ROBOTIC SANDING OF FREE-FORM SURFACES WITH TRAJECTORY PLANNING FOR UNIFORM SURFACE COVERAGE

A Thesis

by

DANIEL JOHN JAEGER

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Prabhakar Pagilla
Committee Members,	Darbha Swaroop
	Suman Chakravorty
Head of Department,	Andreas Polycarpou

December 2020

Major Subject: Mechanical Engineering

ABSTRACT

This thesis provides several improvements to producing a uniform surface finish on free-form surfaces in a robotic sanding procedure. The three improvements which this thesis discusses involve surface reconstruction of the workpiece to be sanded, uniform trajectory planning, and modeling physical factors which may affect the material removal.

While surface reconstruction is an ongoing area of research, many studies do not consider whether the reconstructed surface is smooth enough for trajectory planning. In this thesis, we address this problem by developing a method for reconstructing a free-form surface which is sufficient for planning trajectories that can be used for surface finishing operations.

In regards to trajectory planning, this thesis presents an improvement on previous tool path planners with the objective of uniform coverage on free-form surfaces with the least amount of sanding overlap. Whereas previous works assume that the sanding pad remains in full contact with the surface, this assumption is not always valid when polishing surfaces with varying curvature. This thesis relaxes this assumption, allowing the contact area between the sanding pad and surface to change throughout the motion. This thesis presents numerical simulations as well as physical experiments, demonstrating that this trajectory planner is able to provide uniform coverage with changing contact area.

Finally, this thesis models external factors (e.g. sand paper grit) which may affect the amount of material removed from the surface. After conducting several empirical trials, a model is constructed which predicts the amount of material removed for a given set of parameters.

DEDICATION

To my parents and sister

ACKNOWLEDGMENTS

Before beginning this thesis, I wish to express my gratitude to the many people who supported me during my graduate program.

To my adviser Dr. Pagilla, thank you for your continual support and guidance during my time of study. I am not able to thank you enough for all of the time you dedicated to help me. I will always be indebted to you for the wisdom you have shown me over the past few years.

To all the members of my lab: Yalun, Riya, Jie, Zongyao, Pranav, Orlando, Juan, and Angel, it has been a pleasure working with you guys on a day-to-day basis. I want to especially thank Yalun, Riya, and Jie for all of the valuable advice you gave me. It will certainly serve me well for years to come.

To Pastor Bard, I will never be able to thank you enough for the spiritual wisdom you have imparted on me during this phase of my life. Never before have I met someone who expresses such zeal for the gospel of Jesus Christ. Thank you for truly being a good and faithful servant of God.

Finally, I wish to express my sincere gratitude to my parents and sister. Thank you Mom and Dad for supporting me in every way throughout the past few years and for being examples of how to live a godly life. Thank you Sarah for being so supportive of me and for being such a great sister. No words can describe how much I love you guys.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This thesis was supervised by a committee consisting of Professor Prabhakar Pagilla (adviser) and Professor Swaroop Darbha of the Department of Mechanical Engineering and Professor Suman Chakravorty of the Department of Aerospace Engineering.

The method of control discussed in this paper was first presented in the following paper: Wen, Yalun, Jie Hu, and Prabhakar R. Pagilla. "A Novel Robotic System for Finishing of Freeform Surfaces." 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019. All other work in this thesis was performed by the student independently under the supervision of Dr. Prabhakar Pagilla. The author is grateful for the resources and discussion provided by Dr. Pagilla which assisted in the completion of this thesis.

Funding Sources

Funding for a portion of the work reported in this thesis was provided by the Advanced Robotics for Manufacturing Institute Robotic Sanding and Finishing grant. Test articles used in the experimentation section were provided by Lockheed Martin, under the aforementioned grant.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Motivation

Since the development of the first robotic arm, articulated robots have had a significant impact on several manufacturing processes including pick-and-place, spot-welding, painting, and surface finishing. With the advent of high memory storage and processing power, articulated robotics has seen a resurgence within the past decade. The number of installed articulated robotics in the world nearly tripled from 178,000 in 2013 to 422,000 in 2018 [1].

Although robots have widely been used for tasks which consist of repeated motion, the use of robots for non-repetitive tasks is still an area of research. One category of such problems is surface finishing problems. Surface finishing describes a class of processes which alter the surface properties of an object. Some examples of surface finishing processes include grinding, polishing, sanding, and deburring. In the majority of these processes, an object's surface is traversed using a rotary tool which is used to smooth natural asperities or defects that are a result of a shaping or formation process.

There is a strong motivation for using articulated robots to perform surface finishing processes. In many cases, surface finishing performed by a human requires several hours of manual labor in non-ideal environments. In the case of sanding, some processes require several hours of manual sanding. The prolonged manual labor often leads to other issues such as poor ergonomics for the operator, poor air quality, and numbness in hands and arms due to vibration from the tool. Another issue arising from manual labor is the inconsistency in accuracy. The accuracy of the performed task is primarily dependent on the skill and experience of the operator. Even when the operator is highly skilled, there is bound to be inconsistency in some areas of the surface finishing due to human imperfection. In addition to removing the human from a hazardous environment, an articulated robot is able to provide a more consistent surface finishing.

There are several reasons why surface finishing processes are significantly more difficult for

articulated robots than repetitive motion processes, such as pick-and-place. One reason why surface finishing may be more difficult is that the motion is non-repetitive. Because defects may appear in random locations on the surface, the motion of the robot end-effector may vary from one object to another, as the objects may vary in shape and contour. Another challenge with using an articulated robot for surface finishing processes is registering an object with respect to the robot. Even if a surface profile is known completely, the surface must be correctly related to the base frame of the articulated robot. A third challenge with robotic surface finishing is the need for both position control of the robot end-effector and force control to regulate the force which is applied to the surface.

There are still many open problems which have prevented fully automated surface finishing from becoming widespread. One obstacle is how to accurately reconstruct a surface and accurately register the surface with respect to the robot base frame. An accurate perception of an object is necessary in order for a quality surface finish. Another open problem is the path that the robot should take to complete a surface finishing task. Many surface finishing tasks require that the entire surface be completely covered. This thesis address the problem of finding the trajectory which uniformly covers the surface in a time-efficient manner.

1.2 Thesis Overview

The remainder of Chapter 1 provides a focused literature overview on relevant research related to robot surface finishing. In Chapter 2, the requirements and specification of the robotic surface finishing problem is explored more in depth. The next three chapters address key problems associated with robotic surface finishing in the order that they occur in a typical sanding procedure: In Chapter 3, the problem of surface reconstruction is addressed and a method for accurately reconstructing a surface using a similar model is introduced. In Chapter 4, the trajectory generation problem is discussed and an unsupervised learning procedure for generating a full-coverage trajectory is presented. In Chapter 5, a controller used for many robotic surface finishing processes is presented, along with its stability analysis. In Chapter 6, we present material models which are applicable to robotic sanding using rotary sanders. Chapter 7 presents the

experimental results using the methods discussed in the previous chapters. The thesis concludes by providing some concluding remarks and future work in Chapter 8.

1.3 Background Information

A complete robotic surface finishing process encompasses several areas of research. The main three areas which will be addressed in this thesis are surface reconstruction, trajectory generation, and motion/force controller development and implementation. This section provides a more in-depth overview of the issues in each of these areas as they apply to robotic surface finishing. In addition to describing the nuances of each problem, a literature review is provided to show the research that has been conducted thus-far to address these problems.

1.3.1 Surface Reconstruction

This subsection describes the problem of surface reconstruction in more detail, specifically how it applies to robotic surface finishing. After providing some background on the types of sensors used in surface reconstruction, an overview of the open problems associated with reconstructing surfaces for robotic surface finishing applications is provided.

1.3.1.1 General Definition

In the context of computer vision, surface reconstruction is the process by which the characteristics of a physical object are captured and represented numerically. In many robotic surface finishing applications, the exact surface contour is not known a priori, meaning that the surface must first be scanned using an optical sensor. The data collected from the sensor is often in the form of a collection of points in 3D space called a point cloud. The point cloud is used to construct a surface which can be used to generate a trajectory.

1.3.1.2 Optical Sensors

The three main types of sensors used for surface reconstruction are stereo-vision, time-of-flight, and structured light. These sensors vary widely in both underlying mechanism and each type has its own advantages.

One method used for surface reconstruction is stereo vision. Stereo vision uses two cameras separated by a known distance to view the same object. Algorithms are used to either match features in both of the images or estimate the disparity between images. The transformation between the two images is used to project the points into three dimensional space [2]. While stereo vision can be used for surface reconstruction, it is more often used for object position estimation and object recognition. Stereo cameras also require calibration of the frames of each lens as detailed in [3]. Examples of stereo vision cameras include the ZED 2K Stereo Camera made by Stereolabs and Intel's Realsense depth camera.

Time-of-flight cameras and sensors are another method of reconstructing three-dimensional surfaces. Time-of-flight sensors shine either a single or multiple beams of light on an object, which is reflected back to the sensor. The sensor measures the time the light takes to be reflected back, allowing the distance between the sensor and object to be calculated using the speed of light [4]. Time-of-flight distance estimation is used in both single-beam lasers and cameras. A few examples of time-of-flight sensors are the Kinect2 manufactured by Microsoft and most opto-electronic distance sensors.

Structured light is similar to stereo vision, except the camera is replaced by a projector. The projector projects a pattern of light (often stripes) onto a surface. Two cameras are used to capture the deformed stripes on the surface and compute the Euclidean distance of the surface from the sensor [5]. Structured light cameras are typically the most accurate and most expensive of the three categories. Examples of structured light sensors are the Kinect V1 manufactured by Microsoft and the FARO Cobalt sensor.

1.3.1.3 Open Problems in Surface Reconstruction

There are several open problems in the field of surface reconstruction. Here we will specifically focus on two problems which are especially important in the area of robotic surface finishing: Feature recognition and accuracy of the reconstructed surface.

The feature recognition problem describes knowing when to classify a deviation in a point cloud as an incorrect measurement or a real feature of the object. When there is no ground truth

object to use for comparison, the only source of information for making this distinction is the points surrounding the deviation. The trivial solution to solving the feature resolution problem is to increase the point cloud resolution so that smaller deviations can be detected. However, the trade-off of increasing point cloud resolution is often times an increase in cost.

The correctness problem is how correct the reconstructed surface is to the real surface. For many applications, the metric for correctness is the Euclidean residual between the a point cloud and the real surface. While positional correctness is important, the correctness of the reconstructed surface normals is equally important in robotic surface finishing applications. The problem of how to reconstruct a surface from noisy data such that the surface normals have very little noise is an open problem which will be addressed later in this thesis.

1.3.1.4 Review of Surface Reconstruction

The three formats most commonly used to represent reconstructed surfaces are polygonal mesh, CSG models, and parametric surfaces [6]. While each of these techniques has its own advantages, the most versatile of these for irregular or non-planar surfaces is parametric surfaces. Unlike the other two methods, the model which describes the surface is purely mathematical, making it the best candidate for generating custom trajectories.

There are two main methods of approach for reconstructing a surface from three dimensional data: Interpolation and approximation. Interpolation techniques reconstruct a surface using only the data itself, whereas approximation techniques use the data in conjunction with approximation algorithms. Approximation techniques are usually preferred because the point cloud data is typically noisy.

There has been much research into using approximation algorithms for surface reconstruction given noisy data. In [7], a least squares method was used to approximate a curve by minimizing the residual error between the points and approximation. Similar least squares approaches have been developed for splines as was performed in [6] [8].

1.3.2 Trajectory Planning

In this subsection, the trajectory planning problem is discussed as it applies to robotic surface finishing. After providing an overall definition of trajectory planning and discussing some open problems in this area, an overview of the research of trajectory planning for robotic finishing operations is provided.

1.3.2.1 General Definition

Trajectory planning refers to a path that a robot is to follow. Specifically, a trajectory is a path in space that is parametrized by time. Trajectories are used for mobile robots, robotic manipulators, and autonomous vehicles. In the case of robotic surface finishing, a trajectory is used to plan the motion of the robot end-effector throughout the finishing process.

1.3.2.2 Trajectory Planning in Robotic Surface Finishing

An accurate and collision-free trajectory is necessary for an articulated robot to successfully perform a finishing process. The most common form of a trajectory is a series of way points defined in Euclidean space. Each way point must contain the Euclidean position of the end-effector as well as the end-effector orientation, commonly represented in the form of a homogeneous transformation matrix.

An accurate trajectory in the context of robotic surface finishing is one in which the orientations and positions of each way point align with the workpiece surface. If the way points are too far from the workpiece surface, the robot end-effector may collide or fail to make contact with the surface. Likewise, if the normals of the way points do not align with the normals of the surface, the tool used for the surface finishing may make contact at an incorrect angle.

Another requirement of trajectories used for surface finishing is that they adequately cover the entire desired area. There are several classes of trajectories that meet this requirement. Three types of trajectory paths which have been developed are scanning paths, Lissajour paths, and Peano paths [9] [10]. These types of trajectories are shown in Figure 1.1 below. The most common of these is the standard scanning path, likely due to its simplicity.

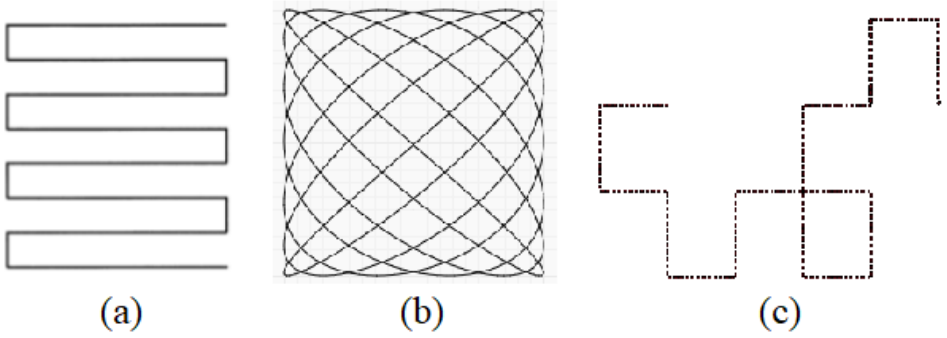


Figure 1.1: a) Scanning path b) Lissajous path c) Peano path

For simple three dimensional objects, the trajectory may be planned in two dimensions and then mapped to the three dimensional surface. However, this mapping becomes more complicated for free-form surfaces with high curvature. As explained in [11], a physically uniform polishing path in three-dimensional space requires that the trajectory account for the change in contact area of the tool with the surface. The subject of developing a trajectory which produces uniform coverage is discussed in much greater detail in Chapter 4.

1.3.2.3 Trajectory Planning for Free-Form Surfaces

An ongoing challenge in trajectory planning is planning a path which fully covers a free-form surface of arbitrary shape. An ideal trajectory should physically cover the entire surface and provide a quality surface finish in minimal execution time and without much overlap with the previously finished area.

One common approach to generating a trajectory which fully covers a surface is to divide the surface into equally sized cells and develop an algorithm for the robot to visit each cell [12] [13]. However, this method becomes significantly more challenging for free-from surfaces since the contact area between the tool and the surface is not constant over curved portions of the surface. In [14], a spiral trajectory was generated using an iterative algorithm which adds trajectory layers onto an initial "path seed". The algorithm employed in this paper ensures that the path coverage problem is met with minimal overlap between adjacent paths.

In [15], the surface is represented using a mesh model, which approximates the surface as several small planar patches. The lines between each plane are called "edges" and the junctions between edges are called "nodes." This paper using a graph-based approach to generate a fully-connected graph of all nodes and then traverse the nodes using a cost function which minimizes joint effort and time. While this method may indeed cover the surface completely, it is not suitable for surface finishing trajectories which should contain a minimal number of sharp turns.

1.3.2.4 Collision-free Trajectories

A common constraint found with trajectory generation problems is developing trajectories that are collision-free. A collision occurs when either a robot collides with itself or with an obstacle while the trajectory is being executed. For robotic finishing processes, collisions are likely to occur as the robotic manipulator is moving towards and away from the surface. Collision with stationary objects may be avoided by planning a trajectory which directs the robot around these objects. The problem of real-time collision avoidance is still an ongoing area of research. Without the use of cameras and other sensors, the location of all objects which may cause potential for collision must be known apriori so that a collision-free trajectory may be generated.

1.3.3 Force Control

The need for robots to interact with their surrounding environment has produced significant research in the area of controls, specifically controlling the motion of the end-effector in constrained environments. In this subsection we briefly provide a literature review of the types of force control used in robotic applications. We particularly focus on the difference on indirect and direct force control as they pertain to robotic surface finishing systems. This section is kept relatively brief, as the main contributions to this thesis pertain to trajectory planning.

1.3.3.1 Indirect Force Control

The first type of force control commonly employed in robotic systems is known as indirect force control. Indirect force control controls the force indirectly by changing the position of the end-effector relative to the workpiece surface [16]. This method of force control assumes that

there is a constant stiffness between the workpiece and end-effector. Some examples of indirect force control include impedance control [17] and compliance control [18].

1.3.3.2 Direct Force Control

Direct force control consists of two control loops, with the outer loop being force-controlled and the inner loop being position-controlled [19]. A load cell placed on the robot end-effector is typically used to close the outer force-control loop. An example of direct force control is what is known as hybrid position/force control [20]. In this method of control, the directions of force control and position control are set to be orthogonal to each other [16]. The main advantage of hybrid position/force control is that the force control and position control can be tuned separately without having a significant effect on each other.

A more in-depth derivation will is presented in Chapter 5 regarding hybrid position/force control, as we utilize this type of control scheme for executing the uniform-coverage trajectory.

1.4 Research Objective

There has already been much research conducted in the area of robotic surface finishing. Robots have already been successfully used to sand flat and curved surfaces and producing refined surface finishes. Despite the accomplishments that have already been made in this field, fully automated robotic surface finishing for arbitrary surfaces is far from being completely solved. In this thesis, we focus on a specific scenario when a curved workpiece is being sanded by a flat sanding tool. This problem has many applications to complex workpieces such as automobile bodies or aeronautical wings and rotary blades. A specific goal of this thesis is to plan a trajectory which will result in uniform sanding of such workpieces.

1.5 Contributions

In this thesis, we provide a simple method for reconstructing a curved helicopter blade section such that the reconstructed model can be easily used for generating accurate trajectories that are suitable for robotic sanding. Additionally, we develop a method for generating a trajectory that ensures uniform sanding of a curved surface utilizing non-uniform rational B-splines

(NURBS). The uniform trajectory planner ensures that the surface is uniformly sanded with minimal overlap, which equally-spaced trajectories may not be able to achieve. After performing some numerical simulations of the trajectory planner, we verify the planner's functionality with some physical experiments.

This chapter provides an overview of the main requirements which are required to perform a surface finishing procedure using an articulated robot. In Section 2.1 we first provide an overview of robotic sanding and provide the hardware and software that are used in the implementation of the robotic sanding experiments presented in this thesis. In Section 2.2 we consider relevant constraints which are imposed by the workpiece being sanded. Specifically, we discuss the relevance of the shape and material of the workpiece as well as the necessity of a workpiece model and registration procedure.

2.1 Robot and Tooling

This section first provides an overview of the operational requirements of robotic sanding from a high-level view. In the second portion, we provide a brief justification for the robot which was chosen for this procedure. Finally, this section provides the physical hardware and software that is used in this thesis to conduct the robotic sanding experiments.

2.1.1 Overview of Robotic Sanding

This subsection reviews the necessary requirements for a robotic sanding system to produce a quality surface finish. While the motion requirements explained in this section also pertain to sanding performed by a human, the requirements pertaining to the force applied to the end-effector are only achieved in a robotic sanding operation. Using these requirements, a robotic sanding procedure can mimic sanding performed by a human operator and even produce a more precise surface finish.

2.1.1.1 Motion of Sanding Tool

The motion of the sanding tool in a robotic sanding procedure should be designed such that the sanding tool successfully sands a given workpiece while possibly mimicking the motion of a skilled human operator. The first requirement of any sanding procedure is that the Euclidean

position of the sanding pad tool center point (TCP) must lie on the surface of the workpiece. If the TCP of the sanding tool is located below the workpiece surface, the robot will attempt to move the tool to this position, resulting in a collision. If the TCP is located above the surface, the sanding tool is no longer in contact with the surface.

As the sanding tool moves along the workpiece surface, it must change position while simultaneously changing orientation. The simultaneous change in position and orientation ensures that the sanding pad makes continuous contact with the workpiece surface and this is achieved through hybrid position/force control. The position of the sander is controlled in the tangential direction and the force is controlled normal to the surface.

Another requirement of the motion of the sanding tool is that the feed rate should be near constant. The feed rate refers to the tangential velocity of the sanding tool along the surface. In most sanding applications, it is desired that the same amount of material be removed from the entire surface. If the feed rate is slower, more material is removed from the surface at a given location. This relationship between feedrate and material removal is described in more detail in Chapter 6.

2.1.1.2 Force Applied to Surface

Another important component of sanding is the level and direction of force which is applied to the surface. In [21], sanding was performed with the sander tilted at a slight angle. Although this positioning is possible, the dynamics involved with finding the true force being applied becomes more complicated. Furthermore, because the contact area is no longer at the center of the pad, it is much more difficult to find the contact area analytically, which makes trajectory planning more difficult. For these reasons, the force is applied normal to the workpiece surface with the sanding tool positioned such that the pad is tangential to the surface.

2.1.1.3 Prior Work on Robotic Sanding

Robotic sanding has already been successfully demonstrated in other robotic sanding systems. A robotic sanding system was developed in [22] which was capable of sanding curved wooden

furniture. In [23], an integrated robotic polishing system was developed to polish small parts used in the aerospace industry. Although there already exists much research in this field, the full automation of sanding and polishing of arbitrary geometries is far from a completely solved problem.

2.1.2 Robot Selection

This subsection describes the type of robots capable of performing robotic sanding and provides a justification for the robot that was selected for performing the experiments later in this thesis.

2.1.2.1 Robot System Types

There are two primary types of robotic manipulators which are used for manipulating objects in three-dimensional space. The first type of robotic manipulator is referred to as a parallel robotic manipulator, due to the formation of the links of the robot. In a parallel robot, multiple links which are attached to a stationary platform and converge to a moving platform. The configuration of parallel robotic manipulators lead to high stiffness and poor load distribution [24].

The second type of robotic manipulator is called a serial robotic manipulator. Serial robotic manipulators are by far the most common type of robotic manipulator used in industry. The descriptor "serial" refers to how links are attached in a serial chain, beginning from a stationary base and ending with the end-effector. Serial robotic manipulators hold several advantages over parallel robotic manipulators. Two very important advantages are that serial robotic manipulators have significantly larger work spaces and the kinematic relationship between the joints and end-effector dynamics is much simpler compared to the kinematics of a parallel manipulator. For these reasons, a serial robotic manipulator would be a good choice for performing a robotic sanding operation and will be used in the experimental portion of this paper.

2.1.2.2 Robot Reachability

The trajectory of the robot is highly constrained by the reachability of the robot. The reachability of a robot is determined solely by the geometry and dimensions of the robot itself. One

characteristic of the robot which affects reachability is the number of joints. A robot with more joints is able to orient the end-effector into a greater number of orientations. This is because the degrees of freedom of a robot is equivalent to the sum of the dimensional connectivities of all the joints. Most articulated robots used in manufacturing processes are composed of six joints, which corresponds to six degrees of freedom. Each of these six degrees of freedom corresponds to one of the six kinematic motions of a rigid body, three rotational motions (roll, pitch, yaw) and three translational motions.

2.1.2.3 Joint Redundancy

A robot is said to be redundant if it the number of degrees of freedom of the joint space does not match the number of degrees of freedom of either the operational space or task space. Three types of redundancies are intrinsic redundancies, kinematic redundancies, and functional redundancies [25]. For example, if the robot is to operate in a plane, only two joints are needed to span two-dimensional space. Any additional joint would be considered redundant. Similarly, any robot which is used for rigid body motion which has greater than six joints would be considered redundant.

For robotic sanding, we only consider serial robots which are non-redundant (6-DOF). Robots with more than six joints, such as the KUKA iiwa, are not considered for robotic sanding, as these robots often have a smaller workspace and greater number of kinematic singularities.

2.1.3 Required Hardware and Software

In this subsection we discuss the hardware and software that was used to conduct the robotic sanding experiments in this thesis. We briefly describe the sander which was selected, the corresponding fixture, the force-torque sensor, and the critical software that was used in the implementation.

2.1.3.1 Robot Model

The robot that is used is an ABB IRB4600 6-DOF articulated robot. This robot was selected for its large payload and large workspace, which makes it an ideal candidate for performing a

sanding procedure. The robot used for experimentation is shown in Figure 2.1 below

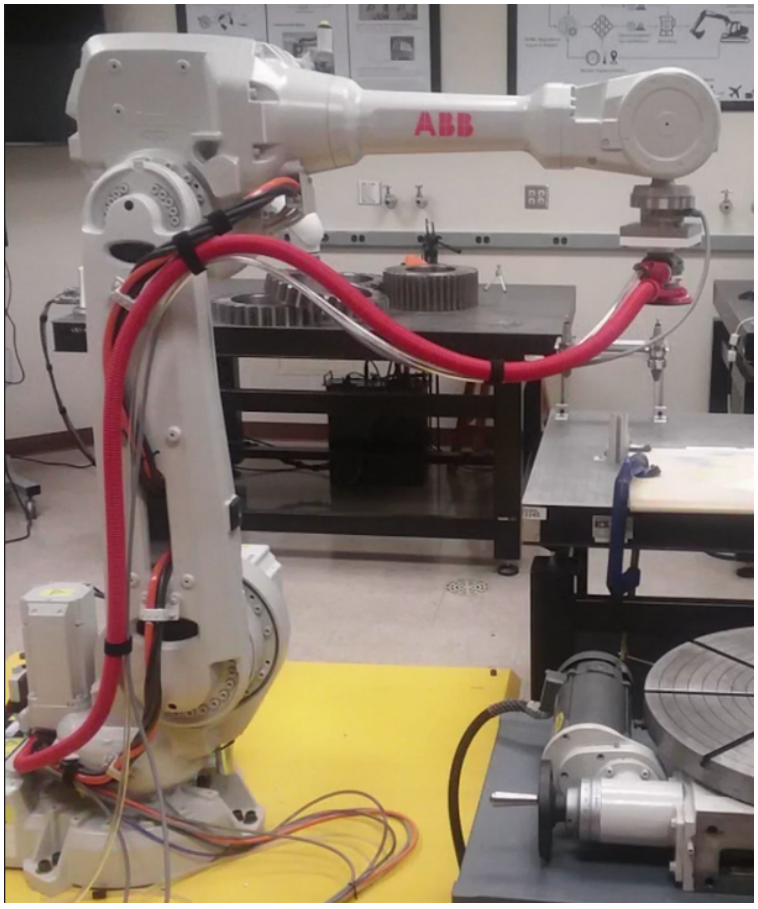


Figure 2.1: ABB IRB4600 industrial robot used for experimentation

2.1.3.2 Tooling and Fixture

The sander that was selected is a hand-held ZFE pneumatic rotary sander. This sander was chosen in part because of its simple design, making it easy to mount to the robot. Furthermore, because the sander is pneumatic, it is significantly more powerful than electric sanders of similar size. To fix the sander to the robot end-effector, we used a two-piece 3D-printed fixture which we designed using CAD.

2.1.3.3 *Sensor Selection*

The force-torque sensor that was used to provide force feedback was an ATI Omega85 force-toque sensor. This sensor was chosen in part because its maximum load (600 N) was well-above the maximum force we would need to apply for a typical sanding operation (50 N). This particular sensor also comes with an online GUI to check its status and is also ROS-compatible. The combined setup of the sensor, sanding tool, and fixture is shown in Figure 2.2 below.

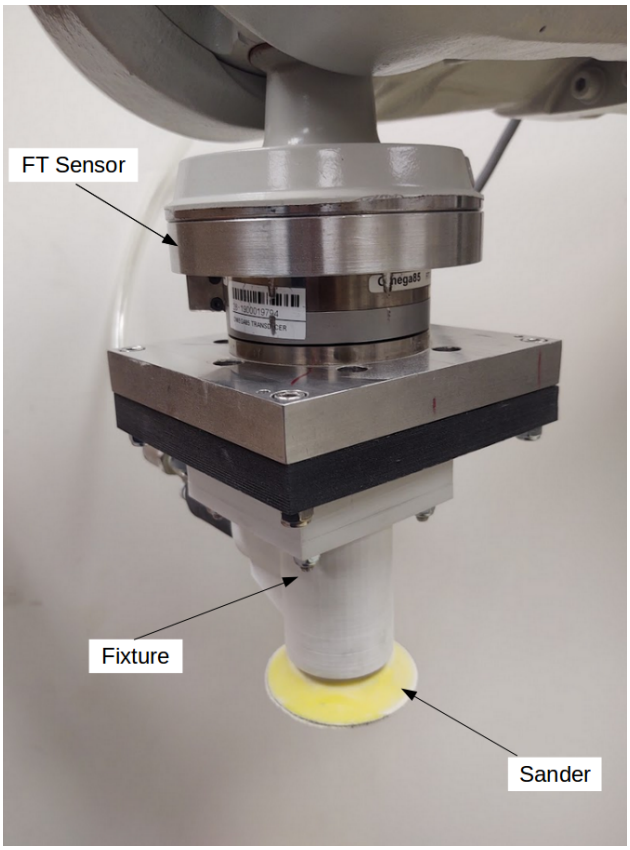


Figure 2.2: FT sensor, fixture, and sander mounted to robot end-effector

2.1.3.4 *ROS Implementation*

Robot Operating System (ROS) is a set of libraries that may be installed on a computer which allows for easy integration of multiple components. For example, ROS allows joint angles and

force data to be simultaneously used in a program which also sends commands to the robot via ROS. This software component is commonly used within the robotics community and is an important component in the experimental implementation.

2.1.3.5 Externally-Guided Motion Library

Externally-Guided Motion (EGM) is a library that is proprietary to ABB which was used to send joint velocities to the robot via an external computer. EGM enabled the real-time reading and writing of joint velocity, which was necessary for the implementation of hybrid position/force control.

2.2 Workpiece Requirements

This subsection discusses the geometrical considerations that must be taken into account during surface finishing of curved surfaces. Unlike the contact area with planar surfaces, the contact area between the tool and a non-uniformly curved surface is not the same everywhere on the surface. The change in contact area has several implications with trajectory planning and end-effector control.

2.2.1 Nonconstant Contact Area

For many robotic finishing processes, the contact area between the tool and the surface plays a critical role in the finishing process. Tools such as rotary sanders and polishers have flat polishing or sanding pads. When the tool is pressed against a curved surface, only a portion of the pad will be physically touching the surface. The contact area is characterized in [26] for surfaces of varying concavities by making the assumption that the local curvature extends beyond the local point of interest. For a convex surface, the contact area is approximated as either a circle or an ellipse. The contact area is circular if surface is completely flat or spherical, meaning that the principal curvatures are equal. The contact area is an ellipse if the principal curvatures are different from each other. The concept of principal curvature will be discussed in Chapter 3.

Likewise, if the surface is concave in a single direction, we may assume that the contact area is the intersection between a hyperboloid and a plane. If the surface is concave in all directions,

then only an outer ring of the sanding pad makes contact with the surface. The shape of these contact areas is shown in Figure 2.3 below

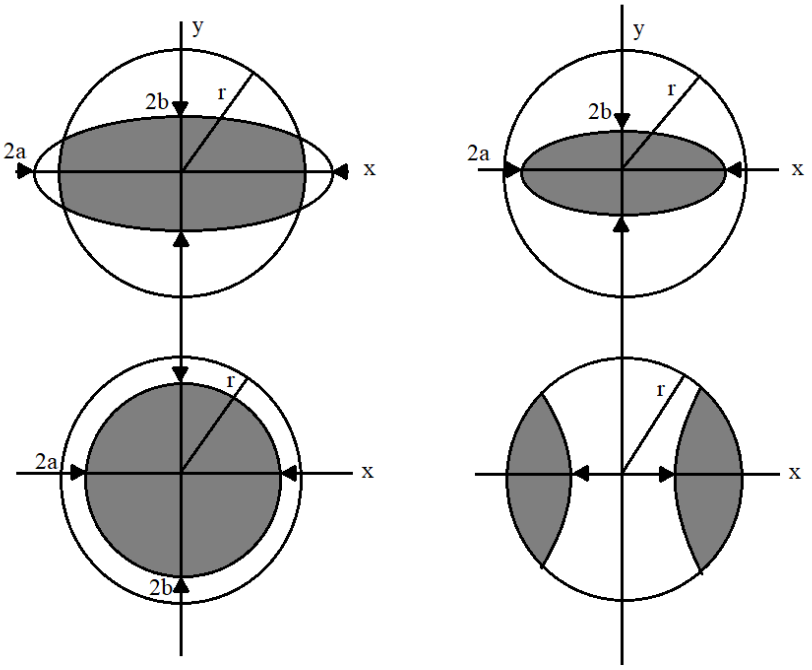


Figure 2.3: Possible contact areas between sanding pad and surface

Because a free-form surface likely has non-uniform curvature, the contact area between the tool and the surface will change as the tool traverses the surface. The change in contact area will impact the generated trajectory, as will be discussed in Chapter 4.

2.2.2 Material Removal

Because there are several variables involved in a sanding process, most of the literature that attempts to characterize material removal via sanding is empirical. One paper focused on predicting the depth of cut for a belt sander grinding process [27]. In this paper, a combination of an artificial neural network and fuzzy rule architecture was used to develop a model that was fairly accurate for a specific material (AL 6061). In [11], a material removal model was developed to

estimate the material removal from metal surfaces using a rotating polisher. Both of these papers assume a Hertzian contact model which is described in [28].

2.2.2.1 Pad Size

One geometrical consideration that must be taken into account is the pad size that is used to sand a particular workpiece. For surfaces which include highly convex or concave regions, a large sanding pad may not be the most suitable. In a region that is highly concave, only the outer portion of the pad will be in contact with the surface. Similarly, in a region that is highly convex, only the center region of the pad will remain in contact.

For regions on the surface with high concavity, it may be necessary to use a smaller-diameter sanding pad. This is because the gap between the sanding pad and concave surface is smaller for a pad of smaller diameter as shown in Figure 2.4 below.

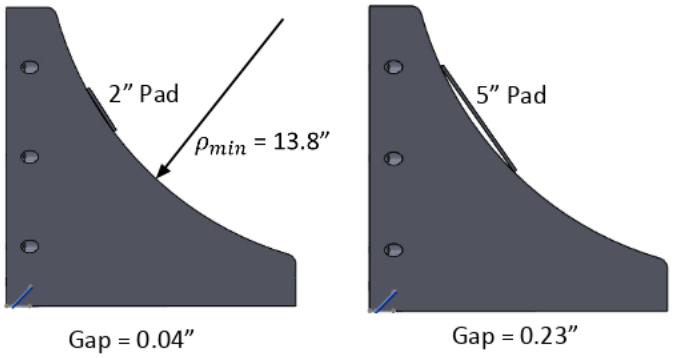


Figure 2.4: Left: Small gap between surface and small diameter sanding pad Right: Large gap between surface and large diameter sanding pad

Although a small gap still remains between the small-diameter sanding pad and surface, this gap can be overcome when a small force is applied to the sanding tool. Furthermore, this gap is small enough to use a thin "buffer pad", which allows the entire sanding pad to make contact with the surface. For simplicity, the experimentation performed in this thesis was only conducted on convex regions, however this analysis is necessary when both concave and convex regions are

sanded.

2.2.3 Workpiece Bounding Region

This subsection describes a generic procedure used to extract a bounding region for a given workpiece. The purpose of using a bounding region is to automate the trajectory generation used for scanning the workpiece for reconstructing the surface that is to be finished. Although a bounding box was not used in this thesis, it is a step that is frequently used for sanding objects which are placed in an arbitrary location.

2.2.3.1 General Procedure

Before a sensor can be used to reconstruct the workpiece, it is necessary to first determine the bounding region of the workpiece. The bounding region is the minimum planar region which fully covers the surface which is to be sanded. The bounding region is used to develop a trajectory for the robot to follow to reconstruct the workpiece surface such that the trajectory lies only inside the bounding region. In addition to shortening the length of time required for reconstructing the workpiece, the bounding region prevents the scanning of regions that are not to be sanded.

In order to determine the bounding region of the workpiece, a depth camera is positioned directly above the object such that the entire object is within the image space of the camera. The positioning of the camera depends on the location of the workpiece with respect to the robot. The image collected from the depth camera is used to construct a minimal boundary which encapsulates the workpiece.

2.2.3.2 Methods of Finding Bounding Box

A survey of literature shows that there are numerous methods for computing the minimal bounding box of an object given an image of that object. These algorithms range in complexity and depend on how much information is known apriori about the workpiece. The most simplistic method of finding a bounding box is to first use a clipping plane algorithm which detects the object by looking at the difference in point clouds of the work cell with the workpiece present and the work cell without the workpiece [29]. A minimal bound box algorithm, such as that presented

in [30], can be used to find the minimal bounding box which encapsulates the workpiece.

In this thesis, we assume that the part location is provided before the object is scanned. Thus, we avoid using any bounding box algorithms for carrying out the experiments shown in later sections. However, if the scanning procedure were to be automated for an arbitrarily-placed workpiece, a bounding-box algorithm would be necessary.

2.2.4 Workpiece Model

A critical requirement for robotic sanding of a workpiece is the development of an accurate model of the workpiece. Once the bounding box of the a workpiece is determined, there are two methods of developing a workpiece model. The first method is to use a prepared CAD model and register this model to the bounding box previously found. This option is quite often use as discussed in [31]. The main limitation of using a CAD model, however, is that the CAD model must closely match the real workpiece.

If a perfect CAD model is unavailable, a model must be developed by measuring the surface profile of the workpiece. This measurement can be performed using any of the optical measurement devices mentioned in the previous chapter. Although obtaining a surface model empirically eliminates the need of a CAD model, obtaining an empirical model which closely matches the surface is an open problem in surface reconstruction. This problem of surface reconstruction is addressed in Chapter 3.

2.2.5 Model Registration

Registration refers to the alignment of two points clouds or surfaces. One surface is comprised of measured data and the other surface is the nominal surface. The registration of a workpiece with respect to the robot frame is a challenging problem. There are two categories of registration: Contact and Noncontact. Contact registration involves making physical contact with the surface at several points to measure their position. Noncontact registration involves using noncontact sensors such as cameras or lasers to take measurements from the surface. Iterative Closest Point (ICP) is an algorithm that is often used for registration [32]. However, this algorithm requires

corresponding points to be known ahead of time, which is known as the correspondence problem [3].

In this thesis, the workpiece used in our experimentation is perfectly rectangular and non-symmetric in one dimension, which makes the registration problem much easier. We take the noncontact approach to measuring the workpiece, as more data can be collected using this method in a shorter amount of time.

3. SURFACE RECONSTRUCTION AND REGISTRATION

This chapter is focused on the surface reconstruction of free-form surfaces. In the field of computer graphics, surface reconstruction refers to the process of obtaining a numerical representation of a physical surface using some form of empirical data. This surface representation can then be used for generating trajectories, which we perform in Chapter 4. In Section 3.1, we introduce some concepts from differential geometry which will be utilized in surface reconstruction and trajectory planning. In Section 3.2, the post-processing of the raw surface data is discussed. The chapter concludes with Section 3.3 which details the process used to register the workpiece section with respect to the robot frame.

3.1 Differential Geometry Preliminaries

This section provides an overview of differential geometry concepts which will be utilized throughout the remainder of this thesis. We first define parametrized curves and how the local curvature and arc length of curves are obtained. In the following subsection, we define two representations of surfaces and discuss which representation is best suited for our scenario. Finally, we discuss some surface transformations which provide local information about the surface.

3.1.1 Parametrized Curves

The simplest definition of a curve in Euclidian space is a continuous subset of points. One common way of defining three-dimensional curves is using a parameter, which maps the real line R to R^3. Let's define this map as α. A parametrized differentiable curve is a differentiable map α: $I \longrightarrow R^3$ of an open interval $I = (a, b)$ of the real line R into R^3. Differentiable means that α is a correspondence which maps each $t \in I$ into a point $\alpha(t) = (x(t), y(t), z(t)) \in R^3$ in such a way that the functions $x(t)$, $y(t)$, and $z(t)$ are differentiable [33].

Given $t \in I$, the arc length of a parametrized curve $\alpha : I \longrightarrow R^3$, from the point t_o, is given by

$$s(t) = \int_{t_o}^{t} \|\alpha^{'}(t)\| dt \tag{3.1}$$

If α is a curve that is parametrized by arc length, then the Frenet-Serret curvature of α at s is given by

$$k(s) = \|\alpha^{''}(s)\| \tag{3.2}$$

3.1.2 Surface Representations

In this subsection we provide two methods of representing a three-dimensional surface. After providing definitions of each representation, we justify why a strictly mathematical representation is needed.

3.1.2.1 Mesh Representation

A mesh is a method of representing a surface which divides the surface into several flat polygons. In regions where the surface is highly curved, the polygons which are used to approximate the surface are smaller. The polygons, or faces, are connected via edges and each edge consists of two vertices. Although quadrilateral meshes are sometimes used, the most common type of mesh is the triangular mesh.

Triangular meshes are frequently used for surface reconstruction and have been used for trajectory generation [34]. Meshes are popular in computer graphics because they are computationally efficient and easy to render. The main limitation of triangular meshes is that they are not a true representation of the surface. Even if very small triangles are used, the surface is still represented by linear interpolation [6].

3.1.2.2 NURBS Representation

Another popular form of representing three-dimensional surfaces is using a NURBS surface, which represents the surface parametrically in the form of basis functions [35]. Unlike triangular meshes, NURBS are usually at least twice continuously differentiable, except for on the surface

boundary.

Similar to triangular meshes, NURBS have also been used extensively for surface reconstruction and trajectory generation. In [36], NURBS were used for reconstructing complex objects including a vase and human head and have been used for generating trajectories in obstacle avoidance scenarios [37].

We provide the definition of a bivariate NURBS surface, defined using the parametric dimensions u and v. Let $U = \{u_0, ..., u_m\}$ and $V = \{v_0, ..., v_m\}$ be two non-decreasing sequences of real numbers. The values u_i and v_i are referred to as *knots* and U and V are referred to as *knot vectors*. Let $N_{i,p}$ be a recursive basis function of degree p defined as

$$N_{i,0}(u) = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{3.4}$$

The above basis function is defined using the u parametric direction, but can likewise be defined for v.

An important component of NURBS surfaces is the *control point*, which is a point in Euclidean space which influences the local shape of the surface. Though control points usually lie near the surface, they may not be coincident with the surface. Let $P_{i,j}$ be a single control point. Then the set of control points $\{P_{i,j}\}$ is a bidirectional control net of control points. Although the control net dimensions are usually orthogonal, this is not a necessary condition.

Let $w_{i,j}$ be the corresponding weights to each control point in the control net [35]. We then define a NURBS surface, which is a bivariate vector-valued piecewise rational function defined for $0 \leq u, v \leq 1$ of the form

$$S(u, v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} P_{i,j} N_{i,p}(u) N_{j,q}(v)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} N_{i,p}(u) N_{j,q}(v)} \tag{3.5}$$

It should be noted that these dimensions form a basis which span the surface, but unlike

Euclidean dimensions, they may not always be orthogonal to each other.

The main advantage NURBS have over triangular meshes is their ability to capture a large variety of shapes (e.g. conics and free-form surfaces) and can represent these surfaces more precisely [38]. The trajectory planner which we present in Chapter 4 requires accurate local curvature information, so for this reason we have chosen to reconstruct the workpiece using NURBS.

3.1.3 Surface Operations

In this subsection we provide the mathematical definitions of three different operations which are performed on parametric surfaces. These concepts are utilized more in 4 for approximating the local contact area on a surface.

3.1.3.1 Gauss Map

The first concept which we use in our derivations is the Gauss map, which transforms any point on a surface to its unit normal. The Gauss map N is a linear operator which maps $p \in S$ to the unit sphere and is defined as

$$N(p) = \frac{\boldsymbol{x_u} \times \boldsymbol{x_v}}{|\boldsymbol{x_u} \times \boldsymbol{x_v}|}(p) \tag{3.6}$$

where $p \in x(S)$. The map N transforms the unit normal at point p and maps the normal to a unit sphere.

If the map N is a differentiable map which associates to each point p a normal vector at p, we say that N is a differentiable field of normal vectors. This differential, dN_p, measures how quickly N pulls away from $N(p)$ [33].

3.1.3.2 Normal Curvature

The second concept which we use to approximate the local contact area is normal curvature. If we define C as a curve in surface S passing through a point $p \in S$, k the curvature of C at p, and $\cos\theta = \langle n, M \rangle$, where n is the normal vector to the curve C and M is the normal vector to

the surface S. The number $k_n = k\cos\theta$ is then called the normal curvature of $C \subset S$ at p, where k is the curvature defined in Equation 3.2. The maximum and minimum normal curvatures at a point p are known as the principal curvatures and are given by k_1 and k_2.

3.1.3.3 Fundamental Forms

The third concept which we define is the fundamental forms of a surface. We first define the first fundamental form. Let $w_1, w_2 \in T_p(s) \subset R$ where $T_p(s)$ is the tangent plane of the surface. To the inner product $\langle w_1, w_2\rangle$ there corresponds a quadratic form $I_p : T_p(S) \to R$ given by

$$I_p(w) = \langle w, w\rangle_p = |w|^2 \geq 0 \tag{3.7}$$

where I_p is called the first fundamental form of the surface S. We can express the first fundamental form in $\{\boldsymbol{x}_u, \boldsymbol{x}_v\}$ basis. Since a tangent vector $w \in T_p(S)$ is also tangent to a parametrized curve $\alpha(t) = \boldsymbol{x}(u(t), v(t))$, $t \in (-\epsilon, \epsilon)$, with $p = \alpha(0) = \boldsymbol{x}(u_o, v_o)$, we obtain

$$\begin{aligned} I_p(\alpha^{'}(0)) &= \langle \alpha^{'}(0), \alpha^{'}(0)\rangle_p \\ &= \langle \boldsymbol{x}_u u^{'} + \boldsymbol{x}_v v^{'} \boldsymbol{x}_u u^{'} + \boldsymbol{x}_v v^{'}\rangle_p \\ &= \langle \boldsymbol{x}_u, \boldsymbol{x}_u\rangle_p (u^{'})^2 + 2\langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle_p u^{'} v^{'} + \langle \boldsymbol{x}_v, \boldsymbol{x}_v\rangle_p (v^{'})^2 \\ &= E(u^{'})^2 + 2F u^{'} v^{'} + G(v^{'})^2, \end{aligned} \tag{3.8}$$

where

$$\begin{aligned} E &= \langle \boldsymbol{x}_u, \boldsymbol{x}_u\rangle_p \\ F &= \langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle_p \\ G &= \langle \boldsymbol{x}_v, \boldsymbol{x}_v\rangle_p, \end{aligned} \tag{3.9}$$

The p subscript in Equations 3.8 and 3.9 can be dropped, as we are usually provided the loca-

tion on the surface.

We likewise define the second fundamental form for a surface. Consider a regular curve $C \subset S$ parametrized by $\alpha(s)$, where s is the arc length of curve C and $\alpha(0) = p$. If N is the normal vector to the curve $\alpha(s)$, we have

$$\langle N(s), \alpha^{'}(s) \rangle = 0 \tag{3.10}$$

By product rule, we can write

$$\langle N(s), \alpha^{''}(s) \rangle = -\langle N^{'}(s), \alpha^{'}(s) \rangle \tag{3.11}$$

The second fundamental form defined at point p is given by

$$\begin{aligned} II_p(\alpha^{'}(0) &= -\langle dN(\alpha^{'}(0)), \alpha^{'}(0) \rangle \\ &= -\langle N^{'}(0), \alpha^{'}(0) \rangle \\ &= \langle N^{(}0), \alpha^{''}(0) \rangle \\ &= \langle N, kn \rangle(p) = k_n(p) \end{aligned} \tag{3.12}$$

In other words, the second fundamental form II_p for a unit vector $v \in T_p(S)$ is the normal curvature of a curve passing through p and tangent to v. Equation 3.12 can be visualized using Figure **??**. The expression of the second fundamental form in the $\{\boldsymbol{x}_u, \boldsymbol{x}_v\}$ basis is given by

$$II_p(\alpha^{'}) = -\langle dN(\alpha^{'}), \alpha^{'} \rangle = e(u^{'})^2 + 2fu^{'}v^{'} + v(v^{'})^2 \tag{3.13}$$

where

$$\begin{aligned} e &= \langle N, \boldsymbol{x_{uu}} \rangle \\ f &= \langle N, \boldsymbol{x_{uv}} \rangle \\ g &= \langle N, \boldsymbol{x_{vv}} \rangle \end{aligned} \tag{3.14}$$

3.1.3.4 Shape Operator

The final concept which we utilize is called the shape operator, which is used to determine the principal curvatures and their respective directions for a given $p \in S$. We provide a short derivation of how the shape operator is used to find the principal curvatures for a given point. Let $\alpha(t) = \boldsymbol{x}(u(t), v(t))$ be a parametrized curve on S. The tangent vector to $\alpha(t)$ at p is given by

$$\alpha^{'} = \boldsymbol{x_u} u^{'} + \boldsymbol{x_v} v^{'} \tag{3.15}$$

Differentiating the Gauss map with respect to these parameters, we have

$$dN(\alpha^{'}) = N_u u^{'} + N_v v^{'} \tag{3.16}$$

where

$$N_u = a_{11}\boldsymbol{x_u} + a_{21}\boldsymbol{x_v} \tag{3.17}$$

$$N_v = a_{12}\boldsymbol{x_u} + a_{22}\boldsymbol{x_v} \tag{3.18}$$

hence,

$$dN \begin{pmatrix} u^{'} \\ v^{'} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} u^{'} \\ v^{'} \end{pmatrix} \tag{3.19}$$

The goal is to find the coefficients a_{ij} which define the differential dN. Using the first and

second fundamental forms defined in Equations 3.8 and 3.12, we have

$$\begin{aligned} -f &= \langle N_u, \boldsymbol{x_v} \rangle = a_{11}F + a_{21}G \\ -f &= \langle N_v, \boldsymbol{x_u} \rangle = a_{12}E + a_{22}F \\ -e &= \langle N_u, \boldsymbol{x_u} \rangle = a_{11}E + a_{21}F \\ -g &= \langle N_v, \boldsymbol{x_v} \rangle = a_{12}F + a_{22}G \end{aligned} \tag{3.20}$$

These equations can be used to solve for the values a_{ij} which are presented below in matrix form

$$\begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} = - \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \tag{3.21}$$

which is equivalent to the differential dN. This matrix is also commonly referred to as the shape operator.

Because the negative of the principal curvatures $-k_1$ and $-k_2$ are the eigenvalues of dN, we have

$$dN(v) = -kv = -kIv \tag{3.22}$$

for some direction v in the tangent plane of S. We see that the right-hand side can be moved to the left, making the determinant zero. Solving for the determinant, we obtain the characteristic equation of a_{ij}

$$k^2 - 2Hk + K = 0 \tag{3.23}$$

where $K = det(a_{ij})$ and $H = -\frac{1}{2}(a_{11} + a_{22})$. From here we can easily solve for k_1 and k_2. Once the eigenvalues of a_{ij} are known, the eigenvector v_i can be easily found for the corresponding eigenvalue k_i.

3.2 Improving Reconstruction Accuracy

In this section we describe the post-processing which is performed on the raw surface data collected from the point laser. In the first subsection, we briefly describe the hardware and scanning trajectory that were used to collect the raw data. In the second subsection, we discuss a method of sampling the surface which improved the accuracy of the NURBS reconstruction. In the third subsection, we explain how multiple linear regression can be used to further filter the noisy raw data so that there are no sudden jumps in curvature.

3.2.1 Scanning Trajectory and Hardware

The device which was used for measuring the surface profile was a SICK DT20 short-range distance point laser. This laser is used for measuring displacement and is accurate to ± 0.5 mm. Because the point laser can only measure distance at a single point, the surface is measured at incremental locations. After mounting the point laser to the end-effector of the robot, the robot follows a scanning trajectory, briefly stopping at incremental way points for the laser to measure the distance to the surface. Several measurements are taken rapidly and the average of these measurements is used as the Euclidean measurement of a given location. These measurements are then used as control points in a non-uniform rational B-spline (NURBS) representation of the surface.

3.2.2 Multiple Linear Regression

The raw data that is collected using the point laser is noisy for a multitude of reasons, such as sensor characteristics and imperfections in the surface. The impact this noise has on the smoothness of the surface is seen in a plot of the raw data in Figure 3.1 which was plotted using the "NURBS Python" library [39].

To reduce the effects of noise on the surface, we employ a process referred to as "skinning". In this process, least squares splines are first constructed in one parametric direction and used to reduce the noise in that direction [40]. This process is greatly simplified when the workpiece takes a particular shape. When the cross-sectional area of an object does not change along a direction,

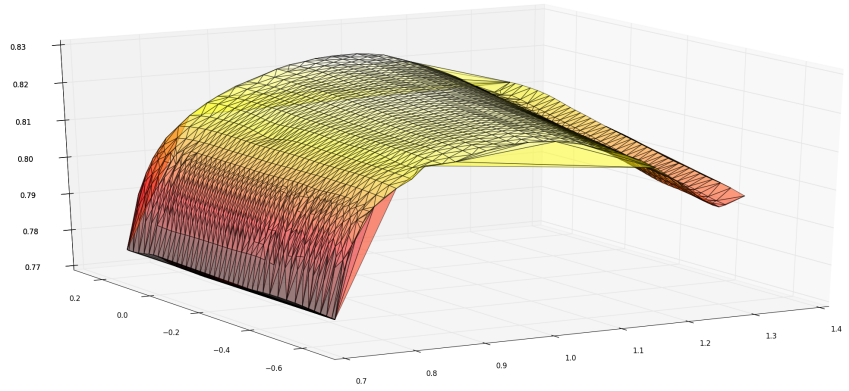


Figure 3.1: NURBS surface plot of raw data collected from point laser, prior to data processing

it is said that the object has uniform cross-sectional area along that direction. For surfaces with uniform cross-sectional area in one direction, we can consider spline approximations to be linear along this direction. The helicopter blade section which we use for experimentation is an example of an object with uniform cross-sectional area.

The linear splines along the direction of uniform cross-sectional area are approximated by applying multiple linear regression to sample data which was collected within a predetermined distance from a given linear spline. The dotted lines in Figure 3.2 represent the approximate region which was used for sampling surface data which would be used for the multiple linear regression. The solid lines are the linear splines which are constructed from the sampled data and the evenly-spaced points are the new control points which are used to reconstruct a surface with the noise removed.

The surface produced by the multiple linear regression operation is significantly smoother than the surface constructed from the raw data and is shown in Figure 3.3.

3.2.3 Local Curvature-based Sampling

In this subsection we describe how we improved the sampling of the surface profile by empirically estimating the curvature of the workpiece in a single direction. A distribution function is computed which distributes more points in regions of higher curvature.

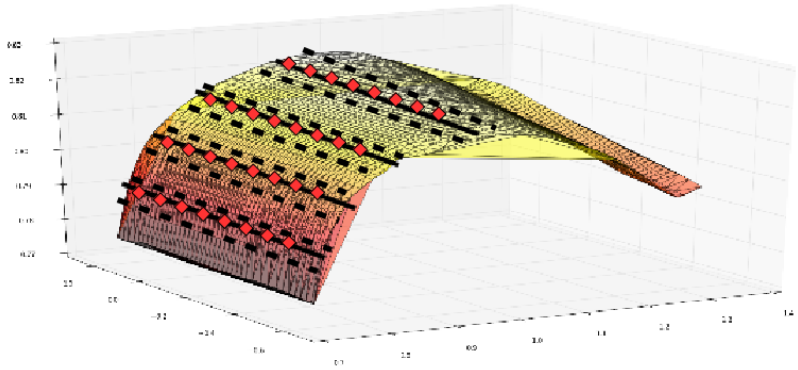


Figure 3.2: Surface data obtained via multiple linear regression

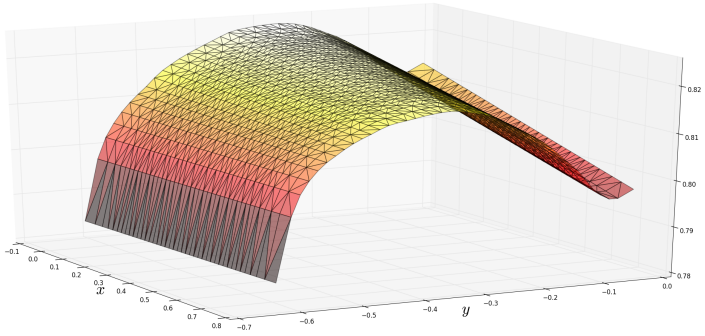
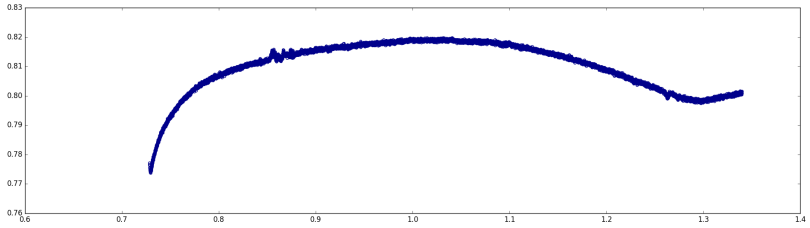


Figure 3.3: Surface after multiple linear regression operation

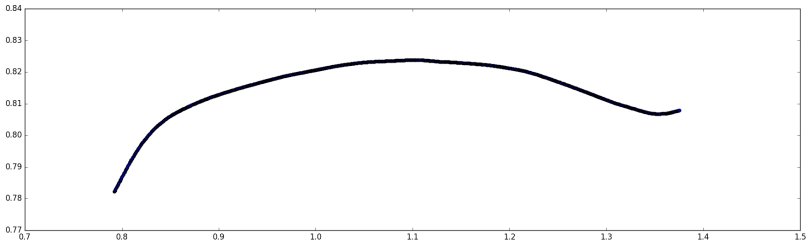
3.2.3.1 Empirical Curvature Measurement

One critical assumption we make for sampling the surface is that the surface is only curved in a single direction. This assumption ensures that the parametric directions which are used to define the NURBS surface align with the Euclidean directions, which greatly simplifies the trajectory generation.

Before we can modify the sampling of the surface, we first need to empirically estimate the surface curvature. As stated earlier, we are assuming that the cross-sectional area of the work-piece is uniform. Thus, it is only necessary to perform empirical measurements in the direction



(a)



(b)

Figure 3.4: a) Laser measurements of blade contour b) Contour spline after local linear regression

in which the curvature changes. The following figures show the data that was collected by ten successive scans in the curved direction of a helicopter blade section. Figure 3.4 shows the raw data measured by the point laser along side the processed data which was obtained using local linear regression.

3.2.3.2 Sampling Distribution Function

Here we present the distribution function which we will use to determine the spacing between successive scans using the point laser [41]. The distribution function takes a curve parametrized by arc length as an input and outputs a distribution function which is used to space points along the curve such that more points are located in regions of higher curvature. The distribution function, parametrized in terms of arc length s, is defined as

$$R(s) = qb(s)/B + (1 - q)s/L \tag{3.24}$$

where L is the total length of the curve, q is a factor which determines how strongly curvature

affects point spacing, and $b(s)$ and B are defined respectively as

$$b(s) = \int_0^s k^2(t)dt \tag{3.25}$$

$$B = b(L) \tag{3.26}$$

where k is the curvature defined in Equation 3.2.

A NURBS curve is formed using the processed data shown in Figure 3.4 and is parametrized in terms of arc length using a rational spline as presented in [42]. The resulting monotonic distribution function which was presented in Equation 3.24 is shown in Figure 3.5

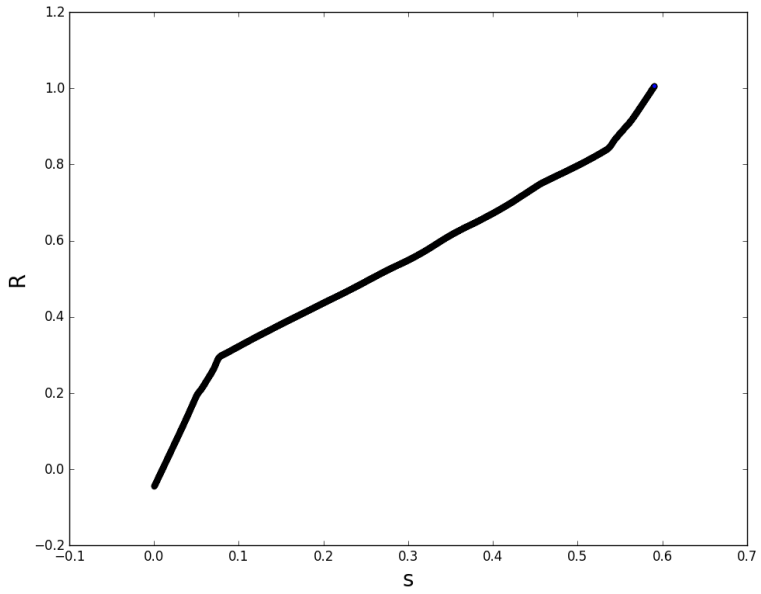


Figure 3.5: Monotonic distribution function for point sampling

The distribution function relates arc length s to a uniform distribution of points, which is represented by $R(s)$. The inverse of this distribution function is used to convert an evenly-distributed sampling to a sampling that distributes more points to regions of higher curvature. Figure 3.6

shows the contour sampling after the distribution function is applied

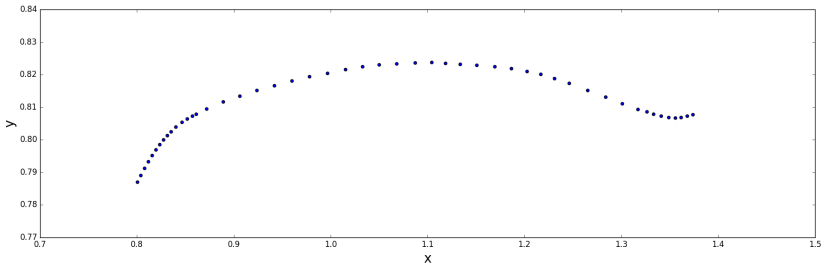
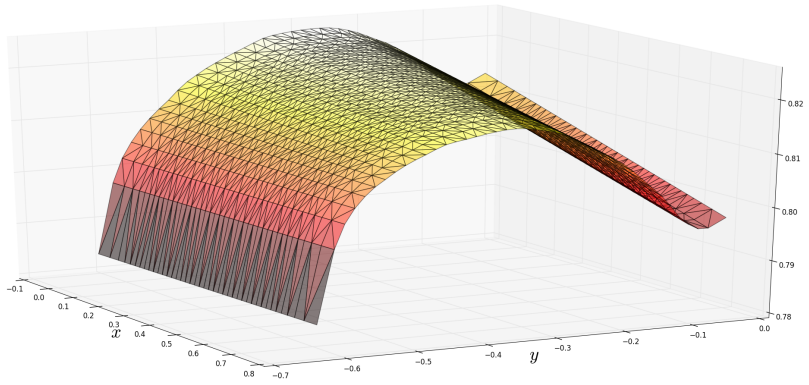


Figure 3.6: Point sampling distributed based on local curvature

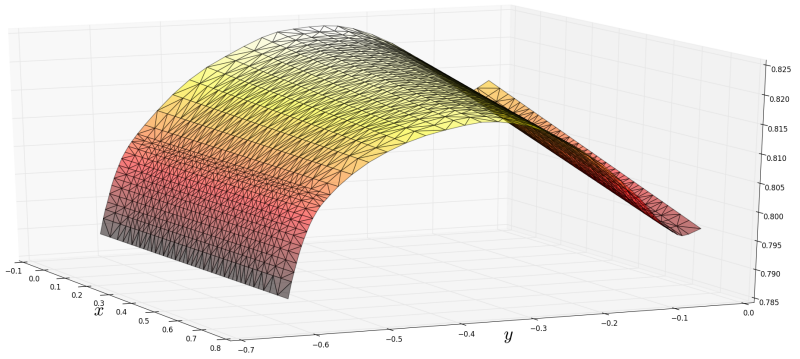
The distributed points shown in Figure 3.6 are used to resample the workpiece such that each horizontal scan has the same x coordinate as the corresponding distributed point. Thus more scans are allocated to regions of the surface with higher curvature. Figure 3.7 shows a side-by-side comparison of a surface produced by data collected at even spacing and a surface produced by data collected at spacing determined by local curvature. The surface which accounts for local curvature has fewer abrupt changes in curvature, especially in regions where the curvature changes quickly.

3.3 Workpiece Registration

In this section we provide a brief procedure which we used to register the workpiece with respect to the robot frame. The purpose of registration is to align a nominal and highly accurate CAD model with a corresponding measurement model which represents the physical workpiece which is to be sanded. Normally, the purpose of registration is to use a precise CAD model for trajectory generation, since accurate surface reconstruction is not possible for many complex geometries. In this thesis, we are able to directly use the reconstructed surface for trajectory generation, as the helicopter blade geometry is simple and no CAD model was provided. However, we provide a brief registration procedure that could be used for registration should a need arise.



(a)



(b)

Figure 3.7: a) Surface with evenly-spaced sampling b) Surface with sampling based on local curvature

3.3.1 Workpiece Assumptions

The registration procedure that we outline requires some assumptions be made about the workpiece that is being registered. The first assumption is that both the CAD model and measured model are assumed to both have a flat side which makes contact with the table used to position the workpiece. Because we know that these two sides must be coincident with each other, we only need to consider rotation about the z axis and translation in the xy plane as demonstrated in 3.8

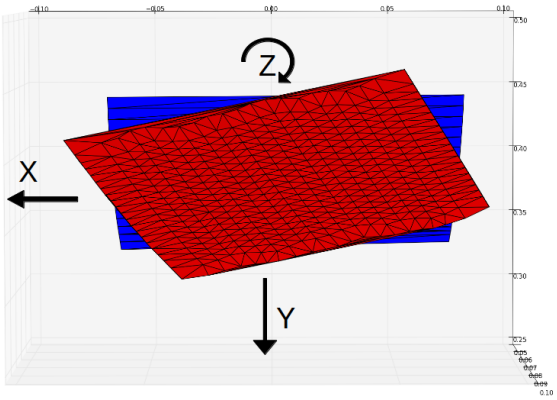


Figure 3.8: Unknown degrees of freedom after assuming flat side of blade models are coincident

The second assumption that is made about the workpiece is that some points may be measured with a higher degree of confidence than other points. In the case of the helicopter blade section used in this paper, we assume that the outside perimeter of the measured model is nearly identical in shape and dimensions to those of the nominal CAD model. Since these points hold a higher degree of confidence, they may used as a starting point for adjusting the position and orientation of the nominal CAD model.

3.3.2 Registration Procedure

Workpiece registration is a complex problem with no single solution which fits every scenario. There are several constraints which limit the accuracy of the registration of a particular workpiece. Some common constraints include the accuracy of the measurement device used to measure the workpiece, the time required to obtain these measurements, and the complexity of the workpiece itself. For example, a workpiece which contains distinct features is easier to register than a highly symmetrical workpiece with many similar features. Because registration is a difficult problem, we propose a procedure which is relatively easy and quick to implement, yet provides a better registration than measuring a few points.

The first step in the registration procedure is rotating the nominal blade such that we minimize the registration error. Though it does not matter about which point we rotate, we choose to rotate about the xy centroid for simplicity. The registration error makes use of the second assumption that the perimeter of the measured workpiece has higher degree of confidence. If both surfaces are represented as NURBS surfaces, the minimization error can be the sum of the squares of corresponding points on the workpiece perimeters. As the nominal blade is rotated, the error becomes smaller as the two models become more aligned. Because there many be local minimums that are not the global minimum, the nominal model should have a similar orientation to the measured model. Figure 3.9 shows an example of point correspondence which are used to find the registration error corresponding to workpiece rotation.

After the two models are aligned along the z-axis, the registration may be further improved by translating the nominal model in the xy plane. A naive method of finding the xy translation would be to simply align the centroids of the models. However, if a more accurate registration is needed, another method of performing the translation would be to align curves of similar shape in each model. These curves can then be used to align the nominal model such that the curves in both models are coincident. Figure 3.10 shows example curves which could be matched such that the resulting translation of the CAD model results in an accurate registration.

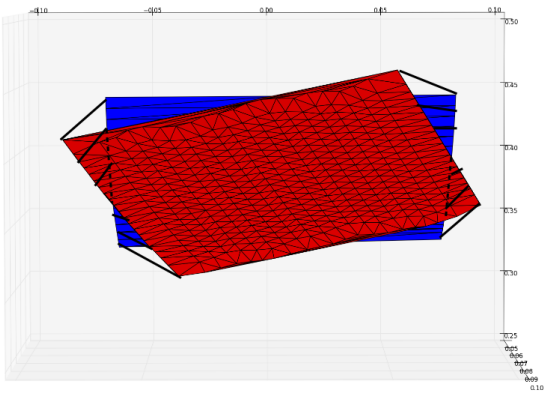


Figure 3.9: Point correspondences used for computing rotation registration error

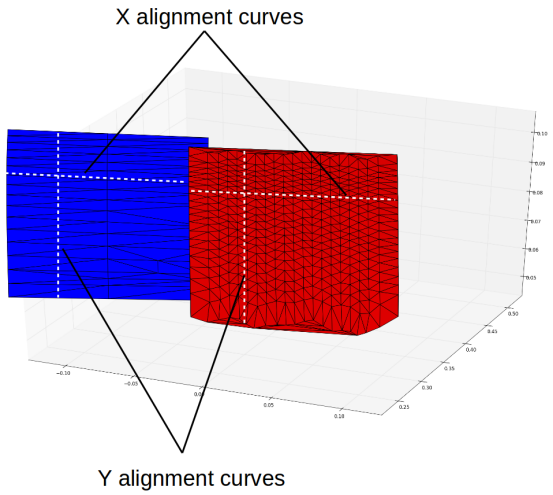


Figure 3.10: Matched alignment curves used for translating CAD model

4. TRAJECTORY PLANNING STRATEGY

In this chapter, we focus on planning the trajectory that the robot end-effector is to follow. Specifically, we develop a trajectory planner which provides uniform coverage of a free-form surface, which lessens the execution time and amount of overlap resulting from the conventional straight-line scanning trajectory. In Section 4.1 we discuss the requirements needed for a trajectory used for sanding a free-form surface. In Section 4.2, we describe the method used for approximating the effective contact area which is used in the trajectory planner. Finally, in Section 4.3, we present the algorithm used for planning a uniform coverage trajectory for free-form surfaces.

4.1 Trajectory Requirements

This section describes the necessary requirements needed for the trajectory used to sand a free-form surface with a rectangular perimeter. A helicopter blade section is used as an example surface, however, the method used in this section for trajectory generation may also be used for other free-form objects of rectangular perimeter.

4.1.1 Trajectory Parameterization

In this subsection, we first provide a general definition of the type of trajectory which will be generated. We then discuss ways in which the trajectory should be parameterized for a free-form object of rectangular perimeter.

4.1.1.1 General Definition

In general, a trajectory is a time-dependent path which describes how the robot end-effector is to move while avoiding collisions. In the robotic surface finishing applications, a trajectory consists of multiple way points which provide the location of either the end-effector or tool as the robot moves across the surface. Figure 4.1 shows a simple scanning trajectory on a helicopter blade section.

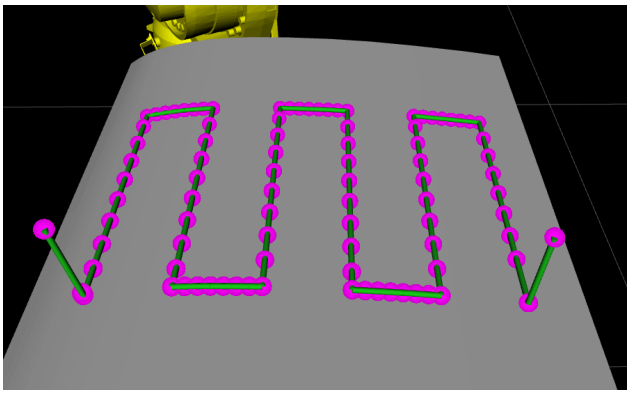


Figure 4.1: Simple scanning trajectory on helicopter blade section

The trajectory generation strategy employed in this section consists of two major steps: First, generate coordinates which are coincident with the workpiece surface along with the surface normal corresponding to each position. The second step is converting this Cartesian trajectory into joint space using the inverse kinematics corresponding to the particular robot being used for the finishing process.

4.1.1.2 Way Point Density

One parameter which should be specified for a trajectory used for robotic surface finishing is the density of the way points which comprise the trajectory. Although the way points may be spaced arbitrarily, it is preferable that the trajectory way points be evenly spaced so as to maintain a constant feedrate across the workpiece surface. If the way points are not evenly distributed, the feedrate may vary due to sudden increases in positional error.

4.1.1.3 Trajectory Shape and Dimensions

A common requirement for a trajectory used for robotic surface finishing applications is that it must completely traverse the entire desired area. One type of trajectory commonly used to achieve this objective is the scanning trajectory, which is comprised of successive back-and-forth passes. Although other trajectories are certainly possible, we chose to generate this type of trajectory for its computational simplicity.

The other trajectory parameters which we specify are the trajectory dimensions and overlap

between passes. The trajectory dimensions consist of the length and width of the area being sanded, as well as the location of this area on the object surface. The contact area refers to the area of the sanding pad which is in contact with the surface, which is dependent on how much normal force is applied. The overlap refers to the amount of overlapping in contact area between successive passes. The overlap may need to be varied depending on the type of tool being used and the surface being sanded. The diagram in Figure 4.2 below shows all of the parameters of the trajectory which are allowed to vary.

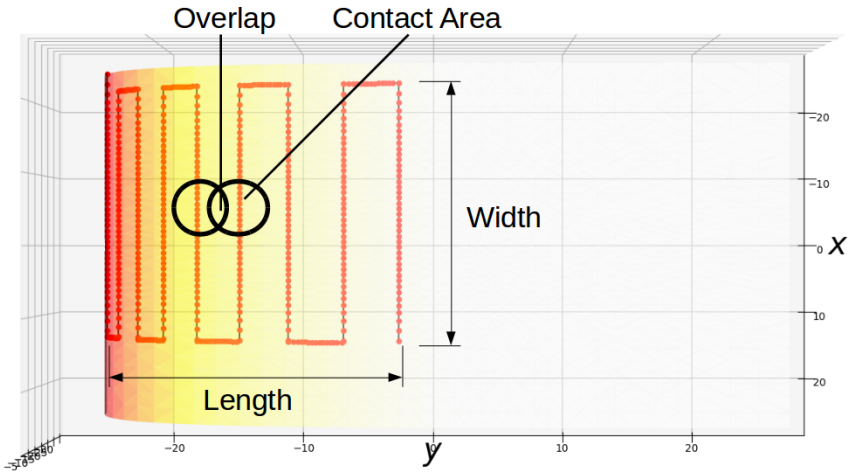


Figure 4.2: Trajectory parameters which are predetermined before sanding operation

4.1.2 Uniform Coverage

A requirement that is very important for surface finishing is uniform coverage of the surface. Uniform coverage refers to the entire surface being given equal treatment from the finishing tool. In the case of robotic sanding, uniform coverage refers to an even amount of material removed from each location in the desired treatment area.

The requirement of uniform coverage provides a significant challenge for generating trajectories when sanding free-form surfaces. As discussed in Chapter 1, the contact area between the sanding pad and workpiece may vary as the tool traverses the surfaces. The change in contact

area reflects the change in local curvature of the surface.

Because the contact area changes as the sanding tool traverses the workpiece surface, a perfectly rectangular scanning trajectory will result in unwanted overlap between successive passes. In order to get constant overlapping between passes, each successive pass must follow the boundary of the sanded area of the previous pass. The diagram in Figure 4.3 shows the overlap of a rectangular trajectory versus a trajectory which produces uniform overlap. These two types of coverage are referred to as geometric coverage and uniform coverage respectively.

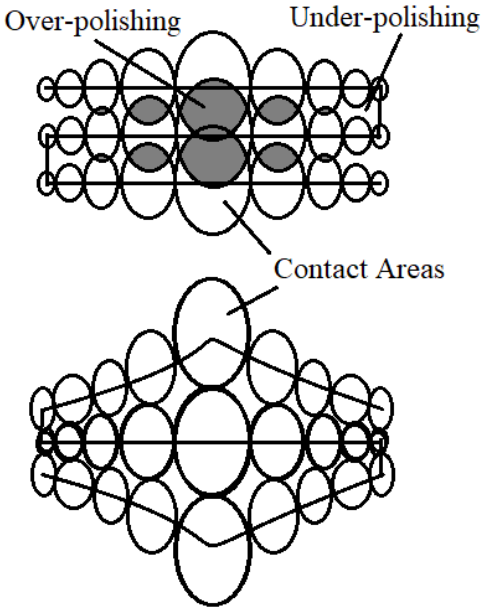


Figure 4.3: Top: Geometric uniform coverage Bottom: Physically uniform coverage

There have been a few past studies which had similar objectives of achieving uniform sanding. In [43], uniform density was achieved by using Hilbert curves (space-filling fractal curves), but there was no consideration given to the contact area between the tool and the workpiece surface. Similarly in [44], a spiral tool path was generated on a curved surface, but the contact area was assumed to be the same throughout the entire path. The tool path planner developed in [45] did extensive analysis on contact area, but the contact area was assumed to be Hertzian which is only

applicable when the tool head is also curved.

In this thesis, we choose to employ local surface curvature as employed in [46] for robotic sanding, but we relax the assumption that the entire sanding pad is in contact with the surface throughout the entire trajectory.

4.2 Contact Area Approximation using Local Geometry

In this section, we describe how the local geometry of the surface is used to approximate the contact area between the sanding pad and surface using the differential geometry concepts discussed in Chapter 3. In the first subsection, we explain how we use the shape operator to find the principal curvatures at a point. We then explain in the following subsection how the contact area is approximated using the principal curvatures.

4.2.1 Finding Local Principle Curvature

Assuming that the workpiece surface is provided in a NURBS representation as discussed in Section 3.1, we used the shape operator to find the principal curvature at any given point on the surface. The NURBS-python library provides functions which compute the local surface normal as well as the first and second derivatives for any (u, v) pair [39]. Figure 4.4 below shows a surface showing the directions of principal curvature along one direction of the surface.

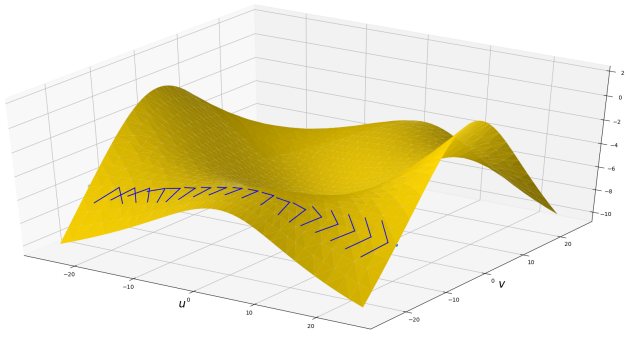


Figure 4.4: Principal directions along single direction of surface

We see that the principal curvatures are always orthogonal to each other and that the direc-

tions change continually as the surface is traversed.

4.2.2 Contact Area Approximation

Here we describe how the effective contact area is derived using the principal curvatures found in the previous subsection. We know that the principal curvatures correspond to the maximum and minimum normal curvatures and are always orthogonal to each other [47]. As presented in [26], the principal curvatures are related to the the radii of curvature by

$$p_i = \frac{1}{k_i} \tag{4.1}$$

Using the radii of curvature, we are able to find the lengths of the semi-major and semi-minor axes of the elliptical contact area, which are given by

$$w_i = 2\sqrt{p_i^2 - (p_i - e)^2} \tag{4.2}$$

where w_i are the magnitudes of the semi-minor and semi-major axes of the ellipse and e is the eccentricity which is how far the sanding pad is depressed onto the surface. Figure 4.5 illustrates where each of these values are located on the effective contact area

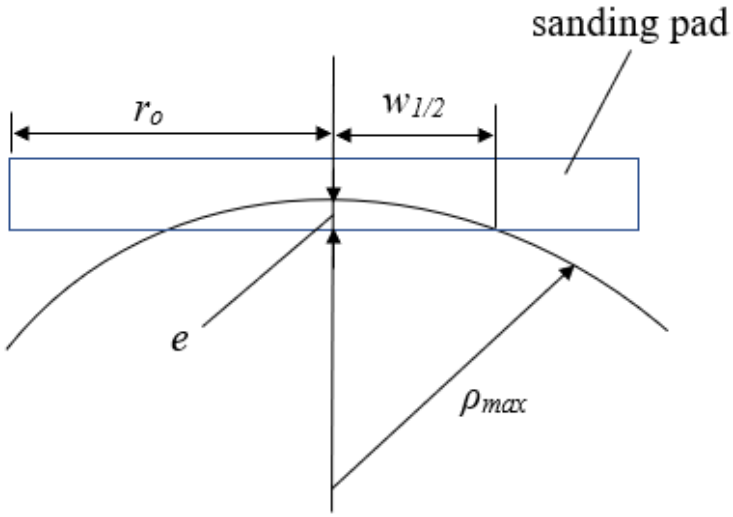


Figure 4.5: Sanding pad contact area dimensions

With the semi-major and semi-minor axes magnitudes and their corresponding directions, we are able to define an approximate boundary of the effective contact area using a parametrized equation of an ellipse given by

$$\boldsymbol{x} = \boldsymbol{f_o} + \boldsymbol{f_1} \cos t + \boldsymbol{f_2} \sin t \tag{4.3}$$

where $\boldsymbol{f_o}$ is the ellipse center, $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$ are congruent vectors corresponding to the ellipse axes, and $t \in [0, 2\pi]$, as illustrated in Figure 4.6

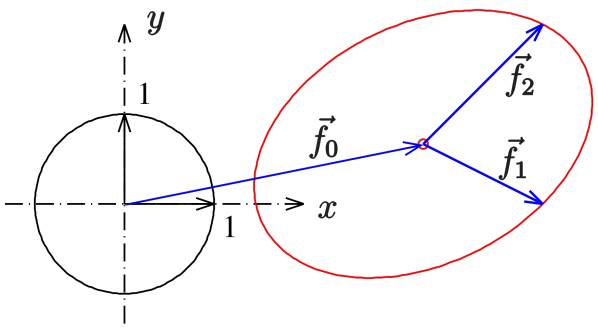


Figure 4.6: Representation of two-dimensional parametric ellipse in three-dimensional space

4.3 Trajectory Generation Procedure

In this section we describe how we utilized the approximation of effective contact area to generate a trajectory which takes into account the change in contact area as the surface is traversed. We describe the main steps taken by the algorithm along with some simulations results for two different geometries.

4.3.1 Iterative Algorithm

In this subsection we describe how the approximated effective contact area described previously is used to approximate the sanding ribbon boundary. This ribbon boundary is used as the reference point in an iterative algorithm which finds the position of the contact areas in the successive path.

4.3.1.1 Approximation of Sanding Ribbon Boundary

The algorithm must initially be provided with a seed curve which serves as a starting point for the trajectory approximation. The seed curve is discretized into base points P_i, where i represents a single base point.

The local effective contact area is found at each P_i using the method discussed in the previous section. Using the parametrized form of an ellipse, the algorithm approximates the boundary of the local effective contact area. Figure 4.7 shows the local effective contact areas plotted along one dimension of a three-dimensional surface.

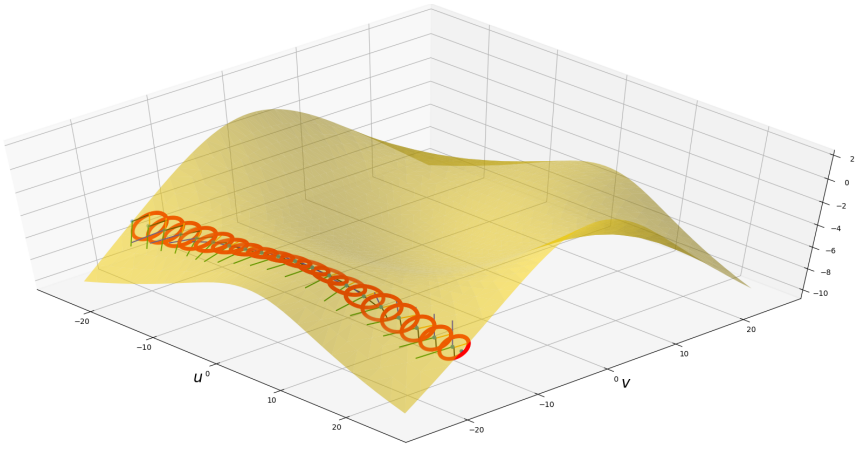


Figure 4.7: Effective contact areas plotted along a single direction on a surface

The sanding ribbon boundary is approximated by finding the location in each ellipse which is furthest away from the initial seed curve, E_{max_i}. These points are then used to form a B-spline S, which will serve as the target for in the search algorithm used to find the effective contact areas of the successive passes.

4.3.1.2 Iterative Bisection Method

Because the local curvature may vary throughout the surface, an iterative algorithm is needed for finding successive passes. For computational efficiency, we used a standard bisection method to guarantee that a defined error e_i converges to zero. The error that is used in the bisection algorithm is the difference between the location of each ellipse which is closest to the seed curve, E_{min_i}, and a point on the B-spline S form the previous pass which is coincident with E_{min_i}, which we will call S_i. Figure 4.8

For brevity, we omit any low-level explanation of the bisection method which is used to position the successive pass. We instead provide high-level pseudocode which provides the frame-

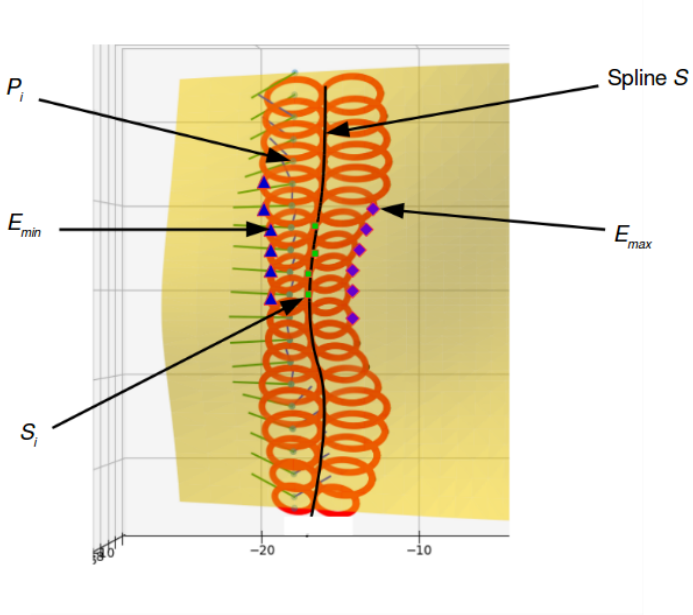


Figure 4.8: Location of minimum and maximum ellipse points and ribbon boundary spline

work logic to the trajectory planner shown in Algorithm 1

Data: Seed curve with base points P_i for $i = 0...n$;

Length of Workpiece L

while $E_{max_i} < L$ **do**

  for $i = 0$ *to len*(P_i) **do**

    while $err > tol$ **do**

      $err = E_{min_i} - S_i$;

      E_{min_i}, P_{i+1} <- Bisection(P_i);

    end

    $P_i = P_{i+1}$;

  end

  S <- Spline($P_{0...n}$);

  S_i <- S

end

Algorithm 1: High-level uniform coverage trajectory generation algorithm

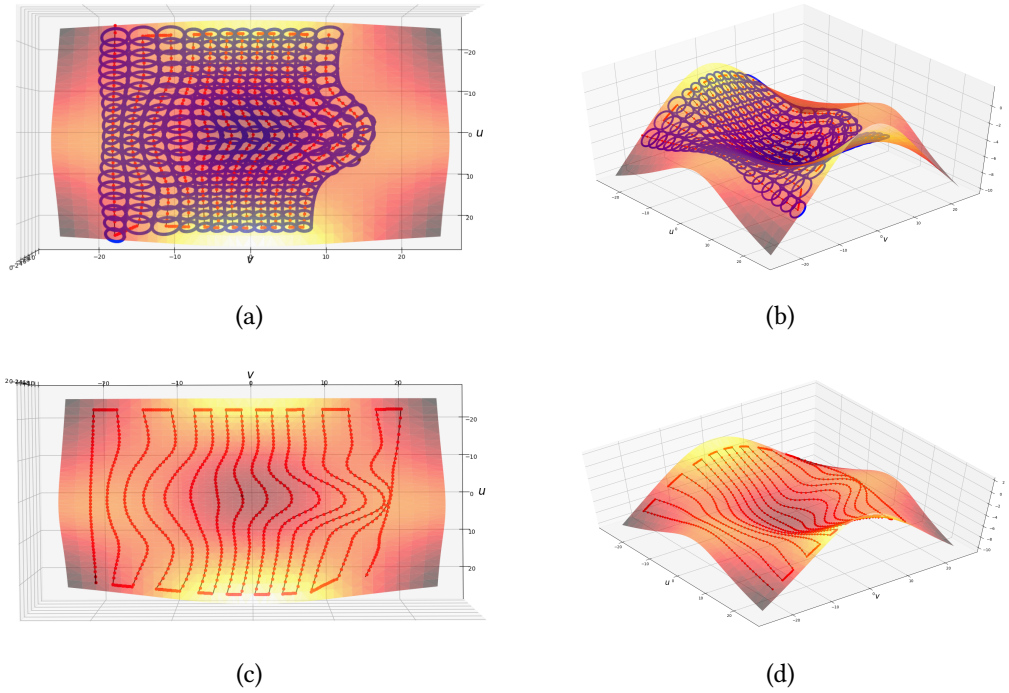
4.3.2 Trajectory Generation

Once the locations P_i of all effective contact areas have been found, a trajectory can be generated by fitting a splines to these points. Cross-over splines are used to connect consecutive paths and are alternated to form the classic scanning trajectory. These splines can then be used to find evenly-spaced way points which comprise the trajectory.

The Euclidean position and surface normal is computed for each waypoint using the NURBS-python library [39]. Using the surface normal and Euclidean position, a homogeneous transformation matrix is formed which describes the pose of the robot end-effector. This pose is then converted to robot joint angles by using a physical description of the robot dimensions and an inverse-kinematics library.

4.3.3 Simulation Studies

Before conducting physical experiments of this trajectory planner, we first tested its functionality using numerical simulation. We first tested the planner on a perfectly smooth NURBS surface with concave and convex regions. In regions of concavity, we assume that the pad makes full contact with the surface in the direction of the concavity. Figure 4.9 shows the results of this simulation.



(a) (b)

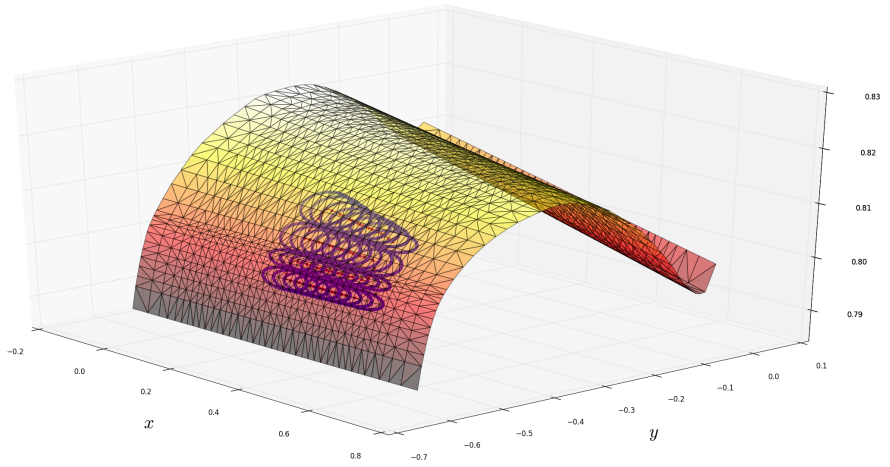
(c) (d)

Figure 4.9: Trajectory planner simulation on perfectly smooth surface with elliptical contact areas (a and b) and trajectory path (c and d)

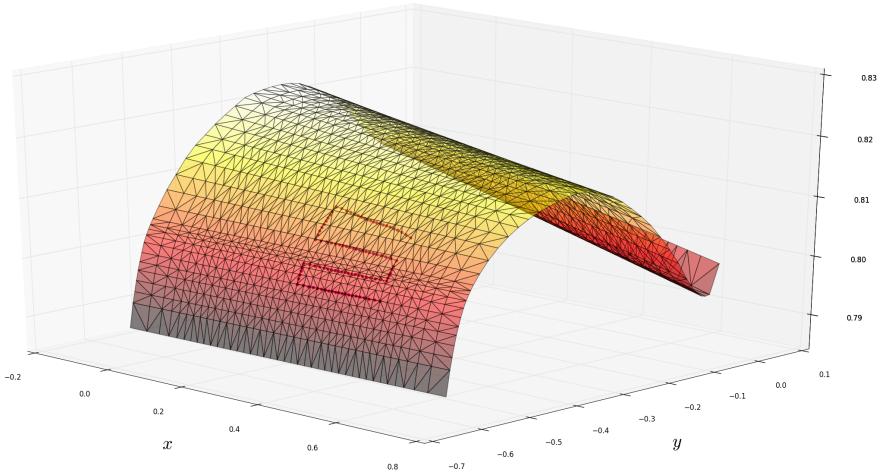
The trajectory planner was then tested on a section of the reconstructed surface which was developed in Chapter 3. Figure 4.10 shows the results of the trajectory planner on the reconstructed surface. In addition to showing that this trajectory planner accounts for change in local curvature, these simulations show that the reconstructed surface produced in Chapter 3 is sufficiently smooth for generating a smooth trajectory.

We can see from these simulations that the trajectory planner developed in this chapter is more time efficient than the conventional straight-line trajectory with equal spacing. Because this trajectory planner takes into account local contact area, the trajectory does not produce significant overlap with areas which have already been sanded.

The final simulation that we perform compares the uniform trajectory developed in this chapter to the two equally-spaced scanning trajectories. The first equally-spaced trajectory uses a spacing of 2 inches, which is also used for comparison in the experiments presented in Chapter



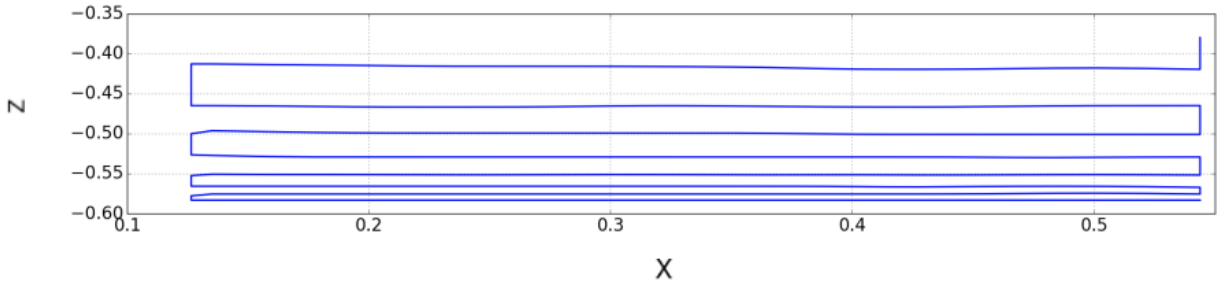
(a)



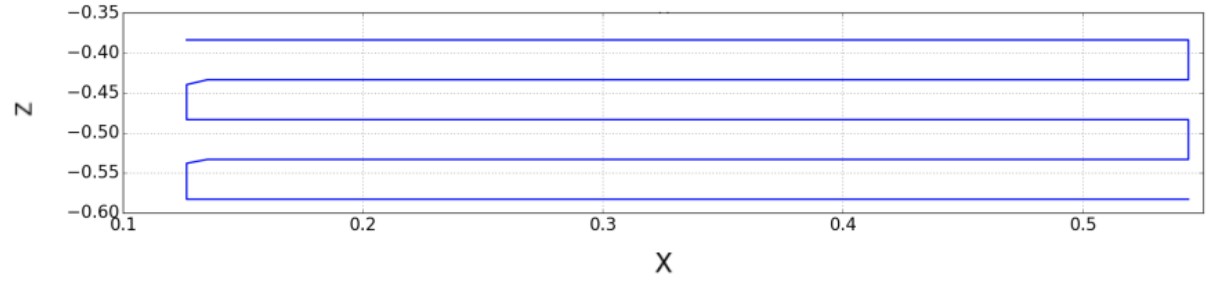
(b)

Figure 4.10: a) Elliptical contact areas on reconstructed surface b) Trajectory path on reconstructed surface

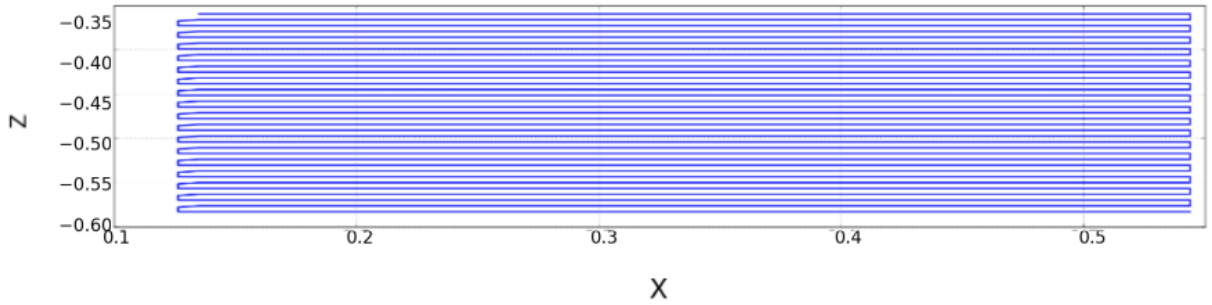
7. The second equally-spaced trajectory uses the minimum spacing required to fully sand the desired area. In order to achieve full-coverage using an equally-spaced trajectory, the spacing between scans must be the minimum spacing between any two consecutive scans in the uniform trajectory. In the simulation shown in Figure 4.11, the minimum spacing from the uniform trajectory was found to be 0.3 inches.



(a)



(b)



(c)

Figure 4.11: a) Uniform coverage trajectory simulation b) Equally-spaced trajectory simulation with 2-inch spacing c) Equally-spaced trajectory using minimal distance to ensure full coverage

We can see from the above simulation that an equally-spaced trajectory takes much longer than the uniform trajectory to fully cover the same amount of area. Measurements of the trajectory lengths confirm this result. The standard trajectory is 12.28m in length, whereas the uniform trajectory is only 3.98m in length. Thus assuming the same velocity is used for both trajectories,

it would take the uniform trajectory one third of the time required by the standard trajectory to sand the same area.

In this chapter we focus on the control strategy which is used to control the motion of the end-effector and normal force applied to the workpiece surface. In Section 5.1, we present the hybrid position-force controller used for the robotic sanding performed in this thesis. In Section 5.2, we provide stability analyses for both the position and force controllers. Finally, in Section 5.3, we describe some considerations which must be given for implementation of the hybrid position-force controller.

5.1 Hybrid Position-Force Controller Design

This section describes the controller that was implemented to control the motion of the robot end-effector as well as the force that the end-effector applies to the surface of the workpiece. In the first subsection, the working principle of the controller is described and a justification is provided as to why this controller was chosen. In the following subsection, the design of the position controller and force controller are described in more detail.

5.1.1 Working Principle

This subsection provides a brief overview of the working principle of the hybrid position-force controller used to control the robot motion and applied normal force. This subsection also discusses why kinematic control was selected as opposed to dynamic control, which is another form of control commonly used with robotic manipulators.

5.1.1.1 Kinematic Control

The type of force controller most commonly found in literature is based on dynamic control, which requires a dynamic model of the robot. The computed torque method is one such method which uses torque as the control input. Most industrial robots, however, only provide direct access to joint position and joint velocity, making it impossible to work with these types of models.

Because of these limitations, the controller must be based on the kinematics of the robot. The

control input is constrained by which inputs can be read by a particular robot. The ABB IRB 4600 robot used in this thesis accepts both joint angles and joint velocities as inputs. Thus, the kinematic control law that is used may either use joint angles or joint velocities as the control input.

A control law with joint velocities as the control input is preferable to a control law which uses joint positions. First, the signal delay from the time a command is sent to the time the command is executed is only 8-16 ms for joint velocity in comparison to 170 ms for joint position [48]. The second reason why joint velocity is preferable for the control input is that it provides for better performance stability for force control. Below is the simplified impedance contact model which does not depend on joint position

$$F = m\ddot{p} + c\dot{p} \tag{5.1}$$

where F is the force vector, m is the virtual mass, b is the virtual damping, and $\ddot{p}$ and $\dot{p}$ are the generalized acceleration and velocity respectively. If the force is the input to the system and position is the output, then the following transfer function is found

$$G(s) = \frac{1/c}{(m/c)s + 1} \tag{5.2}$$

The above transfer function matches the transfer function of a low-pass filter. The output of a low-pass filter is resistant to high frequencies that may be found in the input, which makes this type of control law well-suited for force control.

5.1.1.2 Hybrid Control Principle

The second main problem that must be addressed with designing a hybrid position-force controller is how to design the controller such that the motion and applied force are controlled separately. This problem is addressed by first considering natural constraints and artificial constraints. Natural constraints are constraints to the system which are imposed by the environment. In the case of sanding a workpiece, a natural constraint is the workpiece surface, which constrains the

motion in the normal direction. An artificial constraint is a constraint applied to the system by the control strategy [49].

In the absence of a controller, the motion of a sanding tool on top of the surface is constrained by the surface itself. The sanding tool is free to move tangential to the surface, but is constrained in the normal direction. This natural constraint exists at every location on the surface. The natural constraint of the surface and the orthogonality of the tangent and normal directions are two good justifications for using a frame attached to the surface as the reference frame for the control law to be developed. The force control acts in the normal direction and the motion control acts in the tangential direction. Figure 5.1 below shows the reference frame attached to the workpiece which is used in the controller.

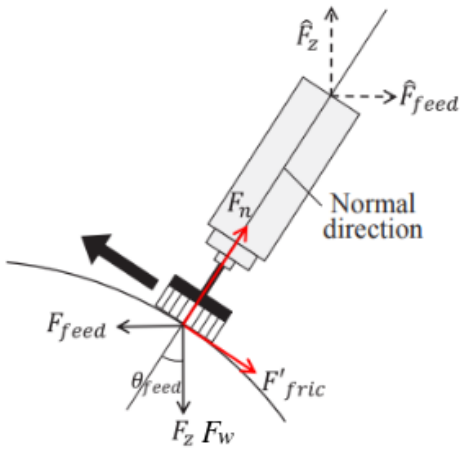


Figure 5.1: Position of sander on surface and direction of normal force

The second reason for using a reference frame attached to the workpiece is that this frame is a moving frame depending on where the tool is located on the surface. The frame must move to adjust to the local concavity and location with respect to the world frame so that the tool may move across the surface while remaining normal to the surface at all times.

5.1.2 Controller Design

In this subsection we discuss the design of the hybrid position-force controller used for this robotic sanding system. First, the position control and force control are discussed separately. These controllers are then combined in parallel to produce simultaneous position-force control, which is discussed in the latter portion of the subsection.

5.1.2.1 Pose Control

Here we provide the control law which is used to control the pose of the robot end-effector. The end-effector pose is given by six degrees of freedom, three rotations and three translations. These may be represented in the commonly used transformation matrix T. Assume the initial and desired transformations are respectively denoted as T_i and T_d, which are composed of a rotation and translation pair (R_i, p_i). Using product of exponentials, we have

$$Rot(\hat{\omega}, \theta) = e^{[\hat{\omega}]\theta_e} = R_i R_f^T \tag{5.3}$$

where $(\hat{\omega}, \theta)$ are the exponential coordinates of the rotation matrix $R_i R_f^T$ and $[\hat{\omega}]$ is the skew symmetric representation of the vector $\hat{\omega}$. The end-effector velocity is represented by a minimum set of coordinates $V_{ee} = [\omega, v]^T$. Proportional control may be used to regulate the robot from the initial pose T_i to the desired pose T_d. The proportional pose control law can then be written as [50]

$$\dot{\theta} = J_b^{-1} \begin{bmatrix} -K_v(p_i - p_d) - K_i \int_0^\tau (p_i - p_d) d\tau \\ -K_v^{'}(\hat{\omega}\theta_e) - K_i^{'} \int_0^\tau \hat{\omega}\theta_e d\tau \end{bmatrix} \tag{5.4}$$

where $K_v, K_v^{'}$ are controller gains. The matrix log, which is the inverse of the matrix exponential, is used to compute $[\hat{\omega}]\theta$ from the rotation matrix R which belongs to the $SO(3)$ lie group.

5.1.2.2 Normal Force Control

Here we introduce the control law which is used to regulate the normal force applied by the sander to a desired force level. It is assumed that the contact model between the sander and the workpiece surface is compliant. Under the compliant model, the contact force is not directly controlled. The force is regulated by using the generalized spring/damper model to convert a difference in force to a positional change of the end-effector. A generalized spring model [51] converts the force error F_e to position error according to

$$F_e = K_{fe}(P_\delta(k+1) - P_\delta(k)) = K_{fe}P_\delta \tag{5.5}$$

where K_{fe} is the stiffness matrix, $P_\delta(k)$ is the end-effector Cartesian position at time k, and $P_\delta(k+1)$ is the end-effector Cartesian position at time $k+1$. The difference in position is fed into the pose tracking loop where it is converted to joint velocity.

5.1.2.3 Simultaneous Position-Force Control

Here we present the development of simultaneous position-force control which may be employed in sanding of free-form surfaces. The controller structure consists of an inner-loop which tracks the pose of the robot end-effector and an outer-loop which regulates the normal force which is applied to the workpiece. An error F_e between the set force F_s and actual force F_r is first converted to an end-effector position change P_δ. This position change is converted to a translational velocity using a proportional gain K_p. The velocity change due to the error in force is added to the pose-tracking velocity. Because these velocities are represented in a coordinate frame attached to the workpiece, the velocity due to the error in force is simply added to the z or normal component of this frame.

The pose tracking portion of the controller converts an error in end-effector position to end-effector velocity using a proportional gain. To maintain a constant feed rate, the velocities from the force and positional errors are summed with a feed-forward velocity which corresponds to the desired feedrate. Using this method of velocity control, the end-effector will traverse the provided

trajectory at a near-constant feedrate, allowing an even amount of material to be removed. The block diagram in Figure 5.2 below shows how the normal force and pose tracking control loops are combined to provide simultaneous position-force control.

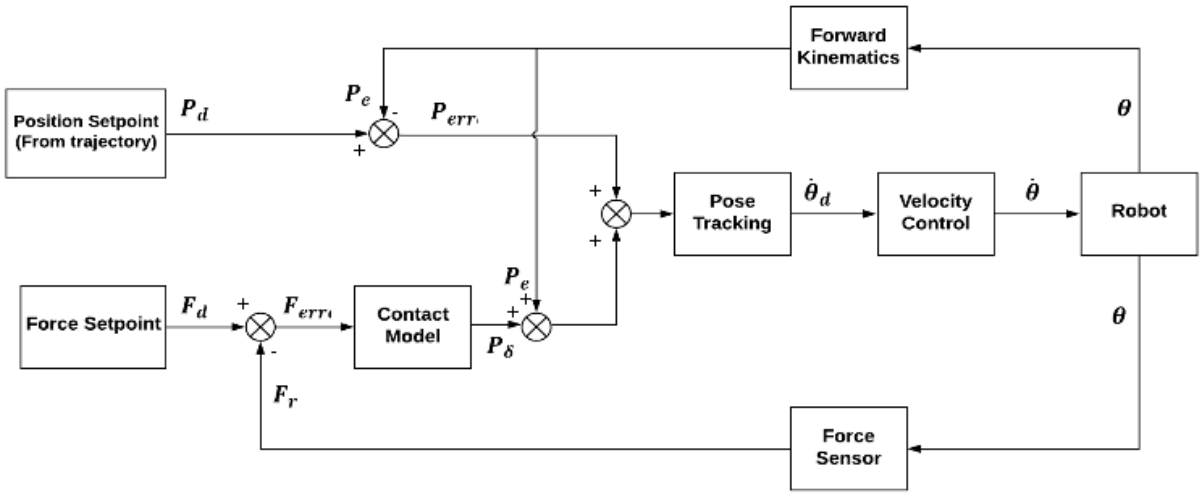


Figure 5.2: Block diagram of hybrid position/force controller

5.2 Stability Analysis

In this section we verify the stability of both the pose tracking controller and normal force controller. For the pose tracking controller stability analysis, we only prove stability for rotation, as the stability proof for translation is straightforward.

5.2.1 Pose Tracking Stability Analysis

Let us first define the rotation error e_r to be

$$e_r = \ln R^{-1} R_d \tag{5.6}$$

where R is the current rotation matrix for the end-effector frame expressed in the robot base frame and R_d is the desired end-effector frame expressed in the robot base frame. We differentiate

the rotation error to obtain the error dynamics

$$\dot{e}_r = \frac{1}{R^{-1}R_d}((\dot{R}^T)(R_d) + R^T(\dot{R}_d)) \tag{5.7}$$

Since R_d is assumed to be constant, the above equation simplifies to

$$\dot{e}_r = \frac{1}{R^{-1}R_d}((\dot{R}^T)(R_d)) \tag{5.8}$$

The derivative of the rotation matrix R is

$$\dot{R} = \omega_s \times R = [\omega_s]R \tag{5.9}$$

where ω_s is the angular velocity of the end-effector expressed in the robot base frame. When a proportional control is used, we can find the angular velocity ω_s using Equation 5.3

$$[\omega_s] = -k[\hat{\omega}]\theta_e = -k \ln R R_d^T \tag{5.10}$$

where k is the proportional gain and $[\omega]$ is the skew-symmetric representation of ω. Using equation 5.9, we have

$$\dot{R}^T = -R^T k \ln R^T R_d \tag{5.11}$$

Substitution of equation 5.11 into equation 5.8 yields

$$\dot{e}_r = -\frac{RR^T}{R_d} k \ln R^T R_d R_d = -k \ln R^T R_d = -k \ln R^{-1} R_d = -k e_r \tag{5.12}$$

which provides exponential convergence of the error $e_r(t)$.

5.2.2 Force Control Stability Analysis

If we assume that the contact between the sanding pad and the surface is elastically compliant, we have the following relationship

$$F_r(t) = K_{fe}(P(t) - P_o) \tag{5.13}$$

where $F_r(t)$ is the estimated force, $P(t)$ is the end-effector position in the robot base frame, P_o is the position at which no force is felt by the end-effector, and K_{fe} is the stiffness matrix. We can rearrange equation 5.13 to get

$$\dot{P}(t) = \frac{\dot{F}_r}{K_{fe}} + \dot{P}_o \tag{5.14}$$

Let F_d be the desired force, F_e be the force error, K_p be the proportional gain, and $P_d(t)$ be the desired end-effector position in the robot base frame. The proportional control law is given by

$$\dot{P}_d(t) = -K_p F_e = -K_p(F_r - F_d) \tag{5.15}$$

If we define $P_e = P(t) - P_d(t)$ as the end-effector error, we have

$$\dot{P}_e(t) = \dot{P}(t) - \dot{P}_d(t) \tag{5.16}$$

$$\dot{P}_d(t) = \frac{\dot{F}_r}{K_{fe}} + \dot{P}_o - \dot{P}_e(t) \tag{5.17}$$

Substituting equation 5.16 into equation 5.15 produces

$$\frac{\dot{F}_r}{K_{fe}} + \dot{P}_o - \dot{P}_e(t) = -K_p F_e \tag{5.18}$$

$$\dot{F}_r + K_{fe}K_p F_e = K_{fe}(\dot{P}_e(t) - \dot{P}_o) \tag{5.19}$$

Assuming that the force set point F_d is constant, it immediately follows that $\dot{F}_e = \dot{F}_r$. Since P_o is also constant, equation 5.18 becomes

$$\dot{F}_e + K_{fe}K_pF_e = K_{fe}\dot{P}_e \tag{5.20}$$

We see from equation 5.20 that the stability of $\dot{F}_e$ depends on $\dot{P}_e$. We can treat $\dot{P}_e(t)$ as stable if the difference between $\dot{P}(t)$ and $\dot{P}_d(t)$ is also stable.

5.3 Real-Time Implementation Considerations

In this section we explain two problems that must be addressed when the hybrid position/force controller is implemented in real-time. The first problem is gravity compensation, which arises when the sanding-tool is tilted at a steep angle. The second problem is making initial contact on the surface.

5.3.1 Gravity Compensation

Gravity compensation refers to the need to compensate for the mass of the sanding tool so that the force controller still regulates to the desired normal force even when the sanding tool is tilted at an angle. The free body diagram in Figure 5.1 shows the forces acting on the sanding tool during a sanding operation.

When the sanding tool is tilted, the component of the weight acting in the normal direction is less than when the sanding tool is upright. We compensate for gravity by finding the difference between the measured no-load force when the tool is vertical and the measured no-load force when the tool is tilted. The tilted no-load force is given by

$$F_{ee} = R_{ee}^{-1}F_w \tag{5.21}$$

where R_{ee} is the orientation of the robot end-effector and F_w is the force due to the weight of the sanding tool. The force applied to the robot end-effector can be split into the forces acting in the each Euclidean direction of the end-effector frame

$$F_{ee} = \begin{bmatrix} Fx_{ee} \\ Fy_{ee} \\ Fz_{ee} \end{bmatrix} \tag{5.22}$$

The force due to the weight of the tool is subtracted from the force acting in the z direction of the end-effector frame to compensate for the tilt of the tool

$$F_N = F_r - F_s + (Fz_{ee} - F_w) \tag{5.23}$$

where F_N is the compensated force, F_r is the force measured by the force-torque sensor, and F_s is the set force.

5.3.2 Making Initial Contact

Before the sanding tool begins to move in the tangential direction, there must be sufficient force being applied to the surface. To ensure that sufficient force is applied, we added a section to our algorithm which slowly lowers the sanding tool to the surface. Once the sanding tool makes contact, the force will slowly increase until it reaches a predetermined threshold. Once the threshold force is reached, the sanding tool is allowed to begin traversing the trajectory.

6. MODELING OF MATERIAL REMOVAL RATE

In this chapter we discuss material removal models, selection of a particular model for robotic sanding, and experiments conducted to determine the model parameters. In Section 6.1, we provide an overview of material removal models which have been used in literature for sanding applications and some assumptions these models make. In Section 6.2, we present Preston's Model as our model of choice and show the derivation of material removal when the sanding tool travels in a straight line along with a simulation. Finally, in Section6.3, we provide the general procedure used for the material testing used for approximating the material constant.

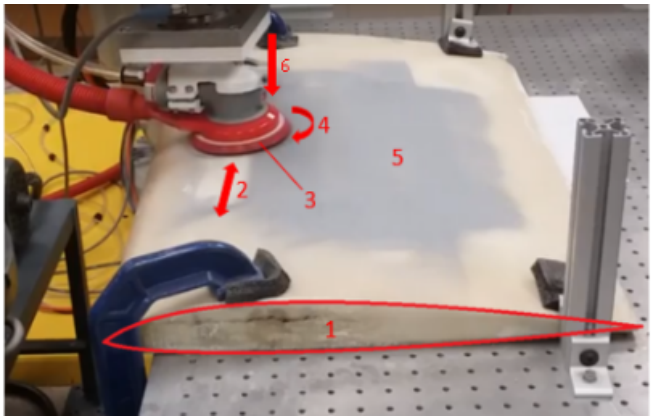
6.1 Overview of Material Removal Models

In this section we provide some background of material removal models specific to material removal in rotary sanding and polishing applications. We examine the models which have been studied in literature and discuss the assumptions and limitations of these models.

6.1.1 Variables Affecting Material Removal

There are numerous factors which affect the rate at which material is removed from a free-form surface during a sanding procedure. Figure 6.1 presents a labeled picture of a typical sanding operation and key factors which may impact the material removal rate during a sanding procedure.

In [27], several factors affecting material removal rate were studied using empirical experimentation to determine which factors had the greatest effect on material removal rate. It was found that the factors that contribute the most to material removal rate are grit value, tool angular velocity, and normal force. Although there may be other factors which affect the material removal rate, these three factors have the greatest impact.



1) Object geometry
2) Translational velocity
3) Sanding pad grit
4) Sander rotational velocity
5) Material being sanded
6) Normal Force

Figure 6.1: Factors which may impact material removal rate

6.1.2 Material Removal Models in Literature

A survey of literature shows two material removal models which are commonly used in computer controlled sanding and polishing procedures. The first model is known as the Preston's model, which is given by [52]

$$f(\boldsymbol{q},t) = \frac{\mathrm{d}z}{\mathrm{d}t} = k \cdot p(\boldsymbol{q},t) \cdot v(\boldsymbol{q},t) \tag{6.1}$$

where $f(\boldsymbol{q},t)$ is the thickness of material removal in unit time, $\boldsymbol{q}$ is the position vector of a given point on the surface, t is time, k is the Preston coefficient which is related to the materials of the polishing tool and surface, $p(\boldsymbol{q},t)$ is the pressure at the polishing point, and $v(\boldsymbol{q},t)$ is the velocity of the tool at the given polishing point.

Many material removal models in literature trace back to Preston's model. Tseng and Wang [53] modified Preston's model to represent the material removal in chemical-mechanical polishing. In [54], Nanz introduced the bending of the pad and flow of slurry into Preston's model.

The second material model commonly found in literature for robotic sanding is Archard's model, which was originally developed to model the wear rate between interlocking parts. Archard's model is given by

$$w(x) = \frac{\mathrm{d}z}{\mathrm{d}x} = k \cdot p(x) \cdot v(x) \cdot H_v^{-1} \cdot V_f^{-1} \tag{6.2}$$

where x is the length that a given point is in contact with the pad, $w(x)$ is the volume of material removed during the apparent contact length, k is a material constant, $p(x)$ is the pressure at a given location along the contact length, $v(x)$ is the velocity of the tool at the contact point due to rotation, H_v is the material hardness, and V_f is the feed rate of the sander.

Unlike Preston's model which models the depth of material removed, Archard's equation models the volume of the material removed. In [55], Archard's model is used to determine the amount of material removed in a steel mould polishing process. In [11], Archard's model was used to determine the profile of material removal using a small polishing tool, assuming Hertzian contact.

The main difference between these two models is the variable of integration. Preston's model integrates over the duration of time that a point on the surface is in contact with the sanding pad, whereas Archard's model integrates over the apparent length that a point is in contact with the pad. The diagram in Figure 6.2 shows a circular contact area with important dimensions. The point of interest P is offset by y_p from the center and the apparent contact length L (used in Archard's model) is the chord which intersects this point. The duration of contact (used in Preston's model) is the time that the sanding pad is in contact with the given point

We see in the succeeding section that if the same assumptions regarding the pressure distribution and velocity are made for each model, they predict the same material removal profile.

6.2 Material Removal for Straight Path

In this section we provide a short derivation of how these models are used to model the amount of material removed in a simple straight-line trajectory. We first make some assumptions regarding the pressure distribution and velocity profiles observed in real life, then integrate these models over their respective variables of integration to find the material removal profile. A comparison of the integrals used for each of these models shows that given the same set of

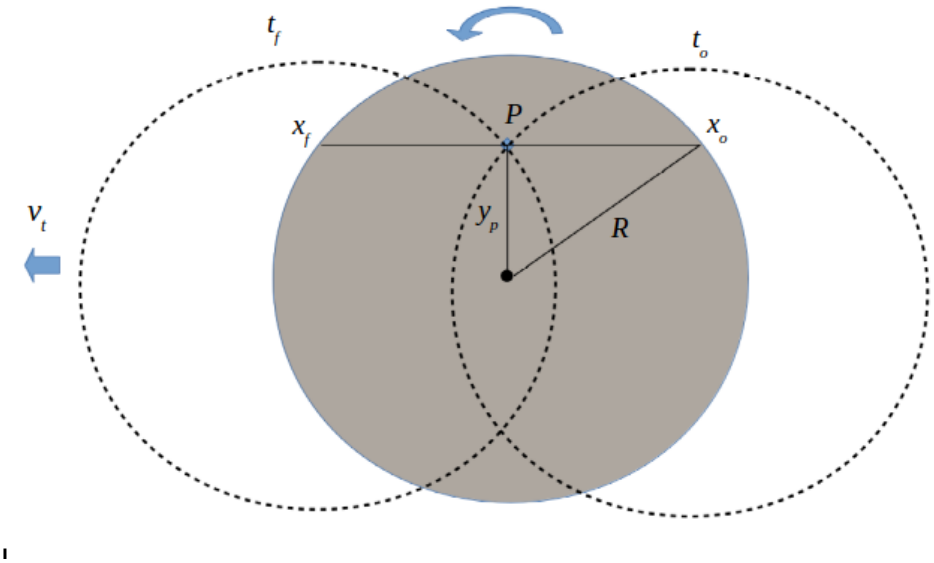


Figure 6.2: Diagram of contact area for straight-line path and dimensions

assumptions, the same material removal profile is predicted.

6.2.1 Model Assumptions

Both Preston's model and Archard's model require assumptions to be made regarding the velocity of the abrasive material and the stress distribution on the surface. The time-varying velocity of the abrasive material at a given point P on the surface may be found for any time t during contact by multiplying the angular velocity of the sander ω by the radial distance r to P.

$$\boldsymbol{v}(\boldsymbol{q}, t) = \boldsymbol{\omega} \times \boldsymbol{r} \tag{6.3}$$

where $\boldsymbol{q}$ is the vector from the origin to point P.

The pressure p at an arbitrary point is much more difficult to estimate. In the field of contact mechanics, the pressure distribution across a contact area may be assumed to follow the Hertzian contact model [28] if the contact area is small relative to the bodies in contact and if the objects making contact are spherical in shape. In sanding processes, the pads used with orbital sanders are quite large and are flat in shape. Furthermore, the contact area that the pad makes with the surface constitutes a large percentage of the total area of the sanding pad. Thus, there is no reason

to believe that the Hertzian model is a good fit for this scenario.

In [56], it is assumed that contact pressure is evenly distributed over the contact area. We make the same assumption, because measuring the true pressure distribution is difficult and this assumption simplifies the model.

6.2.2 Application of Models

To find the total material removed at a given point P during a straight-line sanding path, we integrate each model with respect to their variables of integration. Starting with Preston's model, we begin by writing Preston's model in equation 6.1 in integral form

$$z(x, y, t) = \int_0^t k \cdot \frac{p(\tau)}{2\pi R^2} \cdot v(\boldsymbol{q}, \tau) d\tau \tag{6.4}$$

Assuming that pressure $p(t)$ is constant, it can be taken out of the integral, leaving only velocity inside

$$z(x, y, t) = k \cdot \frac{p}{2\pi R^2} \cdot \int_0^t v(\boldsymbol{q}, \tau) d\tau \tag{6.5}$$

The velocity and integral bounds are written in terms of known quantities

$$z(x, y, t) = k \cdot \frac{p}{2\pi R^2} \cdot \int_{-\frac{\sqrt{r^2 - y_p^2}}{v_t}}^{\frac{\sqrt{r^2 - y_p^2}}{v_t}} \omega_t \cdot \sqrt{((v_t \tau)^2 + y_p^2} d\tau \tag{6.6}$$

The result after integration is

$$z(x, y, t) = k \cdot \frac{p}{2\pi R^2} \cdot \frac{\omega_t}{v_t} \cdot [R\sqrt{R^2 - y_p^2} + y_p^2 (\ln\Big(R + \sqrt{R^2 - y_p^2}\Big) - \ln(|y_p|)] \tag{6.7}$$

Notice that the material removal is directly proportional to the material constant k, the pressure p, and the rotational velocity ω_t. We combine all of the constant quantities into a single quantity I

$$I = k \cdot \frac{p}{2\pi R^2} \cdot \frac{\omega_t}{v_t} \tag{6.8}$$

A similar analysis can be performed using Archard's model. In [11], Archard's model shown in equation 6.2 is converted to the following form

$$w_{h/x} = \frac{dh}{dx} = k\frac{p(x)v(x)}{H_v V_f} \tag{6.9}$$

where h is the depth of material removed, x is the length of contact, $p(x)$ is the pressure along the length of contact, $v(x)$ is the velocity of the abrasive across the length of contact, H_v is the hardness of the surface, and V_f is the feedrate of the sanding tool.

If we assume that the stress distribution is uniform and compute the velocity as was done for Preston's model, we obtain the following integral

$$h(x) = \frac{kp}{H_v V_f}\int_{x_o}^{x_f} \omega_t \sqrt{x^2 + y_p^2}dx \tag{6.10}$$

When we integrate Equation 6.11, we obtain

$$h(x) = \frac{kp}{2H_v V_f}[y_p\sqrt{x^2 + y_p^2} + y_p^2(\ln|x + \sqrt{x^2 + y_p^2}|)]\Bigg|_{x_o}^{x_f} \tag{6.11}$$

which is of the same form as Equation 6.7. Thus, we can conclude that Preston's model and Archard's model produce the same removal profile given the same set up assumptions regarding stress distribution and velocity.

6.3 Model Simulation

In this section we perform a preliminary simulation of these material removal models. Because the same assumptions are used for each model, these simulations are representative of both models. Figure 6.3 shows the material removal profile for a straight-line path corresponding to circular and elliptical contact areas. We varied the constant I defined in equation 6.8 to show how the material profile changes when applied force, rotational speed, or sandpaper roughness are varied.

Using the material removal profile for a single straight-line pass, the profile of multiple passes

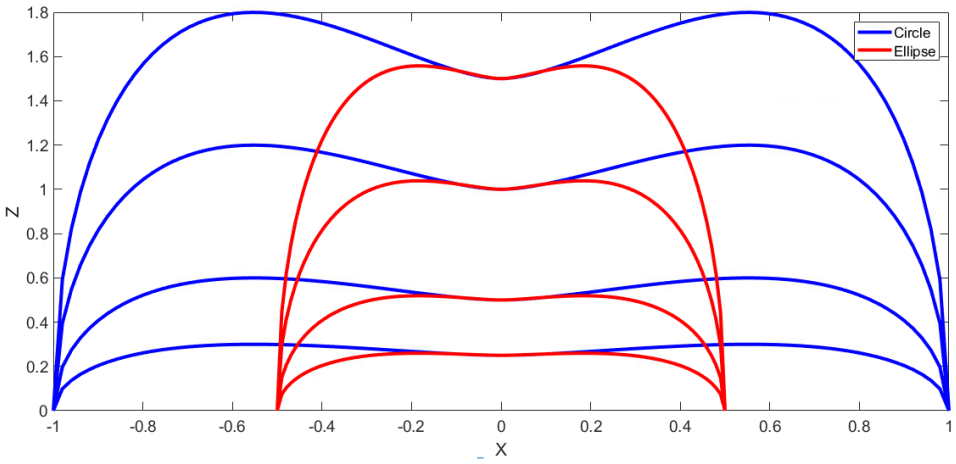


Figure 6.3: Material removal profiles for circular and elliptical contact areas

can be estimated by shifting the profile and adding it via superposition. The distance by which the profile is shifted is used to compute the overlap used in the trajectory planner. Figure 6.4 shows the superposition of two straight-line material removal profiles

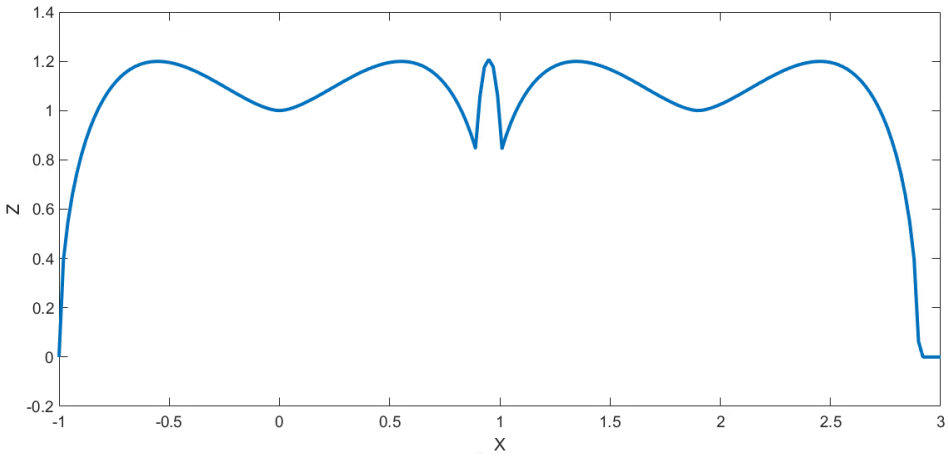


Figure 6.4: Superposed material removal profile for consecutive scans

6.4 Material Model Empirical Testing

In this section we provide an overview of the empirical testing which must be performed to develop a material removal model for a robotic sanding operation. While the applied force and angular velocity can be measured using sensors, the material constant k requires several empirical trials to determine.

There are several considerations which must be taken into account when designing the trials used for collecting empirical data. One important factor is the type of material being sanded. For these experiments, we used a commonly used putty used for repairing dents in automobile and aircraft surfaces (3M Platinum Plus).

Another important design consideration is how the variables will be varied throughout the trials. According to Preston's and Archard's models presented above, the applied force, rotational velocity, and material constant k are the three factors that affect the material removal. The main factor which impacts the material removal constant is the grit of the sand paper used for sanding.

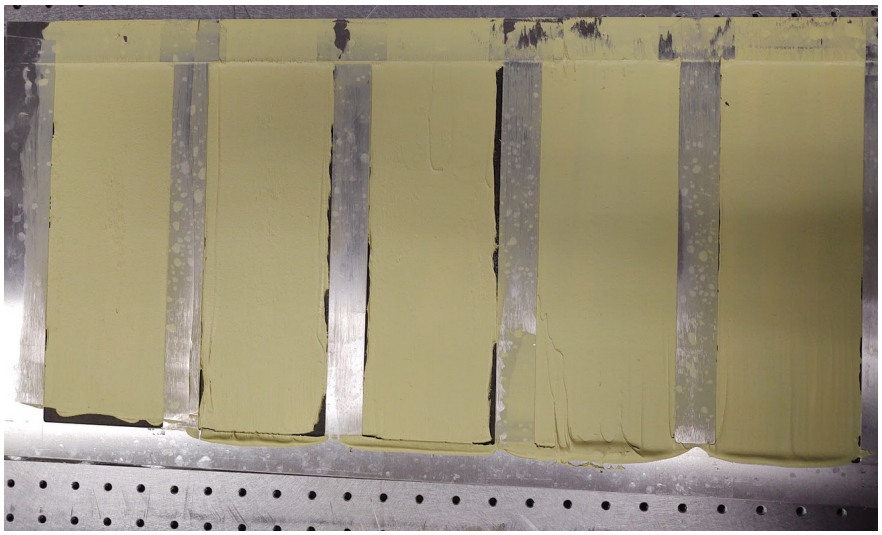
Because it is not known how the grit of sand paper affects the material constant, one way to design the trials is to conduct several trials of varying forces and velocities for different grits of sand paper. Table 6.1 below shows a possible design of experiments using these variables

Grit	Force (N)	Velocity (RPM)
80	15	1000
120	20	2000
150	25	3000
180	30	4000

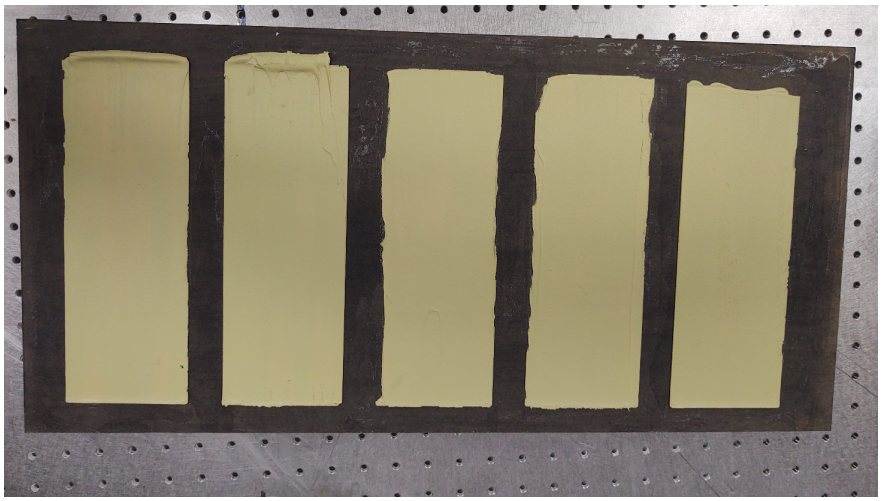
Table 6.1: Levels of variables to be tested.

To measure the effect of these variables, we want to control for the geometry of the surface being sanded, the material being sanded, and the thickness of the material. The strategy was to use the same type of test article.

The test article is comprised of five even layers of putty on a particle board surface. The putty layers are formed by using a sheet metal template of predetermined thickness. Figure 6.5 below shows the process by which the test articles are constructed



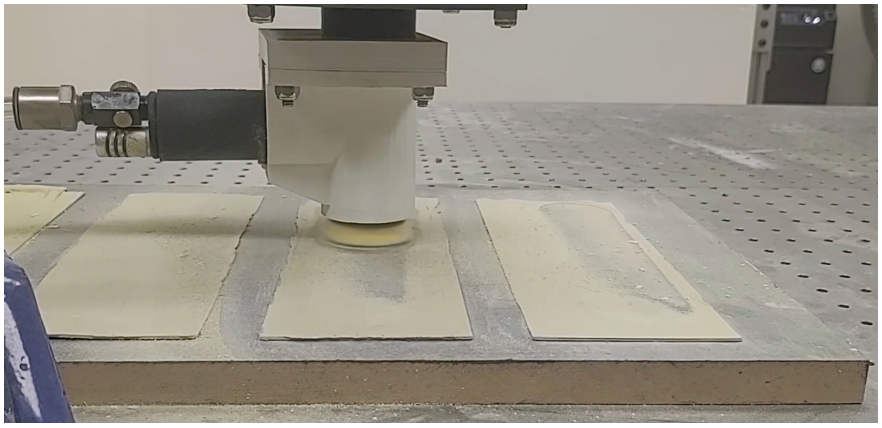
(a)



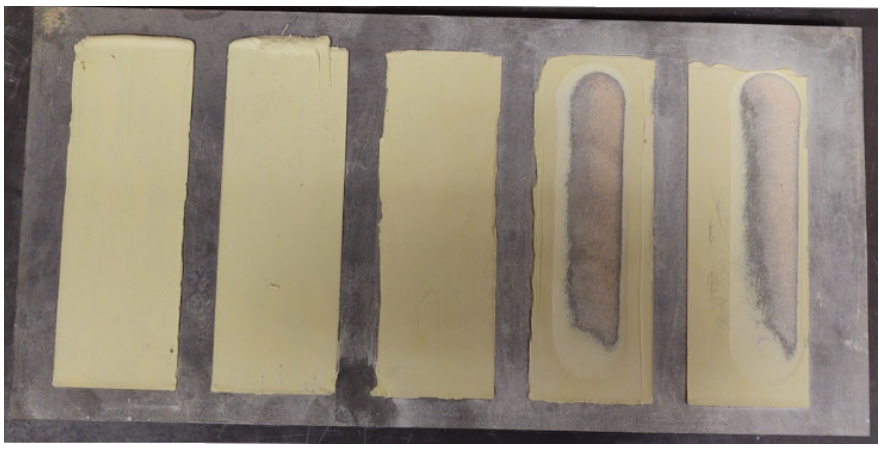
(b)

Figure 6.5: a) Putty is applied to surface with even thickness using template b) Test article with template removed

The material testing involves making several passes over each test article using the values specified in Table 6.1. Once the layer of putty is completely sanded, the thickness of the putty is divided by the number of passes to obtain the average depth of material removed each pass. Figure 6.6 shows the sanding of the test article and the test article after the trial is complete.



(a)



(b)

Figure 6.6: a) Sanding a test article b) Completed test articles

Although these trials are not yet complete, we hope to soon finish these trials and use the results to develop a material model for predicting the amount of putty removed for a given applied force and sander rotational velocity.

In this chapter we perform some experiments which serve to validate the smoothness of the surface model developed in Chapter 3 and uniform coverage trajectory planner developed in Chapter 4. In Section 7.1, we compare the coverage of uniform coverage and equally-spaced trajectories. In Section 7.2 , we provide the results of some related work, which could also be used for proving the effectiveness of the uniform coverage trajectory planner.

7.1 Testing Uniform Coverage Planner

In this section we present the results of an experiment which was conducted using the uniform trajectory planner developed in Chapter 4. In this experiment, we compare the coverage of the uniform trajectory planner to the coverage of a trajectory with equal spacing between scans.

7.1.1 Force and Position Measurements

The experimental setup consists of a helicopter blade section which will be sanded using both trajectories. The sander used for performing these experiments is a hand-held pneumatic sander with a 3-inch pad, shown in Figure 7.1

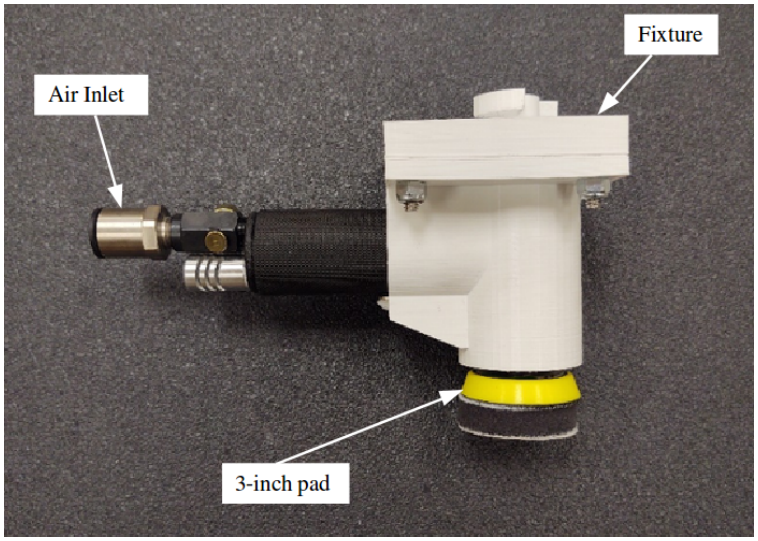


Figure 7.1: Sander with 3-inch pad used for experimentation

Before executing each trajectory, the blade section is first covered with a thin layer of 3M Platinum Plus putty, so that the area of sanding is more visible. We first sand the helicopter blade using the uniform coverage trajectory developed in Chapter 4. The end-effector position and normal force were measured during execution of the trajectory and are shown in Figure 7.2. We also provide the uniform coverage trajectory simulation which was performed in Chapter 4 for comparison

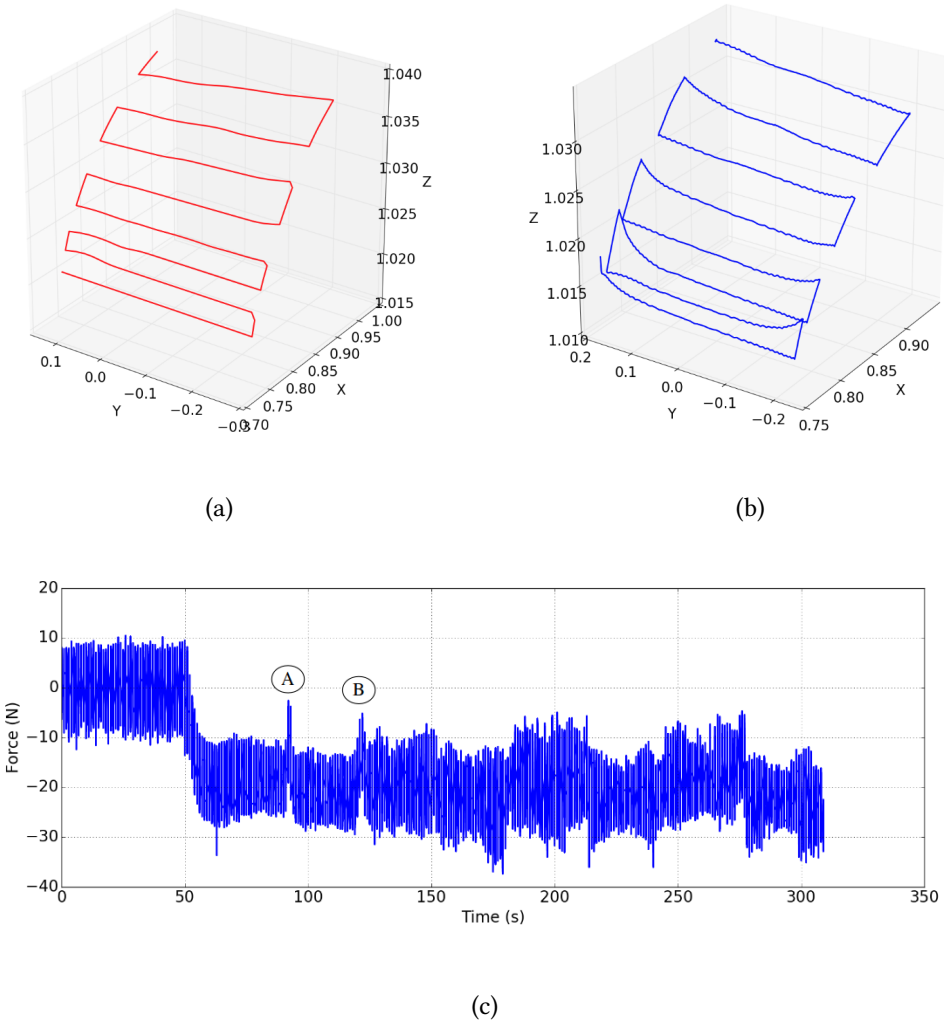
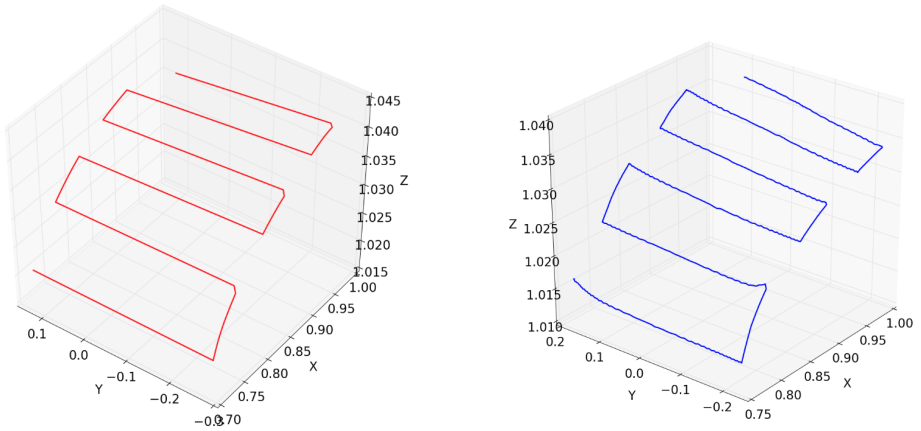


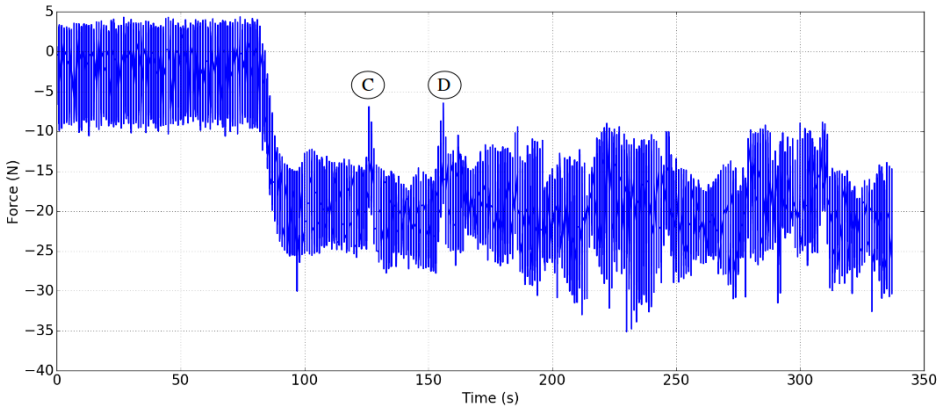
Figure 7.2: a) Uniform coverage trajectory simulation b) Uniform coverage trajectory empirical end-effector position c) Uniform coverage trajectory normal force

The points A and B in Figure 7.2 are two instances when the uniform trajectory planner deviated from the true surface. The deviations in the trajectory result in slight fluctuations in normal force. Two possible reasons for these deviations are poor surface curvature estimation during trajectory planning or uneven trajectory way-point distribution.

We then sand the helicopter blade section using an equally-spaced scanning trajectory. We choose the spacing between scans to be a reasonable distance which may be naively guessed based on the diameter of the sanding pad. If the surface being sanded were flat, a spacing of approximately 3 inches would ensure full coverage. Because the surface is curved, we choose a trajectory spacing of 2 inches. The end-effector position and normal force were once again measured during execution of the trajectory and are shown in Figure 7.3. As before, we provide the simulation of the equally-spaced trajectory performed in Chapter 4 for comparison



(a) (b)



(c)

Figure 7.3: a) Equally-spaced trajectory simulation b) Equally-spaced trajectory empirical end-effector position c) Equally-spaced trajectory normal force

The points C and D in Figure 7.3 are two instances when the equally-spaced trajectory deviated slightly from the surface. Because the equally-spaced trajectory does not require estimations of surface curvature, it is likely that the root cause of these deviations is inaccuracy in the surface model produced in Chapter 3.

7.1.2 Coverage Comparison

A comparison of the coverage of each trajectory shows the effectiveness of the uniform trajectory planner. Figure 7.4 shows the putty applied to the blade prior to any sanding. Figure 7.5 shows the helicopter blade after execution of the uniform-coverage and equally-spaced trajectories. Both trajectories began on the curved portion of the blade and ended on the flat portion. The points A, B, C and D correspond to the points shown in the the end-effector position and normal force measurements.

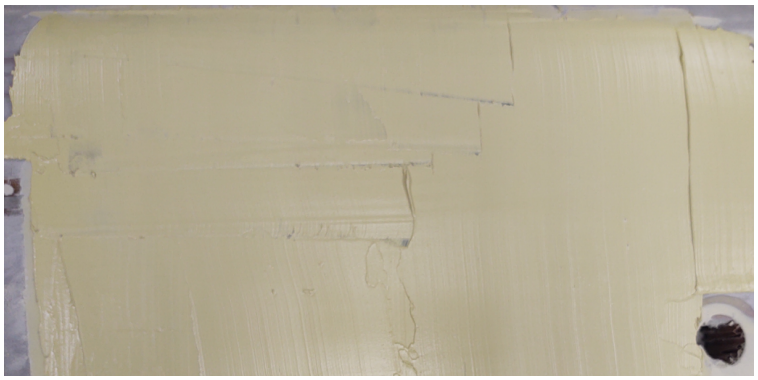
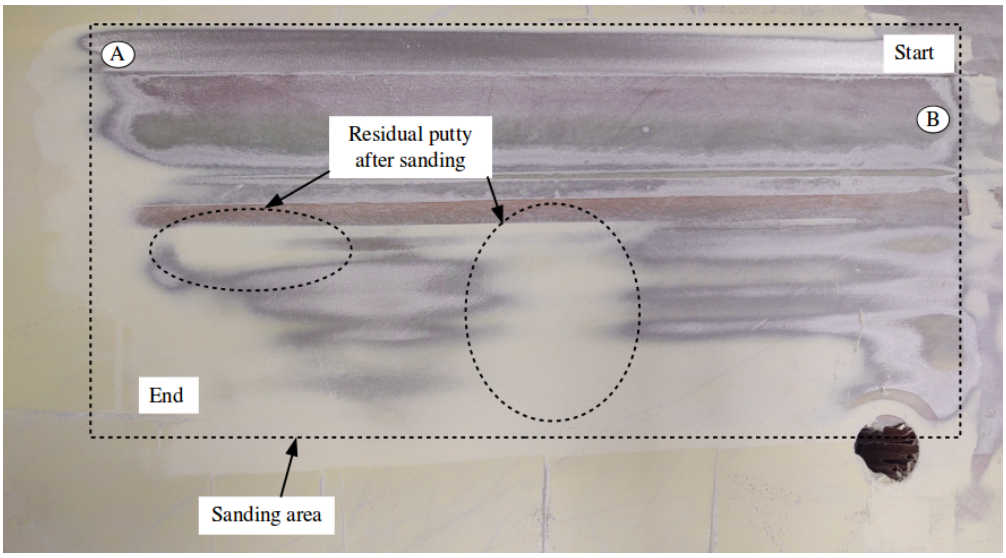
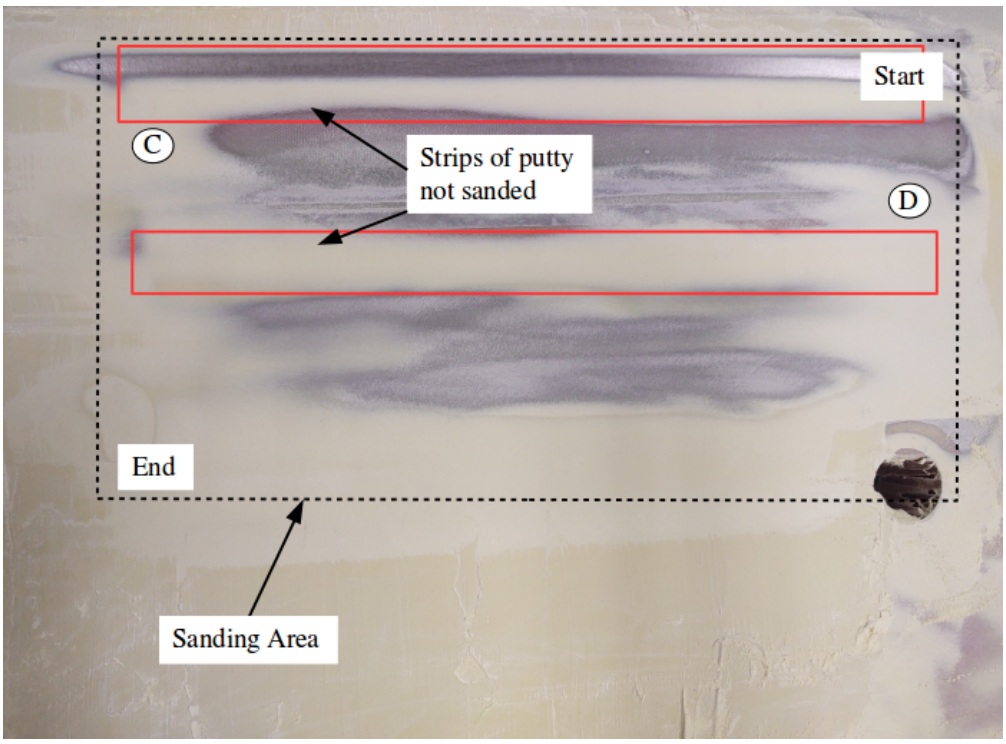


Figure 7.4: Blade with putty before sanding

We see in Figure 7.5 that the uniform trajectory planner provides much better coverage, especially in the region of higher curvature. The area of sanding has been indicated in each picture. Because the putty is not easily sanded, both trials shows streaks of residual putty that have been left after sanding. However, because the equally-spaced trajectory does not account for the surface geometry, we observe large strips of putty which are not sanded at all when this trajectory is used.



(a)



(b)

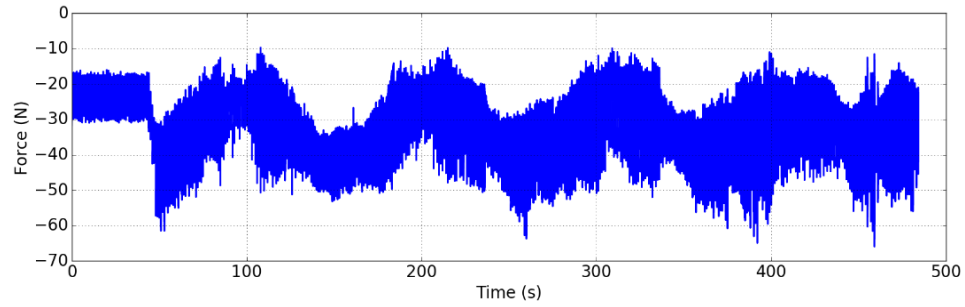
Figure 7.5: a) Coverage of uniform trajectory b) Coverage of equally-spaced trajectory

7.2 Sanding Concave Surfaces

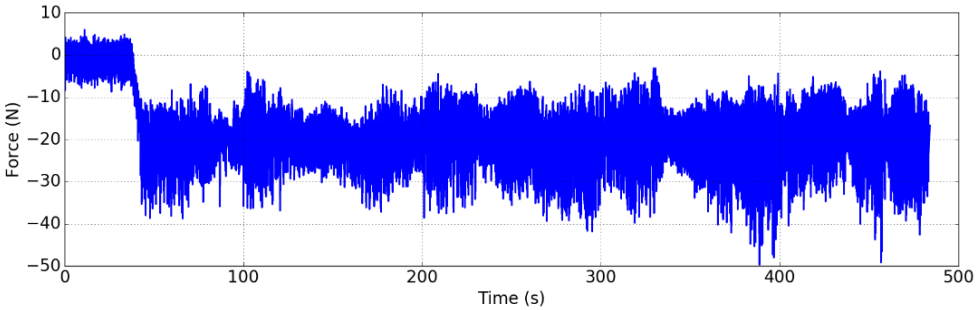
In this section we present the end-effector position and normal force measurements obtained from sanding a concave surface. Because of the shape of this surface, gravity compensation is required to compute the effective normal force. Figure 7.6 shows the concave surface before the sanding operation.

Figure 7.6: Concave surface before sanding

Because portions of this surface are near-vertical, the end-effector must tilt sideways in order to reach these portions. Because the weight of the sanding tool is no longer coincident with the normal force, gravity compensation must be used. Figure 7.7 shows the uncompensated force measured by the load cell and compensated force, respectively.



(a)



(b)

Figure 7.7: a) Concave surface force measured by load cell with no gravity compensation b) Normal force applied to surface after gravity compensation

The end-effector position was also recorded during this sanding operation and is presented in Figure 7.8

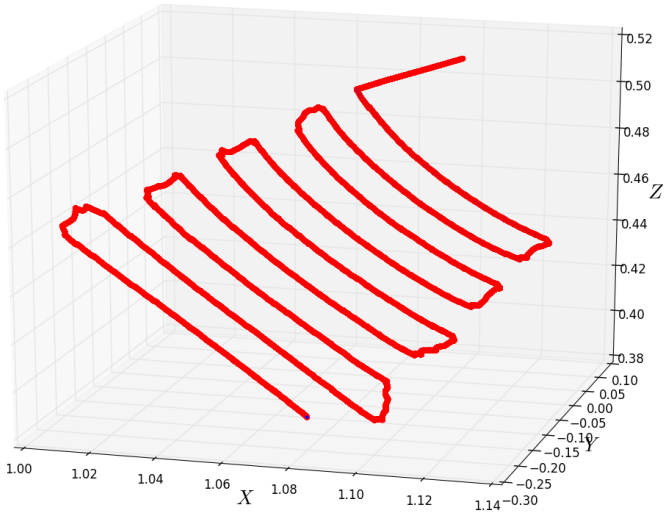


Figure 7.8: End-effector position during sanding of concave surface

Because the trajectory planner used for planning this trajectory did not account for uniform coverage, the sanding operation does not fully cover the surface. Figure 7.9 shows the surface after sanding, with uncovered strips between the scans similar to the ones shown in Section 7.1.

Figure 7.9: Concave surface with uncovered strips between passes

Although we were not able to use the uniform coverage trajectory planner on this surface, it provides another example as to when uniform coverage is needed.

8. CONCLUSIONS AND FUTURE WORK

In this thesis we provided a simple method for reconstructing a free-form surface using a NURBS surface representation which produces a smooth surface that is suitable for trajectory generation. We then developed a method for approximating the local contact area between the sanding pad and surface using differential geometry concepts. Using this contact area approximation, we developed a contact area-based trajectory planner which plans a trajectory for sanding free-form surfaces. The trajectory produced by this trajectory planner is more time-efficient and reduces unnecessary overlap between scans. We then discussed the control used for positioning the robot end-effector and explained why hybrid position/force control is suitable for robotic sanding. We showed that this control method is indeed stable and provided a high-level block diagram of the control scheme we implemented. Finally, we discussed material models which have been used in literature and showed how these models may be applied to robotic sanding. We then used these models to develop a material removal profile for a straight-line path, which can be used to further improve the spacing between scans.

To validate the trajectory planner developed in this thesis, we are currently conducting an experiment which will compare a straight-line trajectory and a contact area-based trajectory sanding the same amount of surface area. We will compare the execution times of these two trajectories and examine the amount of overlap. We are also currently conducting empirical trials to validate the material removal models presented in this paper. Once enough trials are conducted, we can predict the amount of material removed given the rotational speed and applied for used during the sanding operation.

There are several areas of future work that are left to explore regarding many of these topics. The main areas of future work include the following:

- Complete the material removal trials and determine whether the empirical data matches the material models which appear in literature. After developing an empirical model, we hope

to validate the model by using it to predict the amount of material removed and comparing it to real data

- Another area we wish to investigate is the affect of surface geometry or curvature on the stress distribution within the contact area. We would like to understand to what extent curvature lessens the load felt by the surface near the edges of the sanding pad
- We are interested in verifying whether either Preston's model or Archard's model provide the correct material removal profiles for rotary sanding tools. If possible, we would like to somehow empirically measure the the profile after sanding a straight line and compare it to the theoretical profile found in Chapter 6
- Finally, we would like to build upon the trajectory planner and determine how changing the orientation of the paths may affect the trajectory execution time. We would also like to investigate whether there is a way estimate the orientation and starting position of the trajectory in a computationally efficient manner

REFERENCES

[1] K. Shaw, *World Robotics Report: Global Sales of Robots Hit 16.5B in 2018*, 2020 (September 1, 2020). https://www.roboticsbusinessreview.com/research/world-robotics-report-global-sales-of-robots-hit-16-5b-in-2018/.

[2] R. Kala, "2 - basics of autonomous vehicles," in *On-Road Intelligent Vehicles* (R. Kala, ed.), pp. 11 – 35, Butterworth-Heinemann, 2016.

[3] X. Armangué and J. Salvi, "Overall view regarding fundamental matrix estimation," *Image and vision computing*, vol. 21, no. 2, pp. 205–220, 2003.

[4] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "High-quality scanning using time-of-flight depth superresolution," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–7, 2008.

[5] V. L. Tran, H.-Y. Lin, and W. Liu, "A structured light rgb-d camera system for accurate depth measurement," *International Journal of Optics*, 2018.

[6] A. Galvez and A. Iglesias, "Particle swarm optimization for non-uniform rational b-spline surface reconstruction from clouds of 3d data points," *Information Sciences*, vol. 192, pp. 174–192, 2012.

[7] N. J. Mitra and A. Nguyen, "Estimating surface normals in noisy point cloud data," *SoCG*, June 2003.

[8] P. Laurent-Gengoux and M. Mekhilef, "Optimization of a nurbs representation," *Computer-Aided Design*, vol. 25, pp. 699–710, 1993.

[9] H.-Y. Tam, O. Lui, and A. Mok, "Robotic polishing of free-form surfaces using scanning paths," *Journal of Materials Processing Technology*, vol. 95, pp. 191–200, Oct. 1999.

[10] Y. Mizugaki, M. Sakamoto, and T. Sata, "Fractal path generation for a metal-mold polishing robot system and its evaluation by the operability," *CIRP Annals*, vol. 41, no. 1, pp. 531 – 534, 1992.

[11] L. Zhang, H. Y. Tam, C.-M. Yuan, Y.-P. Chen, and Z.-D. Zhou, "An investigation of material removal in polishing with fixed abrasives," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 216, no. 1, pp. 103–112, 2002.

[12] G. A. Korsah, A. T. Stentz, and M. B. Dias, "Continuous-field path planning with constrained path-dependent state variables," May 2008.

[13] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.

[14] F. Ren, Y. Sun, and D. Guo, "Combined reparameterization-based spiral toolpath generation for five-axis sculptured surface machining," *The international journal of advanced manufacturing technology*, vol. 40, no. 7-8, pp. 760–768, 2009.

[15] J. Hess, G. D. Tipaldi, and W. Burgard, "Null space optimization for effective coverage of 3d surfaces using redundant manipulators," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1923–1928, 2012.

[16] B. Siciliano and V. Luigi, "From indirect to direct force control: A roadmap for enhanced industrial robots," 01 2000.

[17] N. Hogan, "Impedance control: An approach to manipulation: Part i—theory," 1985.

[18] C. haur Wu, "Compliance control of a robot manipulator based on joint torque servo," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 55–71, 1985.

[19] T. Winiarski and A. Woźniak, "Indirect force control development procedure," *Robotica*, vol. 31, no. 3, pp. 465–478, 2013.

[20] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," 1981.

[21] F. Tian, C. Lv, Z. Li, and G. Liu, "Modeling and control of robotic automatic polishing for curved surfaces," *CIRP Journal of Manufacturing Science and Technology*, vol. 14, pp. 55 – 64, 2016.

[22] F. Nagata, Y. Kusumoto, Y. Fujimoto, and K. Watanabe, "Robotic sanding system for new designed furniture with free-formed surface," *Robotics and Computer-integrated Manufacturing*, vol. 23, pp. 371–379, 2007.

[23] E. B. Kalt, "Development of an integrated robotic polishing system," Jan 2016.

[24] S. Pedrammehr, B. Danaei, H. Abdi, M. T. Masouleh, and S. Nahavandi, "Dynamic analysis of hexarot: axis-symmetric parallel manipulator," *Robotica*, vol. 36, no. 2, p. 225–240, 2018.

[25] L. Huo and L. Baron, "Kinematic inversion of functionally-redundant serial manipulators: application to arc-welding," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 29, no. 4, pp. 679–690, 2005.

[26] M. Tsai, J. Huang, and W. Kao, "Robotic polishing of precision molds with uniform material removal control," *International Journal of Machine Tools and Manufacture*, vol. 49, no. 11, pp. 885 – 895, 2009.

[27] V. Pandiyan, W. Caesarendra, T. Tjahjowidodo, and G. Praveen, "Predictive modelling and analysis of process parameters on material removal characteristics in abrasive belt grinding process," *Applied Sciences*, vol. 7, no. 4, p. 363, 2017.

[28] A. Roswell, F. J. Xi, and G. Liu, "Modelling and analysis of contact stress for automated polishing," *International Journal of Machine Tools and Manufacture*, vol. 46, no. 3, pp. 424 – 435, 2006.

[29] I. E. Sutherland and G. W. Hodgman, "Reentrant polygon clipping," *Commun. ACM*, vol. 17, p. 32–42, Jan. 1974.

[30] G. Barequet and S. Har-Peled, "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions," *Journal of Algorithms*, vol. 38, no. 1, pp. 91 – 109, 2001.

[31] F. Nagata, T. Hase, Z. Haga, M. Omoto, and K. Watanabe, "Cad/cam-based position/force controller for a mold polishing robot," *Mechatronics*, vol. 17, no. 4, pp. 207 – 216, 2007.

[32] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Object recognition supported by user interaction for service robots*, vol. 3, pp. 545–548, IEEE, 2002.

[33] M. do Carmo, *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Dover Books on Mathematics, Dover Publications, 2016.

[34] H. Khakpour, L. Birglen, and S.-A. Tahan, "Uniform scanning path generation for abrasive waterjet polishing of free-form surfaces modeled by triangulated meshes," *The International Journal of Advanced Manufacturing Technology*, vol. 77, no. 5, pp. 1167 – 1176, 2015.

[35] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 2012.

[36] P. Pal and R. Ballav, "Object shape reconstruction through nurbs surface interpolation," *International Journal of Production Research*, vol. 45, no. 2, pp. 287–307, 2007.

[37] M. E. Flores and M. B. Milam, "Trajectory generation for differentially flat systems via nurbs basis functions with obstacle avoidance," pp. 7 pp.–, 2006.

[38] D. Saini, S. Kumar, and T. R. Gulati, "Nurbs-based geometric inverse reconstruction of free-form shapes," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 1, pp. 116 – 133, 2017.

[39] O. R. Bingol and A. Krishnamurthy, "NURBS-Python: An open-source object-oriented NURBS modeling framework in Python," *SoftwareX*, vol. 9, pp. 85–94, 2019.

[40] L. Piegl and W. Tiller, "Algorithm for approximate nurbs skinning," *Computer-Aided Design*, vol. 28, no. 9, pp. 699 – 706, 1996.

[41] V. Hernández-Mederos and J. Estrada-Sarlabous, "Sampling points on regular parametric curves with control of their distribution," *Computer Aided Geometric Design*, vol. 20, no. 6, pp. 363 – 382, 2003.

[42] R. D. Fuhr and M. Kallay, "Monotone linear rational spline interpolation," *Computer Aided Geometric Design*, vol. 9, no. 4, pp. 313–319, 1992.

[43] Y. Sun, D. Feng, and D. Guo, "An adaptive uniform toolpath generation method for the automatic polishing of complex surfaces with adjustable density," *The International Journal of Advanced Manufacturing Technology*, vol. 80, pp. 1673–1683, Oct 2015.

[44] F. Ren, Y. Sun, and D. Guo, "Combined reparameterization-based spiral toolpath generation for five-axis sculptured surface machining," *The International Journal of Advanced Manufacturing Technology*, vol. 40, pp. 760–768, Feb 2009.

[45] M. Rososhansky and F. J. Xi, "Coverage based tool-path planning for automated polishing using contact mechanics theory," *Journal of Manufacturing Systems*, vol. 30, no. 3, pp. 144 – 153, 2011.

[46] Y. Han, L. Zhang, M. Guo, C. Fan, and F. Liang, "Tool paths generation strategy for polishing of freeform surface with physically uniform coverage," *The International Journal of Advanced Manufacturing Technology*, vol. 95, pp. 2125–2144, Mar 2018.

[47] A. N. Pressley, *Elementary differential geometry*. Springer Science & Business Media, 2010.

[48] T. T. Andersen, "Optimizing the universal robots ros driver.," 2015.

[49] K. Lynch and F. Park, *Modern Robotics*. Cambridge University Press, 2017.

[50] Y. Wen and P. R. Pagilla, "Robotic Surface Finishing of Curved Surfaces: Real-Time Identification of Surface Profile and Control," vol. Volume 3: Manufacturing Equipment and Systems, 06 2018. V003T02A045.

[51] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[52] I. D. Marinescu, E. Uhlmann, and T. Doi, *Handbook of lapping and polishing*. CrC Press, 2006.

[53] W.-T. Tseng and Y.-L. Wang, "Re-examination of pressure and speed dependences of removal rate during chemical-mechanical polishing processes," *Journal of the Electrochemical Society*, vol. 144, no. 2, p. L15, 1997.

[54] G. Nanz and L. E. Camilletti, "Modeling of chemical-mechanical polishing: A review," *IEEE transactions on semiconductor manufacturing*, vol. 8, no. 4, pp. 382–389, 1995.

[55] R. Almeida, R. Börret, W. Rimkus, D. K. Harrison, and A. K. DeSilva, "Material removal simulation for steel mould polishing," *Production & Manufacturing Research*, vol. 5, no. 1, pp. 235–249, 2017.

[56] W. Lin, P. Xu, B. Li, and X. Yang, "Path planning of mechanical polishing process for freeform surface with a small polishing tool," *Robotics and Biomimetics*, vol. 1, no. 1, p. 24, 2014.