome to produce an improved system.

6.2.5 Sensor Selection and Limitations

Although the captured floor profile aligns with manual inspection with using a straight edge, the floor profile capture system and methodology have not yet been validated. Validation could be achieved by using a 3D TLS and comparing the captured floor surface of the two systems. This is out of scope for this project due to resource limitations, and remains as future work to be done.

Material Reflection and Laser Scanner

According to the literature surrounding the use of laser scanners, the reflectivity of the material being measured by a laser scanner is a highly important variable [12]. Tests performed throughout the research demonstrated this effect. During the initial floor capturing tests, the black tape used to mark the boundary of the target zones was found to be measured as significantly higher (for the carpet and coated asphalt tests) and significantly lower (for the asphalt tests), due to the reflectivity of the black tape compared with these three materials. In addition, the surface thickness captured for the carpet surface was significantly thicker than for the other surfaces, suggesting a greater amount of noise. This could be due to the fibres in the carpet causing measurement errors and changes in the deflection of the laser.

Light Interference and RGB-D Sensor

A limitation of a RGB-D camera is its sensitivity to light. Due to the projection and capture of infrared light, natural light can interfere with the readings. This can be calibrated and adjusted for through camera settings for indoor and outdoor applications (Section 3.7). However, this issue persists in applications where the camera is exposed to both indoor and outdoor lighting in a dynamic environment. The camera can continuously auto-expose; however, this can introduce other errors and takes time to complete. Auto-exposure can thus result in erroneous readings. This limitation will have to be considered during application.

6.2.6 Justification for Improvements

Throughout the development and testing of the research platform, a number of system improvements were made. Many of these were implemented after initial testing revealed limitations in the application and through the literature. Two key decisions are justified in the following sections.

2D Extrapolation Limitations

AMCL localisation using a horizontal 2D laser scanner can provide a relatively accurate method for global localization; however, this only operates in 2D. In order to create an accurate 3D map of a surface, the full 3D orientation and position of the robot is required. This is often achieved by using expensive 3D sensors such as 3D Terrestrial Laser Scanners, but this research aims to utilise cheaper sensors to achieve similar results. In order to extrapolate the 2D location into the 3D environment, the 6 DOF data from the IMU is fused with odometry and AMCL data. This gives the robot location on a 2D map together with the full orientation (roll, pitch, and yaw). However, it does not address the z height of the robot. The z height must be taken into consideration in order for an accurate surface to be created, as illustrated in Section 3.6.

There are limitations to the methodology applied, in that the IMU is often subject to drifting, and therefore exposes the robot's 3D location to additional noise. Location noise combined with noise from measurements could result in significantly inaccurate measurements. These limitations remain a challenge and this remains an area for further research improvements. A potential solution is to apply a point cloud registration matching algorithm for z adjustments. This algorithm could utilise common methods of ICP to identify the best-fit point cloud when adjusting only the global z value of the point cloud. This solution works best with a large number of features, whilst a floor typically has minimal features. This challenge therefore remains to be overcome.

Localisation

Global localisation was significantly improved throughout the development. Tuning of odometry parameters and utilisation of the AMCL node helped increase localisation reliability and reduce jumping in x and y position estimation. The localisation method used was an adapted 2D approach, where 2D solutions such as finding the x and y global position through AMCL was applied and then extended through the use of an IMU and position estimates. This proved to be an acceptable solution that was capable of successfully mapping a ramp, although the accuracy of the 3D extrapolation could be improved as part of the further development of the robot system. In particular, the use of the IMU makes the system prone to drift and resulting errors. This could be overcome using additional visual sensors, such as a camera for orientation estimation, or through an improved IMU.

6.3 Grinding Model

6.3.1 Model Assumptions and Justification

A number of assumptions were made to simplify the grinding model and reduce the computation required. These assumptions are discussed in depth in Section 4.4. The assumptions made can be refined through further model development and testing. It may be possible to work around some assumptions, such as the 3D grind profile, using experimental data. This could greatly reduce the required CPU load while also improving the reliability of the model. A number of challenges remain to be overcome, such as tool wear, varying degrees of concrete floor hardness, and grinding head localisation.

6.4 Model Validation

6.4.1 Continuous Wavelet Transform Analysis

The Continuous Wavelet Transform (CWT) provides an analysis method for identifying deviations from flatness across a floor. This gives an improved analysis compared with a simple correlation of two surfaces, as it does not require the surfaces to be perfectly aligned. Due to the CWT, the analysis is carried out across a continuous spectrum of wavelengths and thus provides a good analysis of floor deviations. Analysing the CWT result on each floor surface can give better insight into how the grinding simulation performed in comparison with the experimental results.

It can be seen that the grinding simulation performed comparably in some areas of the floor, but also failed to capture some aspects observed in the experimental results. From visual analysis of the pre-grind floor (Figure 6-6) and the post-simulation floor (Figure 6-8), it is clear that floor flatness has been improved. In particular, some significant areas of deviation from flatness have resulted in increased wavelength deviations; this indicates that the high or low area in the floor has 'spread out' and thus the floor has become flatter. In the top middle-left of the pre-grind floor, there is significant deviation from flatness at a number of wavelengths, producing a long and thin peaked deviation. In the post-simulation floor, this same area has been reduced to a single, wider deviation, and can be said to have been smoothed. Similarly, in the bottom middle-right of the pre-grind floor there is a region of significant deviation composed of various wavelengths and two main peaks. In the post-simulation floor, this area is reduced to two main wavelengths of deviation, and the two main peaks identified are now merged into a single smoother deviation from flatness. This suggests that the two peaks in the floor are smoothed and the floor is once again moving towards flatness.

Comparing the post-grind floor (Figure 6-7) with the post-simulation floor (Figure 6-8) highlights a number of interesting points that give insight into the model's performance in capturing the floor grinding process. The bottom middle-right region of the post-grind floor illustrates some deviation from flatness and is comparable to the post-simulation floor. The post-grind floor has one of the two previously discussed high peaks remaining, whereas the simulation has removed this high point. Interestingly, to the right of this area the simulation does not present a deviation from flatness, while the post-grind floor shows a low region. This region could have been ground more than expected due to a number of unknown factors: the high regions around this area could have resulted in a small amount of scalloping resulting in increased grinding at that point; the operator could have unknowingly performed additional grinding in that area; and the floor could have some unknown variations in hardness resulting in a variation in material removal.

In addition, similar deviations are detected in the top middle-left region. In the pre-grind floor, the deviations are long and relatively thin, suggesting significant peaks in the floor that need to be smoothed. The post-grind floor has wider-wavelength deviations suggesting that the peak has been smoothed during the grinding process. This is reflected in the post-simulation floor.

The simulation did not perform well in a few areas. In the bottom left corner area of the floor, there is a detected high region of significant size. This is seen in both the pre-grind and post-simulation floors. However, in the post-grind floor, this area has been reduced and is not detected as a deviation from flatness at all. This difference in performance could be due to a number of factors such as floor hardness variations and operator path variations. In addition, the performance difference could be due to the end conditions of the model, and the fact that the model does not continue past this high region. In the experiment, the grinding machine moves beyond this defined area and this could result in additional material removal in this detected high region. This could be investigated further by extending the size of the captured floor area and ensuring similar machine paths are followed.

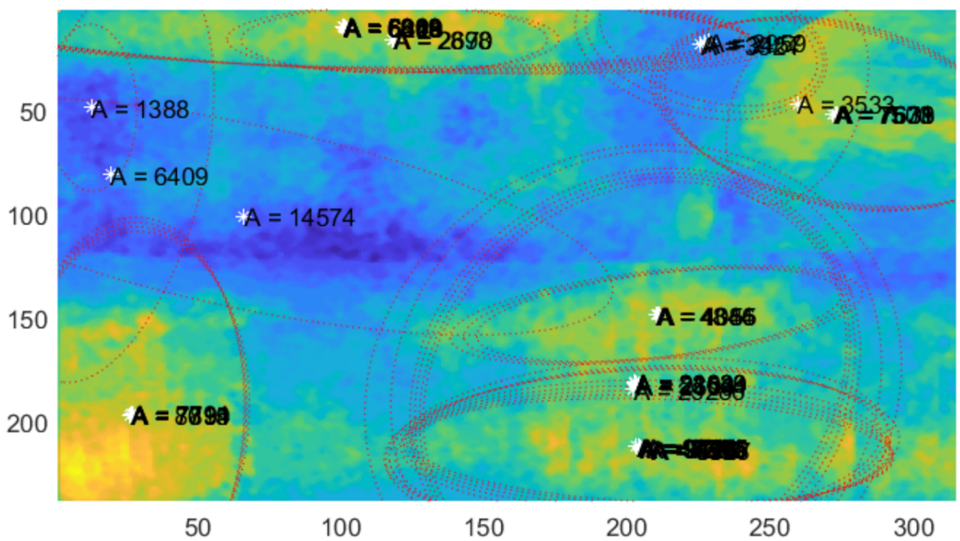


Figure 6-6: Deviations from flatness, pre-grind floor

6.4.2 Floor Capture Performance

The floor capturing process was able to capture the general profile of the floor; however, due to the depth resolution of the camera and unreliable global z correction, accurate analysis of high and low points is not possible. The system was able to detect local high and low areas, and general trends in the floor. However, the system is not successful in accurately capturing small (1-5 mm) deviations across the floor. In addition, the accuracy of the estimated height of the detected high and low areas is not known, which could introduce errors in the modelling. Accurate deviation estimation in addition to accurate global z height correction could provide useful information for the grinding process and requires further research to accurately capture.

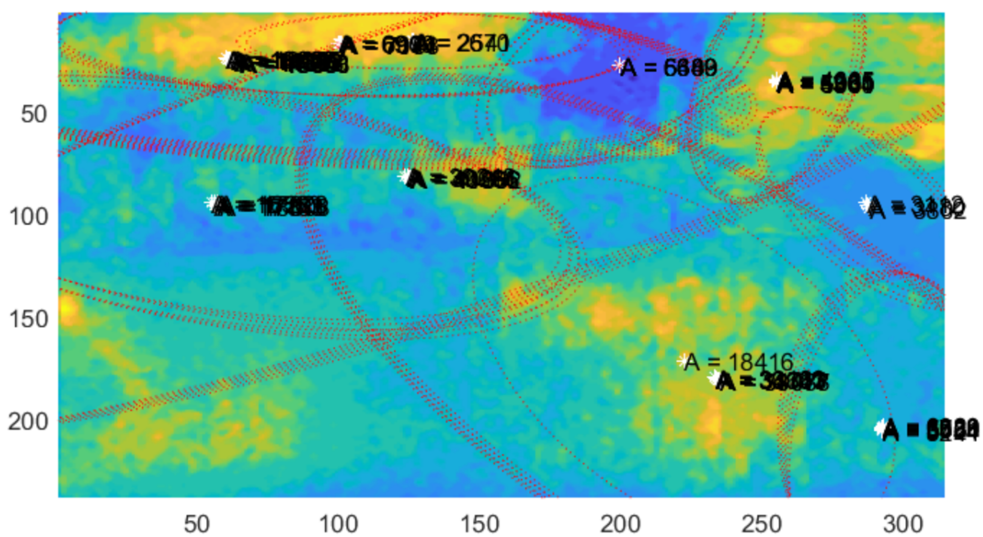


Figure 6-7: Deviations from flatness, post-grind floor

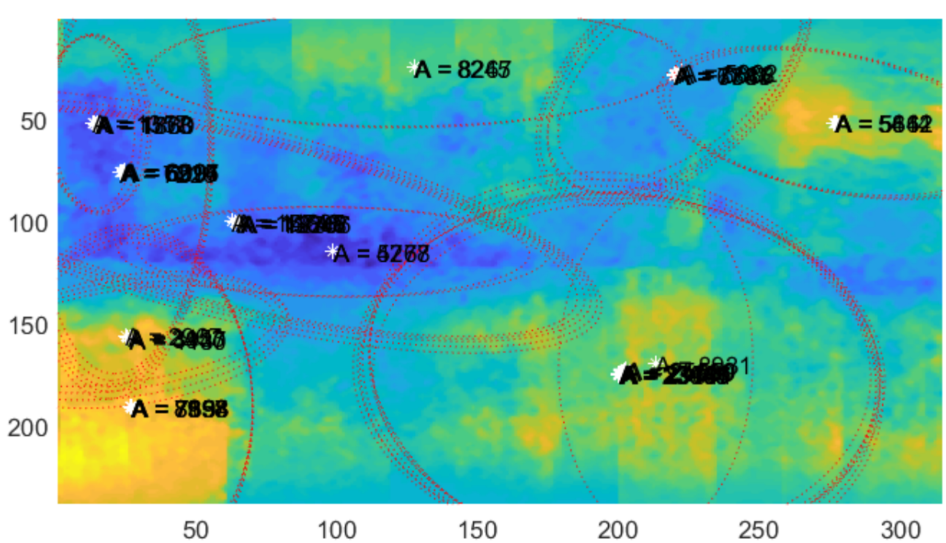


Figure 6-8: Deviations from flatness, simulated grind floor

6.4.3 Model Performance

The simplified grinding model was able to produce comparable results to experimental data under some conditions. Grinding model parameters such as cut rate, grind profile, and grind control have significant impacts on the result of the model. These must therefore be analysed further to identify the correct parameters to be used in an accurate model. This can be achieved through conducting further experiments following the same methodology and comparing results. Further, with more data, an optimisation technique such as a genetic algorithm could be used to find optimal process parameters for the grinding model and could then be applied to many situations. The model end condition assumptions could have introduced errors into the model, and these could be reduced by capturing larger floor areas for comparison. A key assumption that needs to be quantified is the material rate of removal from one pass of the grinding head. This value is currently an estimation only and this could lead to errors.

6.4.4 Sources of Error

The performance of the model compared with the experimental grind results was average, as shown by the contour plots and correlation of the surfaces. Visually, the floor appears to be flattened in that the high peaks are spread out. However, the correlation of the output experimental floor compared with the modelled post-grind floor was low. In addition, some aspects of the floor appear to fit hypotheses better in that the upper half of the floor correlates well, and visually looks to be improved. The lower half of the floor does correlate well; it could have suffered from measurement drift or could be affected by a number of unknown experimental parameters. This is likely attributable to a number of sources of error that can compound to result in a substantial difference.

Floor Capture Error

The floor capture process has been tested in a number of scenarios and produces respectable results; however, the accuracy of the heights measured has not been validated and so remains a source of error. The scanning methodology relies on the fusion of IMU, Odometry, and AMCL information for the 6 DOF pose estimation. The IMU can introduce errors and can be subject to drift resulting in inaccurate z height and floor angle estimation. This can have a significant impact on the captured floor profile, and thus will need to be overcome in future work. This could be achieved by providing additional sensors for z height estimation, or by improving IMU reliability. There was a noticeable error in z height estimation caused by the IMU in the post-grind test, and this is reflected in the surface as a significant low point. Reviewing the IMU data, this error could be caused by a bump whilst performing the second turn of the path, or arise from IMU drift. Interestingly, the error corrected itself after the third turn and the estimated z height was then consistent with the rest of the floor. Improvement could be achieved by developing a more robust method of estimating the z height, including integrating the IMU data and using additional sensors.

In addition, the process of stitching the point cloud information into an Octomap grid is not robust in that the stitching process is dependent on CPU load and the arrival of the point cloud data. This can introduce small errors between tests in the final surface generated. Any small positional errors in the surface can result in poor correlation between floor surfaces because the correlation only compares values at each position, rather than the features of the floor relative to one another. This results in poor correlation results when visually the floors could be similar, and alternative methods of analysing the floor profile were therefore required. This situation could be improved through a more robust method of stitching point clouds together.

Experimental Procedure Errors

There were a number of steps in the experimental procedure that could result in errors, and the aim is to overcome these in future work. The model does not consider the condition of the grinding head, which can vary during experimentation resulting in varying degrees of material removal along the floor. The condition of the grinding head can be affected by diamond tool wear, grinding head and disc RPM, and machine velocity. In addition, the concrete grinding machine did not follow a strict path; the path was determined by the operator. As a strict path was not followed and thus the coordinates of the grinding machine could not be accurately tracked, the location of the grinding machine throughout the experiment is unknown. The operator could have unknowingly ground parts of the floor more than others during the process.

Throughout the simulation, the grinding model relies on an accurate position of the grinding machine for estimation of the amount of cutting at each time step. As the position of the grinding head throughout the experiment is not known, this aspect of the model could contribute to some additional error and differences in the resulting floor. In addition, the grinding model does not consider floor hardness, and this can be particularly important factor in material removal. A change from soft to hard can result in greatly reduced material removal. Finally, the grinding model does not take into account the forces acting on the grinding head as a result of the grinding action itself. These forces can cause the grinding head to lift from the floor and reduce the rate of material removal. The forces are dependent on a number of unknown variables, such as diamond edge protrusion. Thus, the resulting change in grinding head orientation is difficult to accurately estimate from basic parameters.

6.4.5 Model Tuning

The simplified grinding model has many parameters that can be tuned to alter the output of the simulation. Some of these, such as depth of cut and diamond exposure level, can have a significant effect on the output of the simulation. However, due to the high number of variables and unknown interactions between variables, future work could include implementing an optimisation method on the grinding model to better match experimental data. Once experimental data can be successfully predicted using the grinding model, the model could be extended to improve the grinding process and identify parameters to be used from the floor profile. For the model implemented, assumptions were made for a number of the grinding variables following discussion with industry experts and analysis of the grinding action. Tuning of the model was considered out of scope for this project, but could provide a method of improving the model's performance. Any tuning would have to be verified against additional experimental data to ensure that the model does not simply fit the captured data.

6.4.6 Model Limitations

A particular limitation of the model validation is that the experimental data being compared with the model data can only be assumed to be accurate. The floor capture process can later be validated against a stationary 3D Terrestrial Laser Scanner; however, this was out of scope for the current research project due to resource limitations. Validation of the data capture process will be an important future task in continued research in this area. Such validation can be achieved through a similar mapping and comparison process as outlined in the experimental methodologies. Use of the robot system entails the correlation of data points to a global coordinate system with respect to the map, and so any captured profiles can successfully be compared.

The simplified grinding model assumes the position of the grinding head to be accurate as it moves along the floor surface; however, without additional sensors this information is hard to quantify. The model requires accurate localisation of the grinding head, which was not achieved in these experiments.

6.5 System Improvements

A number of areas of further research can improve the overall system.

6.5.1 Point Cloud Registration

A point cloud registration method could be applied to the system to aid the 3D alignment of scans. IMU data alone can result in drift and inaccurate results over time, particularly when turning. Point cloud registration could help reduce the effects of any IMU drift and provides a means for loop closure, greatly increasing the accuracy of 2D maps and hence, potentially increasing the accuracy of the 3D floor map creation. This could also be used to estimate any changes in the global z height.

6.5.2 Global z Height

Improved estimation of the global z height can not only improve the performance of the floor capture system, but can also give insight into the amount of material removed at each point along the floor. In the current system, this cannot be measured and comparisons can only be made of one floor with another. Currently, the z datum point is assigned by the robot at the beginning of each test, and the z height of the robot with respect to its surroundings or between tests is not considered. If the floor is lower from one test to another (as it is in the process of grinding), then the robot resets its datum point when it begins the next test, so no estimation of material removal is possible. Such estimation would provide useful information for grinding performance and model improvement.

6.5.3 Grinding Model Parameters

The grinding model can be improved by implementing more parameters in the grinding process. In addition, an important aspect is the exact location of the grinding head throughout the grinding process, as this results in significantly different simulation output results. While grinding process parameters such as diamond tool protrusion and estimated cutting rate can be tuned to improve the model, this was considered out of scope for this project.

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

This project has successfully identified further areas of research required for the development of a controlled automatic grinding process. For accurate control, the grinding process can be broken into two main areas: the capture of floor surface profile data, and using this data to improve control of the machine to achieve better performance. Capture of the floor surface profile was demonstrated using a robotic research platform with two sensors: a horizontal 2D laser scanner for SLAM, and a second, swappable, sensor to capture the floor data. The experimental results showed that the system was able to capture some features of the floor, but its full capability is yet to be verified. The RGB-D camera performed better than the laser scanner, providing greater insight into the high and low areas of the floor, which were confirmed using a straight edge. The developed system utilises cheap, accessible sensors to create a 3D floor surface map of the environment that can be used as prior knowledge. This provides advantage in a number of areas, notably polishing, grinding, cleaning, navigation, inspection, and terrain traversability.

A basic grinding model was devised to calculate material removal along the floor surface, taking into account the uneven floor geometry and how this affects performance. The model was validated against experimental data and was found to have reasonable success in predicting the post-grind floor. Some differences were observed in the simulated data compared with the experimental data and data, which arose from a number of uncharacterised variables. The model can therefore be improved by incorporating more aspects of the grinding process; in particular an accurately known position of the grinding head, tool wear, variations in floor hardness, and the condition of the grinding head. In addition, further tuning of the current model parameters could increase the model's ability to simulate the experimental results.

A number of areas of further research have been identified. Validation of the accuracy of the scanning process remains a requirement for further development. In addition, robust methods of estimating the z height for 3D extrapolation can be explored. The grinding model can be developed further and some limitations of the grinding model can be overcome. Key areas of the grinding process can be incorporated into the grinding model, and this is expected to improve the performance of the model. Overall, this project was successful and has added value to the research field in identifying a plausible, cheap, and fast method of capturing a floor profile and then using this information to improve floor condition through the grinding process. This has the potential to improve both productivity and floor flatness in the floor grinding process, as well as other processes such as cleaning and inspection tasks.

7.2 Recommendations

The following recommendations are made based on the results and analysis of the presented system and model.

- The floor profile capture system using the RGB-D camera should be further developed to improve robustness and mapping methods.
- Limitations of the RGB-D camera such as exposure calibration and environmental changes, such as lighting and surface reflectivity, should be addressed through further software and hardware developments.
- Alternative methods of estimating the global z height of the robotic system should be explored.
- The floor capture system should be validated using a 3D Terrestrial Laser Scanner or equivalent stationary, high-performance scanning system.
- Further floor capture experiments of floor pre- and post-grind should be performed and analysed. Sensor calibration prior to each experiment and maintaining consistent methodology, particularly location and speed, when capturing data are key considerations for further testing.
- The grinding model should be tuned further to better fit experiment data.
- The grinding model should be developed further to overcome some current limitations and assumptions.
- The grinding model should be validated against more experimental data for a variety of situations and floors.

References

[1] Omron Adept. Pioneer p3-at. "http://www.mobilerobots.com/ResearchRobots/P3AT.aspxx", 2016.

[2] Omron Adept. Pioneer p3-dx. "http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx", 2016.

[3] SICK AG. Lms200/211/221/291 laser measurement systems. "http://sicktoolbox.sourceforge.net/docs/sick-lms-technical-description.pdf", 2006.

[4] World Health Organisation: International agency for Research on Cancer. Arsenic, metals, fibres, and dusts: a review of human carcinogens, 2012.

[5] Mahdi S Alajmi, Fawzan S Alfares, and Mohamed S Alfares. Selection of optimal conditions in the surface grinding process using the quantum based optimisation method. *Journal of Intelligent Manufacturing*, pages 1–13, 2017.

[6] Ahmad Alhasan. Terrestrial laser scanning roughness assessments for infrastructure. In *Conference on Autonomous and Robotic Construction of Infrastructure*, page 21, 2015.

[7] Ahmad Alhasan, David J White, and Kris De Brabanterb. Continuous wavelet analysis of pavement profiles. *Automation in Construction*, 63:134–143, 2016.

[8] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[9] Cosmin Banica, Sanda Victorinne Paturca, Sorin Dan Grigorescu, and Ana Maria Stefan. Data acquisition and image processing system for surface inspection. In *Advanced Topics in Electrical Engineering (ATEE), 2017 10th International Symposium on*, pages 28–33. IEEE, 2017.

[10] Filippo Basso, Matteo Munaro, Stefano Michieletto, Enrico Pagello, and Emanuele Menegatti. Fast and robust multi-people tracking from rgb-d data for a mobile robot. In *Intelligent Autonomous Systems 12*, pages 265–276. Springer, 2013.

[11] Hadjira Belaidi, Hamid Bentarzi, Abderrahmane Belaidi, and Abdelfetah Hentout. Terrain traversability and optimal path planning in 3d uneven environment for an autonomous mobile robot. *Arabian Journal for Science & Engineering (Springer Science & Business Media BV)*, 39(11), 2014.

[12] Wolfgang Boehler, M Bordas Vicent, Andreas Marbs, et al. Investigating laser scanner accuracy. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(Part 5):696–701, 2003.

[13] Johann Borenstein and Liqiang Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on robotics and automation*, 12(6):869–880, 1996.

[14] Frédéric Bosché and Baptiste Biotteau. Terrestrial laser scanning and continuous wavelet transform for controlling surface flatness in construction–a first investigation. *Advanced Engineering Informatics*, 29(3):591–601, 2015.

[15] Frédéric Bosché and Emeline Guenet. Automating surface flatness control using terrestrial laser scanning and building information models. *Automation in construction*, 44:212–226, 2014.

[16] Michael Bosse, Paul Newman, John Leonard, and Seth Teller. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *The International Journal of Robotics Research*, 23(12):1113–1139, 2004.

[17] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[18] British Standards Institution (BSI). Bs 8204 - screeds, bases and in situ flooring, 2009.

[19] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[20] Huang-Cheng Chang and J-J Junz Wang. A stochastic grinding force model considering random grit distribution. *International Journal of Machine Tools and Manufacture*, 48(12-13):1335–1344, 2008.

[21] Mengxiao Chen, Shaowu Yang, Xiaodong Yi, and Dan Wu. Real-time 3d mapping using a 2d laser scanner and imu-aided visual slam. In *Real-time Computing and Robotics (RCAR), 2017 IEEE International Conference on*, pages 297–302. IEEE, 2017.

[22] Jacky CK Chow, Derek D Lichti, Jeroen D Hol, Giovanni Bellusci, and Henk Luinge. Imu and multiple rgb-d camera fusion for assisting indoor stop-and-go 3d terrestrial laser scanning. *Robotics*, 3(3):247–280, 2014.

[23] David M Cole and Paul M Newman. Using laser range data for 3d slam in outdoor environments. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1556–1563. IEEE, 2006.

[24] Walid Darwish, Shenjun Tang, Wenbin Li, and Wu Chen. A new calibration method for commercial rgb-d sensors. *Sensors*, 17(6):1204, 2017.

[25] Kristin M de Payrebrune and Matthias Kröger. Reduced models of grinding wheel topography and material removal to simulate dynamical aspects in grinding. *The International Journal of Advanced Manufacturing Technology*, 88(1-4):33–43, 2017.

[26] MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241, 2001.

[27] David Droeschel, Max Schwarz, and Sven Behnke. Continuous mapping and localization for autonomous navigation in rough terrain using a 3d laser scanner. *Robotics and Autonomous Systems*, 88:104–115, 2017.

[28] US Patnaik Durgumahanti, Vijayender Singh, and P Venkateswara Rao. A new model for grinding force prediction and analysis. *International Journal of Machine Tools and Manufacture*, 50(3):231–240, 2010.

[29] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[30] Felix Endres, Jurgen Hess, Jurgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an rgb-d camera. *IEEE Transactions on Robotics*, 30(1):177–187, 2014.

[31] Next Engine. Nextengine 3d scanner tech specs. "`http://www.nextengine.com/assets/pdf/scanner-techspecs-uhd.pdf`", 2018.

[32] Open Source Robotics Foundation. Turtlebot2. "`http://www.turtlebot.com/turtlebot2/`", 2017.

[33] Feng Gao. Interferometry for online/in-process surface inspection. In *Optical Interferometry*. InTech, 2017.

[34] B. Gerkey and T. Pratkanis. Ros map server package. "`http://wiki.ros.org/map_server`", 2016.

[35] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34–46, 2007.

[36] Giorgio Grisettiyz, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2432–2437. IEEE, 2005.

[37] Mirosław Grzelka, Radomir Majchrowski, and Łukasz Sadowski. Investigations of concrete surface roughness by means of 3d scanner. *Proceedings of Electrotechnical Institute*, 16, 2011.

[38] Martin Gunther, Thomas Wiemann, Sven Albrecht, and Joachim Hertzberg. Building semantic object maps from sparse and noisy 3d data. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2228–2233. IEEE, 2013.

[39] Jianliang Guo, Chenggang Liu, Jun Chi, and Lianqing Chen. Comprehensive model for predicting surface roughness of ground workpieces considering the plowing effect. *Machining Science and Technology*, pages 1–17, 2017.

[40] Dirk Hahnel, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 206–211. IEEE, 2003.

[41] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *Experimental robotics*, pages 477–491. Springer, 2014.

[42] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[43] Husqvarna. Floor grinders and polishing; husqvarna pg 820 rc. `"https://www.husqvarnacp.com/us/machines/floor-grinders/pg-820-rc/967302902/"`, 2017.

[44] FARO Technologies Inc. Faro focus. `"https://www.faro.com/products/construction-bim-cim/faro-focus/"`, 2018.

[45] Intel. Intel realsense depth camera d435. `"https://click.intel.com/intelr-realsensetm-depth-camera-d435.html"`, 2018.

[46] ASTM International. Astm e1486-14, standard test method for determining floor tolerances using waviness, wheel path and levelness criteria, 2014.

[47] Philipp Krüsi, Paul Furgale, Michael Bosse, and Roland Siegwart. Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments. *Journal of Field Robotics*, 34(5):940–984, 2017.

[48] Pil-Ho Lee, Jung Soo Nam, Chengjun Li, and Sang Won Lee. An experimental study on micro-grinding process with nanofluid minimum quantity lubrication (mql). *International Journal of Precision Engineering and Manufacturing*, 13(3):331–338, 2012.

[49] Hao Nan Li, Tian Biao Yu, Zi Xuan Wang, Li Da Zhu, and Wan Shan Wang. Detailed modeling of cutting forces in grinding process considering variable stages of grain-workpiece micro interactions. *International Journal of Mechanical Sciences*, 126:319–339, 2017.

[50] Haonan Li, Tianbiao Yu, Lida Zhu, and Wanshan Wang. Analysis of loads on grinding wheel binder in grinding process: insights from discontinuum-hypothesis-based grinding simulation. *Int J Adv Manuf Technol*, 78:1943–1960, 2015.

[51] Yi Lin, Juha Hyyppä, and Antero Kukko. Stop-and-go mode: Sensor manipulation as essential as sensor development in terrestrial laser scanning. *Sensors*, 13(7):8140–8154, 2013.

[52] Yinchen Ma, Jianguo Yang, Beizhi Li, and Jun Lu. An analytical model of grinding force based on time-varying dynamic behavior. *The International Journal of Advanced Manufacturing Technology*, 89:2883–2891, 2017.

[53] Anthony Mandow, Jorge L Martinez, Jesús Morales, José L Blanco, Alfonso Garcia-Cerezo, and Javier Gonzalez. Experimental kinematics for wheeled skid-steer mobile robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1222–1227. IEEE, 2007.

[54] Andrew McDonald, Al-Mokhtar O Mohamed, Andrew Warkentin, and Robert J Bauer. Kinematic simulation of the uncut chip thickness and surface finish using a reduced set of 3d grinding wheel measurements. *Precision Engineering*, 49:169–178, 2017.

[55] Sungwoo Moon, Byongsoo Yang, Jeonghwan Kim, and Jongwon Seo. Effectiveness of remote control for a concrete surface grinding machine. *Automation in Construction*, 19(6):734–741, 2010.

[56] Kenneth Olofsson, Johan Holmgren, and Håkan Olsson. Tree stem and height measurements using terrestrial laser scanning and the ransac algorithm. *Remote sensing*, 6(5):4323–4344, 2014.

[57] Sebastian Pütz, Thomas Wiemann, Jochen Sprickerhof, and Joachim Hertzberg. 3d navigation mesh generation for path planning in uneven terrain. *IFAC-PapersOnLine*, 49(15):212–217, 2016.

[58] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[59] Nils Raabe, Christian Rautert, Manuel Ferreira, and Claus Weihs. Geometrical process modeling and simulation of concrete machining based on delaunay tessellations. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 2, pages 19–21. WCECS, San Francisco, USA, 2011.

[60] Matthias Rasim, Patrick Mattfeld, and Fritz Klocke. Analysis of the grain shape influence on the chip formation in grinding. *Journal of Materials Processing Technology*, 226:60–68, 2015.

[61] Giulio Reina, David Johnson, and James Underwood. Radar sensing for intelligent vehicles in urban environments. *Sensors*, 15(6):14661–14678, 2015.

[62] Fabio Remondino and Sabry El-Hakim. Image-based 3d modelling: a review. *The Photogrammetric Record*, 21(115):269–291, 2006.

[63] RobotShop. Hokuyo urg scanning laser rangefinder. "`https://www.robotshop.com/en/hokuyo-urg-04lx-ug01-scanning-laser-rangefinder.html`", 2017.

[64] J Seo, S Moon, S Kim, W Lee, and J Lim. Prototyping and automating a concrete surface grinding machine for improving infrastructure conditions. In *Proc. Int. Symp. Automation Robotics in Construction*, pages 660–664, 2006.

[65] TruePoint 3D Laser Scanning Services. 3d laser scanning. "`https://www.truepointscanning.com/faq/`", 2017.

[66] Dinesh Setti, Sudarsan Ghosh, and P Venkateswara Rao. A method for prediction of active grits count in surface grinding. *Wear*, 382:71–77, 2017.

[67] J. Steinbaeck, C. Steger, G. Holweg, and N. Druml. Next generation radar sensors in automotive sensor fusion systems. In *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6, Oct 2017.

[68] Todor Stoyanov, Martin Magnusson, Henrik Andreasson, and Achim J Lilienthal. Path planning in 3d environments using the normal distributions transform. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3263–3268. IEEE, 2010.

[69] Jinyuan Tang, Jin Du, and Yongping Chen. Modeling and experimental study of grinding forces in surface grinding. *Journal of materials processing technology*, 209(6):2847–2854, 2009.

[70] Pingbo Tang, Daniel Huber, and Burcu Akinci. Characterization of laser scanners and algorithms for detecting flatness defects on concrete surfaces. *Journal of Computing in Civil Engineering*, 25(1):31–42, 2010.

[71] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1-2):99–141, 2001.

[72] E Valero and F Bosché. Automatic surface flatness control using terrestrial laser scanning data and the 2d continuous wavelet transform. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 33, page 1. Vilnius Gediminas Technical University, Department of Construction Economics & Property, 2016.

[73] Huanhuan Wang, Yi Lin, Zeliang Wang, Yunjun Yao, Yuhu Zhang, and Ling Wu. Validation of a low-cost 2d laser scanner in development of a more-affordable mobile terrestrial proximal sensing system for 3d plant structure phenotyping in indoor environment. *Computers and Electronics in Agriculture*, 140:180–189, 2017.

[74] Sheng Wang, Changhe Li, Xiaowei Zhang, Dongzhou Jia, Dongtun Zhang, and Qiang Zhang. Modeling and simulation of the single grain grinding process of the nano-particle jet flow of minimal quantity lubrication. *Open Materials Science Journal*, 8:55–62, 2014.

[75] Claus Weihs, Nils Raabe, Manuel Ferreira, and Christian Rautert. Statistical process modelling for machining of inhomogeneous mineral subsoil. In *German-Japanese interchange of data analysis results*, pages 253–263. Springer, 2014.

[76] Chenglu Wen, Ling Qin, Qingyuan Zhu, Cheng Wang, and Jonathan Jonathan Li. Three-dimensional indoor mobile mapping with fusion of two-dimensional laser scanner and rgb-d camera data. *IEEE Geoscience and Remote Sensing Letters*, 11(4):843–847, 2014.

[77] Scott Wilson, Johan Potgieter, and Khalid Arif. Floor surface mapping using mobile robot and 2d laser scanner. In *Mechatronics and Machine Vision in Practice (M2VIP), 2017 24th International Conference on*, pages 1–6. IEEE, 2017.

[78] Worksafe. Silica dust in the workplace. "`https://worksafe.govt.nz/topic-and-industry/dust-and-fumes/dust/silica-dust-in-the-workplace/`", 2017.

[79] XtionLivePro. Primesense carmine 1.09. "`http://xtionprolive.com/primesense-carmine-1.09`", 2017.

[80] Yanbin Zhang, Changhe Li, Heju Ji, Xiaohui Yang, Min Yang, Dongzhou Jia, Xianpeng Zhang, Runze Li, and Jun Wang. Analysis of grinding mechanics and improved predictive force model based on material-removal and plastic-stacking mechanisms. *International Journal of Machine Tools and Manufacture*, 2017.

[81] Robert Zlot and Michael Bosse. Efficient large-scale 3d mobile mapping and surface reconstruction of an underground mine. In *Field and service robotics*, pages 479–493. Springer, 2014.

Appendix A

Floor Surface Mapping using Mobile Robot and 2D Laser Scanner