Surface Estimation from Multi-modal Tactile Data

by

Isura Thrikawala

B.Sc. Sir John Kotelawala Defence University, 2019

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE FACULTY OF GRADUATE STUDIES
OF LAKEHEAD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Surface Estimation from Multi-modal Tactile Data

by

Isura Thrikawala

Supervisory Committee

Dr. Thiago E. Alves de Oliveira,
Supervisor
(*Department of Computer Science, Lakehead University, Thunder Bay, Ontario, Canada*)

Dr. Yimin Yang,
Internal Examiner
(*Department of Computer Science, Lakehead University, Thunder Bay, Ontario, Canada*)

Dr. Vinicius Prado da Fonseca,
External Examiner
(*Department of Computer Science, Memorial University of Newfoundland, St. John's, Newfoundland and Labrador, Canada*)

ABSTRACT

The increasing popularity of Robotic applications has seen use in healthcare, surgery, and as an industrial tool. These robots are expected to be able to make physical contact with the objects in the environment which allows tasks such as grasping and manipulation, while also allowing to obtain information about the objects such as shape, texture, and hardness. In an ideal world, a complete model of the environment would be known beforehand and robots would not need to explore objects and surfaces since their information would be available in the model of the world. In the real world, most environments are unstructured and robots must be able to operate safely without causing harm to themselves or objects while taking into account environmental uncertainties and building models for the environment and its objects. To overcome this, the trend has been to use computer vision to detect objects in the environment. Although computer vision has seen great advancement in this regard, there are some problems that cannot be solved by using vision alone. Objects that are occluded, transparent, or do not have rich visual features cannot be detected by using vision. It is also impossible to estimate features such as hardness or tactile texture using vision. To this end, we use a bio-inspired tactile sensor consisting of a compliant structure, a MARG sensor, and a pressure sensor along with a robotic manipulator to explore surfaces with the only assumption that the general location of the surface is known. This sensing module allows the robotic manipulator to have a predetermined angle of approach which is essential when exploring unseen surfaces. The robotic manipulator is carefully controlled while probes are made onto the surface while the deformation of the sensing module is used to collect orientation data along the normals of the surface. We calculate control points that define the curvature of the surface in between contact points which allow for fewer probes. We then create an estimate of the surface in 3D using Bezier surfaces. Once an estimate of the surface is obtained, we also demonstrate that the texture of the surface can be explored from the vibrations obtained by sliding the sensing module over the surface.

ACKNOWLEDGEMENT

I would like to give thanks to the following funding sources, for their financial support to my research:

- **Lakehead University Faculty of Graduate Studies**;
- **Lakehead University Faculty of Science and Environmental Studies**;
- **Dr. Thiago Oliveira**;

I would like to express my sincere gratitude to my supervisor, Dr. Thiago Oliveira for his endless support and guidance throughout my research. I would also like to thank my friends and family for their support, encouragement, and love.

Contents

Supervisory Committee ii

Abstract iii

Acknowledgements iv

Table of Contents v

List of Figures vii

1 Introduction 1

1.1 Objectives . . . 3
1.2 Methodology . . . 4
1.3 Thesis Organization . . . 5

2 Background 6

2.1 Introduction . . . 6
2.2 Shape Perception . . . 7
2.2.1 Local Methods . . . 7
2.2.2 Global Methods . . . 10
2.3 Pose Estimation . . . 12
2.4 Texture Recognition . . . 15
2.5 Discussion . . . 17

3 Experimental Setup 19

3.1 Tactile Perception . . . 19
3.2 Tactile Sensing Module . . . 20
3.2.1 Structure . . . 20
3.2.2 MARG Sensor . . . 21
3.2.3 Orientation Representation . . . 22

3.2.4 Estimating Contact Orientation 24
3.3 Robotic Manipulator 27
3.3.1 Manipulator 27
3.3.2 Software Used 28
3.3.3 Kinematics 29
3.3.4 Transformation between Module and Base-Link Frames 32

4 Surface Estimation **35**
4.1 Data Collection 35
4.2 Control Point Calculation 36
4.3 Surface Estimation 42

5 Results and Discussion **48**
5.1 Synthetic Surfaces 49
5.1.1 Surface 1 49
5.1.2 Surface 2 54
5.1.3 Surface 3 57
5.1.4 Surface 4 61
5.1.5 Surface 5 65
5.2 Everyday Objects 68
5.2.1 Vitamin C Bottle 68
5.2.2 Peanut Massage Ball 72
5.2.3 Hair Trimmer 75
5.2.4 Cream Bottle 79
5.2.5 Transparent Bottle 83
5.2.6 Screw 86
5.2.7 Earphone Case 90
5.2.8 Inhaler 93
5.3 Texture Exploration 98

6 Conclusion **103**

Bibliography **106**

List of Figures

Figure 3.1	Sensing Module Figure [1], 1-MARG, 2-Flexible Structure, 3-Barometer	20
Figure 3.2	Robotic Manipulator with the Sensing Module	28
Figure 3.3	OpenMANIPULATOR-X Kinematic Chain	32
Figure 3.4	Sensor aligned TF	34
Figure 4.1	Data Collection	36
Figure 4.2	Control Point Calculation	42
Figure 4.3	A 3*3 Surface Patch	43
Figure 4.4	Convex Surface	44
Figure 4.5	Convex Surface - Method 1	44
Figure 4.6	Convex Surface - Method 2	45
Figure 4.7	Convex Surface - Method 3	45
Figure 4.8	Convex Surface - Method 4	45
Figure 4.9	Concave Surface	46
Figure 4.10	Concave Surface - Method 1	46
Figure 4.11	Concave Surface - Method 2	46
Figure 4.12	Concave Surface - Method 3	47
Figure 4.13	Concave Surface - Method 4	47
Figure 5.1	Surface 1	49
Figure 5.2	Surface 1 Contact Points	50
Figure 5.3	Surface 1 Barometer Graph	50
Figure 5.4	Surface 1 Accelerometer Graph	50
Figure 5.5	Surface 1 Gyroscope Graph	50
Figure 5.6	Surface 1 Translation Graph	51
Figure 5.7	Surface 1 Orientation Graphs	51
Figure 5.8	Surface 1 Markers	52
Figure 5.9	Surface 1 Estimation	53

Figure 5.10	Surface 2	54
Figure 5.11	Surface 2 Contact Points	54
Figure 5.12	Surface 2 Barometer Graph	54
Figure 5.13	Surface 2 Accelerometer Graph	54
Figure 5.14	Surface 2 Gyroscope Graph	55
Figure 5.15	Surface 2 Translation Graph	55
Figure 5.16	Surface 2 Orientation Graphs	55
Figure 5.17	Surface 2 Markers	56
Figure 5.18	Surface 2 Estimation	56
Figure 5.19	Surface 3	57
Figure 5.20	Surface 3 Contact Points	57
Figure 5.21	Surface 3 Barometer Graph	57
Figure 5.22	Surface 3 Accelerometer Graph	58
Figure 5.23	Surface 3 Gyroscope Graph	58
Figure 5.24	Surface 3 Translation Graph	58
Figure 5.25	Surface 3 Orientation Graphs	59
Figure 5.26	Surface 3 Markers	59
Figure 5.27	Surface 3 Estimation	60
Figure 5.28	Surface 4	61
Figure 5.29	Surface 4 Contact Points	61
Figure 5.30	Surface 4 Barometer Graph	61
Figure 5.31	Surface 4 Accelerometer Graph	62
Figure 5.32	Surface 4 Gyroscope Graph	62
Figure 5.33	Surface 4 Translation Graph	62
Figure 5.34	Surface 4 Orientation Graphs	63
Figure 5.35	Surface 4 Markers	63
Figure 5.36	Surface 4 Estimation	64
Figure 5.37	Surface 5	65
Figure 5.38	Surface 5 Contact Points	65
Figure 5.39	Surface 5 Barometer Graph	65
Figure 5.40	Surface 5 Accelerometer Graph	65
Figure 5.41	Surface 5 Gyroscope Graph	66
Figure 5.42	Surface 5 Translation Graph	66
Figure 5.43	Surface 5 Orientation Graphs	66
Figure 5.44	Surface 5 Markers	67

Figure 5.45 Surface 5 Estimation 67
Figure 5.46 Vitamin C Bottle 68
Figure 5.47 Vitamin C Bottle Contact Points 68
Figure 5.48 Vitamin C Bottle Barometer Graph 68
Figure 5.49 Vitamin C Bottle Accelerometer Graph 68
Figure 5.50 Vitamin C Bottle Gyroscope Graph 69
Figure 5.51 Vitamin C Bottle Translation Graph 69
Figure 5.52 Vitamin C Bottle Orientation Graphs 69
Figure 5.53 Vitamin C Bottle Markers 70
Figure 5.54 Vitamin C Bottle Estimation 71
Figure 5.55 Peanut Massage Ball 72
Figure 5.56 Peanut Massage Ball Contact Points 72
Figure 5.57 Peanut Massage Ball Barometer Graph 72
Figure 5.58 Peanut Massage Ball Accelerometer Graph 72
Figure 5.59 Peanut Massage Ball Gyroscope Graph 73
Figure 5.60 Peanut Massage Ball Translation Graph 73
Figure 5.61 Peanut Massage Ball Orientation Graphs 73
Figure 5.62 Peanut Massage Ball Markers 74
Figure 5.63 Peanut Massage Ball Estimation 74
Figure 5.64 Hair Trimmer 75
Figure 5.65 Hair Trimmer Contact Points 75
Figure 5.66 Hair Trimmer Barometer Graph 75
Figure 5.67 Hair Trimmer Accelerometer Graph 76
Figure 5.68 Hair Trimmer Gyroscope Graph 76
Figure 5.69 Hair Trimmer Translation Graph 76
Figure 5.70 Hair Trimmer Orientation Graphs 77
Figure 5.71 Hair Trimmer Markers 77
Figure 5.72 Hair Trimmer Estimation 78
Figure 5.73 Cream Bottle 79
Figure 5.74 Cream Bottle Contact Points 79
Figure 5.75 Cream Bottle Barometer Graph 79
Figure 5.76 Cream Bottle Accelerometer Graph 79
Figure 5.77 Cream Bottle Gyroscope Graph 80
Figure 5.78 Cream Bottle Translation Graph 80
Figure 5.79 Cream Bottle Orientation Graphs 80

Figure 5.80	Cream Bottle Markers	81
Figure 5.81	Cream Bottle Estimation	82
Figure 5.82	Transparent Bottle	83
Figure 5.83	Transparent Bottle Contact Points	83
Figure 5.84	Transparent Bottle Barometer Graph	83
Figure 5.85	Transparent Bottle Accelerometer Graph	83
Figure 5.86	Transparent Bottle Gyroscope Graph	84
Figure 5.87	Transparent Bottle Translation Graph	84
Figure 5.88	Transparent Bottle Orientation Graphs	84
Figure 5.89	Transparent Bottle Markers	85
Figure 5.90	Transparent Bottle Estimation	85
Figure 5.91	Screw	86
Figure 5.92	Screw Contact Points	86
Figure 5.93	Screw Barometer Graph	86
Figure 5.94	Screw Accelerometer Graph	87
Figure 5.95	Screw Gyroscope Graph	87
Figure 5.96	Screw Translation Graph	87
Figure 5.97	Screw Orientation Graphs	88
Figure 5.98	Screw Markers	88
Figure 5.99	Screw Estimation	89
Figure 5.100	Earphone Case	90
Figure 5.101	Earphone Case Contact Points	90
Figure 5.102	Earphone Case Barometer Graph	90
Figure 5.103	Earphone Case Accelerometer Graph	90
Figure 5.104	Earphone Case Gyroscope Graph	91
Figure 5.105	Earphone Case Translation Graph	91
Figure 5.106	Earphone Case Orientation Graphs	91
Figure 5.107	Earphone Case Markers	92
Figure 5.108	Earphone Case Estimation	92
Figure 5.109	Inhaler	93
Figure 5.110	Inhaler Contact Points	93
Figure 5.111	Inhaler Barometer Graph	94
Figure 5.112	Inhaler Accelerometer Graph	94
Figure 5.113	Inhaler Gyroscope Graph	94
Figure 5.114	Inhaler Translation Graph	95

Figure 5.115	Inhaler Orientation Graphs	95
Figure 5.116	Inhaler Markers	96
Figure 5.117	Inhaler Estimation	97
Figure 5.118	Surface 5	98
Figure 5.119	Surface 5 Estimation	98
Figure 5.120	Textures	99
Figure 5.121	Dynamic Tactile Sensing Path	99
Figure 5.122	Barometer Comparison: Texture A and Texture B	100
Figure 5.123	Accelerometer Comparison: Texture A and Texture B	100
Figure 5.124	Gyroscope Comparison: Texture A and Texture B	100
Figure 5.125	Magnetometer Comparison: Texture A and Texture B	100
Figure 5.126	Barometer Comparison: Texture C and Texture D	101
Figure 5.127	Accelerometer Comparison: Texture C and Texture D	101
Figure 5.128	Gyroscope Comparison: Texture C and Texture D	101
Figure 5.129	Magnetometer Comparison: Texture C and Texture D	101

Chapter 1

Introduction

Nowadays, robots are being used in healthcare, surgery and as an industrial tool for tasks such as manufacturing. The intention is to use these robots in such a way that they interact safely with humans and objects in the environment while also causing no harm to themselves. Achieving this is a much simpler task if a complete model of the environment is available beforehand and the robot can execute commands perfectly with respect to the known environment. But in most real-world scenarios, it is not possible to obtain a complete model of the environment. Therefore, it is required to plan the robots without any prior knowledge other than the general knowledge of the location of the object while also taking into account any uncertainties of the environment and the building model of the environment and its objects. Historically, this has been achieved through vision, by attaching cameras to robots. Although vision is a viable way of achieving this, it has some limitations which can be solved using tactile sensing. In unstructured environments, the field of view can be obstructed by other objects or the robot itself which renders vision useless. There are also cases where vision could be misleading such as the presence of transparent objects in the environment. It is also impossible to detect properties such as texture or hardness using vision. To overcome these situations, robotic systems are being equipped with tactile sensors to carry out tactile exploration.

The sense of touch is an essential component that humans use to efficiently explore their surroundings and handle various objects. Using the feedback obtained from touch, humans are able to identify properties such as pressure, temperature, and texture. Using this information, it is possible to manipulate and identify objects and

also react to potential danger. The property of human skin that allows for such tasks is the softness or the physical conformability of the skin. It enables the grasping of objects with different shapes, detects and stops slippage, allows fast reaction if any danger such as high or low temperature is detected, and also enables the handling of fragile objects which may break if too much force is used.

Drawing inspiration from humans, the sense of touch has been partially replicated as tactile sensing and used in robotic systems. These systems allow robots to make physical contact with objects which is essential for safe and efficient exploration. Tactile sensing allows the estimation of contact parameters such as shape, texture, and hardness which in turn allows for action-related tasks such as slip detection [2] and control-related tasks such as grasping and manipulation [3].

Tactile exploration can be categorized into static tactile exploration and dynamic tactile exploration [4, 5]. Static exploration is done by pressing the tactile sensor on the object and then removing it and pressing it on a new place on the object. Using this method, the global shape of the object can be obtained. Once this initial estimate is obtained, static exploration can be done to perform manipulation tasks and even obtain local features depending on the size of the sensor.

Dynamic exploration deals with sliding the sensor on the object which can be used to obtain local features of the object such as texture. To perform dynamic exploration, the shape of the object must be known. In dynamic exploration, the contact between the sensor and the object changes and the signals obtained from this change can be used to describe local features of the object.

To use tactile exploration to its fullest potential, static and dynamic exploration should be used to complement each other. Using static exploration, the global shape of the object can be estimated. Using this estimation, dynamic exploration could be carried out on the object to uncover its local features.

In order to carry out real-world applications as discussed above, an estimate of the environment must first be obtained. As this environment is most likely unstructured, care must be taken to protect the robot and the environment and therefore static tactile exploration must be used. Once this estimate is obtained, tasks such as grasping and texture recognition can be carried out by using dynamic tactile exploration.

1.1 Objectives

This thesis aims to use tactile sensing to explore unseen surfaces in semi-structured environments and obtain their estimates. We also aim to show that this estimation can be used to perform tasks such as texture exploration.

The environments that robots interact with can be divided into structured, unstructured, and semi-structured environments. Structured environments assume that a complete model of the environment and its objects can be obtained which is impractical in real-life scenarios. Unstructured environments fall into the category where nothing about the environment and its objects are known. Semi-structured environments are similar to unstructured environments except for the fact that the general position of the object to be explored is known although the orientation of the object is unknown. In this thesis, we perform our explorations in semi-structured environments which closely relate to real-life scenarios.

In the context of semi-structured environments, the approach taken by the robot to touch the surface must be carefully planned. It needs to have a predetermined angle of approach and must be precisely controlled so that there is no harm done to the robot, tactile sensor or the environment. Usually, to get the best tactile data, the angle of approach must be normal to the surface. In the most challenging scenarios, it is impractical for robots to approach the surface along its normals as prior information about the surface normals may not be available. To overcome this, we use a bio-inspired tactile sensor that has adapted the cutaneous tactile properties of human skin.

The tactile sensor used in this thesis consists of a compliant structure along with a MARG (Magnetic, Angular Rate, and Gravity) sensor and a pressure sensor. The compliant structure allows the sensing module to deform when it comes into physical contact. The MARG is used to collect orientation data relative to the deformation of the sensing module. The pressure sensor collects pressure data when the sensing module comes into physical contact. A constant angle of approach for the robotic end-effector holding the sensing module is used for each contact with the surface. This allows the sensing module to deform along the surface normals at each contact point to obtain good orientation data. Once contact is made, the pressure sensor can detect when to stop the robot to protect itself and the environment from harm.

To this end, the following objectives facilitate the achievement of this aim.

1. Use static tactile exploration to obtain a global estimate of the surface. We use probes to collect points on the surface which are used to obtain an estimate of the surface.

2. Use a constant angle of approach to collect the points. This is important as we are collecting data using the deformation of the sensor. As the surface normal at a given contact point is unknown, we keep a constant angle of approach before deformation when collecting data on different places on the surface. Using this technique, the surface normal could be obtained. Using a constant angle of entry also helps to reduce any damage that could occur to the robot or sensor when entering an unstructured environment as the robot could be stopped when the sensor feels pressure.

3. Calculate control points between the contact points which reflect the curves between the contact points. This allows the use of fewer probes to obtain a similar estimate of the surface. Using fewer probes means that less effort is used which can save time and resources.

4. Using the obtained estimate, use dynamic tactile exploration to show that the texture of the surface can be explored.

1.2 Methodology

This thesis explores the task of estimating a surface in 3D using a robot equipped with a bio-inspired multi-modal sensing module. We use the gripper of a 4-DOF (degree of freedom) robotic manipulator to hold the sensing module to collect points on a surface by using static tactile exploration while keeping the orientation of the gripper constant. Using this experimental setup, we are able to explore surfaces in unstructured environments. The deformation of the sensing module is used to collect pose data. We use the position of the end effector of the robot and the orientation of the MARG sensor to collect the data. Using the surface normals and the knowledge of geometric operations, we calculate control points that reflect the curve in between the contact points which allows for a better estimation using fewer probes. Using a combination of the collected contact points and the calculated control points, we use Bezier surface patches to obtain a similar estimate of the surface. Using the estimate

of the surface, we use the robot to slide the sensing module over the surface to obtain information about the texture of the surface.

1.3 Thesis Organization

The thesis is organized into the following chapters:

Chapter 2: Literature Review

A comprehensive review of the literature is presented in this chapter. It is broken down into 3 main sections; pose estimation, shape perception, and texture recognition.

Chapter 3: Experimental Setup

A thorough explanation of the sensing module and the robotic manipulator is presented in this chapter. It also contains the method used to align the sensing module with the robotic manipulator.

Chapter 4: Surface Estimation

This chapter dives into how data is collected using the sensing module along with the manipulator and how that data is leveraged to estimate the surfaces. We also look into how control points are calculated by using various geometric equations.

Chapter 5: Results and Discussion

The results obtained through surface estimation are shown and discussed here. A small demonstration of texture recognition is also shown.

Chapter 6: Conclusion

The work done in this thesis is summarized here and it also gives insight into future work.

Chapter 2

Background

2.1 Introduction 6
2.2 Shape Perception 7
2.2.1 Local Methods 7
2.2.2 Global Methods 10
2.3 Pose Estimation 12
2.4 Texture Recognition 15
2.5 Discussion 17

2.1 Introduction

Tactile perception is an emerging field that imitates the sense of touch in humans. It has been inspired by the success of computer vision and hence visuo-tactile approaches are becoming increasingly popular. Tactile perception is essential in tasks such as grasping [6, 7] and slip detection. It is also used to measure features such as hardness [8] and texture [9, 10]. For such tasks, an initial estimate of the object must first be obtained. As such, tactile perception has been used in Shape Perception, Pose Estimation, and Texture Recognition. Their related work is discussed in the following sections.

2.2 Shape Perception

Shape perception aims to recognize the shape of an object. In tactile sensing, this is of utmost importance as it directly correlates to the accuracy of manipulation and exploratory strategies. Historically, shape perception has been achieved using computer vision techniques. However, computer vision has a few problems which it cannot solve by itself such as the presence of occluded objects, either by the environment or by the robot itself and the presence of objects which are not rich in visual features [11]. Shape perception through tactile sensing is not affected by these issues.

Similar approaches which closely resemble the surface estimation method used in this thesis have been proposed which were successful in surface estimation [12, 13, 14]. However, it was assumed that the surface to be estimated was known and a method to estimate unknown surfaces was not considered.

Tactile perception can be divided into local tactile perception and global tactile perception. Local perception of an object aims to identify properties in a small area of the object and these features are usually extracted using a single touch. Global perception aims to extract features from the object to fully recognize the object. It is usually done by collecting multiple points on the object by taking into account the data from the tactile sensor as well as the position of the robotic end-effector.

2.2.1 Local Methods

Many local shape perception tactile systems have been proposed which can be divided by the shape descriptors used such as raw tactile readings, statistical readings, principal component analysis (PCA) based features, vision-based descriptors, and self-organizing features [15].

Raw Tactile Readings

Earlier applications saw raw tactile readings being used as features for shape perception. A robot with touch-sensitive fingertips was used to identify objects in [16]. The training data which was a collection of tactile images was clustered using k-means and a histogram was created for each touch from the touch-sensitive fingertips using the clusters and the frequency of the predicted clusters. The histograms were then used as bag-of-features which minimized the number of touches needed to identify the

object. A similar bag-of-features model which used appearance-based features was proposed in [17]. Tactile pressure sensing was used by [18] to recognize the shape of an object that is in contact with a robot hand. A multi-element pressure array sensing system called "Tekscan" was used on a "Shadow" robot hand for this purpose. The output of this system was a pressure map. A 512-feature vector was extracted from this pressure map which was used in a neural network to classify the shape. These methods were susceptible to pose variations and therefore might give different results for the same object. An active contour following method to shape recognition was proposed by [19]. An iCub fingertip was used in an action-perception cycle in which the fingertip taps the object and the data obtained was processed and classified to figure out the next position to tap. Local shapes were classified using the path of the taps.

Statistical Features

The advantage of statistical features is that they are very easy to obtain but on the flip side their effectiveness is not guaranteed. Statistical features consisting of maximum, minimum, and mean pressure were used in [20] to form a 155-dimension vector which showed a low accuracy of 60%. In [21], the statistical features of tactile arrays were used to measure the object class and the internal state of bottles and cans.

Computer Vision Descriptors

Inspired by computer vision, several tactile shape perception adaptations treat tactile arrays as images so that image descriptors could be applied to them. Several approaches considered image moments as feature descriptors [17, 22, 23, 24, 25]. In these approaches, a tactile reading was considered as $f(x, y)$ where x and y stand for the horizontal position and vertical position of the tactile image respectively. Its image moment of order $p + q$ was defined by m_{pq} where

$$m_{pq} = \sum_{x} \sum_{y} x^p y^q f(x, y) \tag{2.1}$$

Many other vision descriptors were explored in tactile sensing such as Scale-Invariant Feature Transform (SIFT) [17, 26, 27, 28], SURF [29] and the 3D descriptor SHOT [30]. Drawing inspiration from SIFT [31], a novel shape descriptor called Tactile-SIFT was proposed in [27]. The descriptor decomposed tactile images into subpatches

which were represented using gradient vectors. A bag-of-features model was used to classify the shapes. It was found that 3 subpatches with 8D descriptors showed the best performance. These descriptors were resistant to object movement and rotation but the scale-invariance factor is redundant in tactile systems. A robust and noise resistant algorithm to edged and edge-less shape recognition used a gradient-based analysis of tactile images [32].

PCA Features

Principal Component Analysis (PCA) has been used in tactile systems by using the Principal Components (PCs) as features. These features are easy to implement and reduce the redundancy of tactile data but it lacks physical meaning. In [33], PCs with the largest Eigenvalues were computed onto a lower-dimensional feature space using readings from a 16x16 tactile array. A different approach to compute the data taken from a pressure array was proposed by [34] in which the tactile image was processed using PCA to produce a computationally efficient and rotation invariant shape classification. A kernel-based PCA was used in [28] which used a fusion of Fourier descriptors along with geometric descriptors for shape perception.

Self-Organizing Features

Raw tactile sensor data has been used in deep neural networks where they learn self-organizing features. This is in contrast to the other features above which are pre-defined and handcrafted. Temporal tactile data was used to develop a generative model which integrated a recursive kernel with a Gaussian process to classify objects in [35, 36]. Features were extracted from raw tactile data by using a sparse coding model of unsupervised hierarchical feature learning in [27, 37]. Another sparse coding model was proposed by [38] in which a joint kernel was used to classify tactile data from multiple fingers. A 20% improvement in accuracy for classifying 20 objects was observed in [39] compared to shallow networks by applying a denoising autoencoder with a dropout function. An efficient neural network for learning tactile features was proposed by [40] which was named the Randomizing Tiling Convolutional Network (RTCN).

Self-organizing features show promising results but their computational costs are very high while it is also very hard to tune the parameters to get good results consistently.

2.2.2 Global Methods

Methods to identify the global shape of objects can be categorized into point based methods, tactile pattern based methods, and deep learning methods.

Point Based Methods

Point based methods adapt techniques from computer vision to fit the obtained contact tactile data into a geometric model to identify objects. In the past, point based methods were used often due to the available tactile sensors being low resolution and single-point contact force sensors [41, 42]. In [41], the tactile readings were fit to super quadratic surfaces while in [42] a polyhedral model was defined to estimate object shapes. Contact points were used to describe curvatures using polynomial fitting of curves in [43, 44]. These approaches highlight the importance of surface normals when estimating curvature.

Taking advantage of the high frame rates of new tactile sensors, a tactile point cloud representation was created in [45]. The point cloud was a probabilistic model created using Kalman filters. The Iterative Closest Point algorithm was used to calculate the distance between 2 point clouds which was used to classify the object. "Iterative Closest Labeled Point" (iCLAP) [46] fused tactile data and kinesthetic cues to recognize the global shape of objects. A bag-of-words model was used and each feature was assigned a label using k-means clustering. Using the label as the 4th dimension, 4D point clouds were created to use for training. To predict the shape, an iterative method was used to calculate the closest distance from the obtained partial 4D point cloud to the learned representations.

Point based methods suffer from the fact that a large number of contact points must be obtained to obtain good results. They are very time consuming especially when working on large objects.

Tactile Pattern Based Methods

The global shape of objects could also be found by using the pressure distribution in tactile arrays. A widely used and popular method in this category is to use a codebook of tactile features and use it in a Bag-of-Words model [16, 17, 37]. In [16], the tactile readings of contact features were clustered to form a dictionary. The cluster centroids were taken as the codewords and the dictionary was used with these

codewords to assign each tactile feature to a fixed-length vector. These vectors were used to represent the shape of the object.

These methods achieve respectable results but they do not represent features in 3D space as they only take local features into account.

Deep Learning Methods

A Deep CNN (DCNN) was used in an object recognition task that used the contact shape of the object to classify it [47]. High-resolution pressure tactile sensors were used to obtain tactile data which was fed into a pre-trained DCNN to extract features. These features were then used in the DCNN with added custom layers to classify the shape of the object. A very similar approach was taken by [48] in which features were learned using transfer learned CNN and classification was done using a custom network created from scratch called "TactNet". This study was extended in [49] by using a gripper in which the palpation process is autonomous which allowed the classification of objects with different elasticities and internal inclusions.

The global based methods are able to obtain the global shape of the object but the methods used are very time consuming and use more resources. The method proposed in this thesis allows the global estimation of surfaces using fewer contact points. The estimation is also accurate as control points are calculated to help estimate the surface.

Combination of Vision and Touch

The recent trend in shape perception is to use vision-based systems and complement them using tactile sensing [50, 51]. This aims to fix the inherent problems of vision using a low number of tactile touches to augment the information obtained using vision. A probabilistic approach to object recognition using both visual and tactile data was proposed by [51]. A model was created using only vision which was modeled with uncertainty. The highest uncertain regions were then touched with a tactile sensor to obtain tactile information which was used to refine the model. It was found that around 10 touches were enough to create models for object recognition. An interesting study on cross-modal visual-tactile object recognition was carried out by [52]. A model was trained using only visual data and was used to perform object recognition using tactile data. It was found that a visual-tactile point cloud repre-

sentation was a good unified representation of visual and tactile data. A method to learn full 3D shapes of objects using vision, touch, and pre-trained object shape priors was proposed by [50]. Using a single RGB or RGB-D image on a neural network trained on the shape priors, a 3D object shape was predicted. Tactile sensing was then used to refine the object shape. An exploration policy that actively selected points that maximize uncertainty was used to reduce the number of touches taken. A contour following tactile sensing system showed better performance when the contour following was assisted by a vision-based system instead of blindly following randomly determined contours [53].

2.3 Pose Estimation

The pose of an object refers to the orientation or position of the object. The method used to determine this orientation and position is known as pose estimation. This estimate is done with reference to a global coordinate system or in robotics, with reference to the robot end-effector. Pose estimation is an important task in robotics especially in object manipulation tasks such as grasping. Earlier applications tended to use single-point contact-based systems as the hardware was not up to standard. These systems were not very efficient as the number of contacts needed is very high. As the performance of tactile sensors increased, it became possible to estimate the pose of an object using the tactile array data obtained after grasping it with a robot hand.

Bayesian Filtering

The information gained from single-point contact data was used to perform 6DOF pose estimation in [54]. The problem was divided into 2 sections called over-constrained and under-constrained. In the over-constrained case, the information obtained from the tactile sensor is enough to perform localization and it was done using gradient descent. In the under-constrained case, a modified Monte Carlo filtering technique called "Scaling Series" was proposed. This technique uses an iterative method to approximate solution regions into samples which reduces computational costs and efficiently estimates the 6DOF pose of the object. This approach could be used to estimate the pose of objects which can be represented as polygonal meshes. The Scaling Series approach was used to perform object localization on under-constrained

and over-constrained scenarios in [55].

Particle filtering was used by [56] to obtain the pose of an object touched by the robot "Robonaut 2". After touching the object, the pose is calculated before grasping the object. A measurement model was created that shows the likelihood of the contact points being on the surface of the object. It also estimates the tactile positions which are not touching the object. The Robonaut 2 was also used in [57] to localize features of flexible objects.

Contact manipulation is where a robot manipulates an object by touching it continuously. The information obtained from contact sensors in such a task is very discriminative between contact and no contact states and therefore performs poorly when used in particle filters. A modified particle filter called "Manifold Particle Filter" (MPF) was proposed by [58] to overcome this problem. The MPF allows multiple manifolds in which different sampling techniques can be used to avoid particle starvation and give better pose estimates.

A Monte Carlo approach was used to globally obtain the pose of an object in [59]. This system can be used in 2 ways; to predict the pose of an object without any prior estimation or to improve a prior estimation. The object polygon mesh must be known in advance to use this approach. A novel algorithm called "Memory Unscented Particle Filter" (MUPF) was proposed by [60] to 6DOF pose estimation using tactile sensing. MUPF uses past measurements to preserve particle quality. As this leads to increased complexity, a sliding window with the most recent measurement is used which could be changed according to the computational power available.

Covariance Analysis

A tactile array sensor attached to the tip of a robot finger was used by [34] to calculate the shape and pose of a local contact area. The information from the tactile array was extracted using Covariance Analysis which led to a computationally fast algorithm. The shape and pose was calculated using a Naive Bayes classifier A global technique to perform 6D pose estimation of grasped objects was carried out by [61]. The covariance matrix of the tactile data was extracted and used to perform Principal Component Analysis to reduce the dimensionality of the data.

Tactile Feature Mapping

Tactile sensing was used to localize an object in a robot gripper in [62]. A key contribution was a tactile map created prior to object manipulation. During manipulation of the object, the tactile information obtained is sent to the tactile map which is used to obtain the pose of the object relative to the robot gripper. A Gelsight sensor was used to perform tactile sensing in this system.

Many pose estimation systems have been brought forth that deal with challenges such as clutter, occlusion, scaling, illumination, and symmetry extremely well. Although these systems deal with these challenges well, they do not perform well when multiple challenges have to be solved at once. Efficient pose estimation using tactile sensing requires a prior estimation of the surface. The pose estimation methods discussed in this section could be integrated with the work done in this thesis to get better estimates.

Combination of Vision and Touch

As the 3D point data generated from tactile and vision systems are very similar, they have been combined to achieve better results in pose estimation. In most of these hybrid systems, an initial estimate of the pose is first obtained using vision and is then refined by using tactile sensing.

One of the earliest systems to use visual and tactile data to predict a 6-DOF pose was a real-time system that used a multi-fingered hand [62]. An initial estimate of the pose of the object was first obtained using a vision system. The pose of the occluded objects was then refined using the tactile data obtained from the multi-finger hand touching the object. An initial pose estimate obtained using Microsoft Kinect camera was refined using tactile data obtained from the fingertips of a Shadow arm in [63]. Using forward kinematics and the tactile data obtained from the fingertips, an iterative 3D transformation called "Levenberg-Marquardt" was used to refine the object's pose so that the fingertips will match the surface of the object. This work was extended by using a novel pose correction algorithm in [64]. The robot DARCI was used to create a haptic map of its surroundings by fusing tactile and visual data [65]. A global optimization method to pose estimation was proposed by [59] in which tactile and force sensing was used to identify the pose of an object by itself or to improve an initial pose obtained by vision. Two recent studies approached in-hand

pose estimation of grasped objects by combining tactile and vision data. An initial estimate obtained from vision was improved using the tactile data obtained from the grasped object using a particle filtering method called "Bootstrap Particle Filtering" in [66] and a deep learning network called "Deep Gated Multi-Modal Learning" in [67].

2.4 Texture Recognition

Texture is an important property of a surface that describes the patterns and the feel of the surface. Dynamic tactile sensing can be used to explore surfaces to recognize their texture [68]. However, to apply this to real-life scenarios, the textures of unknown surfaces must be found which must be considered by the researchers.

Statistical Methods

A polymer-based microelectromechanical tactile array system called MEMS was used in [69] to classify textures. The single-point data obtained from the MEMS tactile array was assumed to be Gaussian and was estimated using the Maximum Likelihood estimation. The Probability Density Function of each array was calculated to classify the texture. Texture recognition was performed in [70] by using an artificial finger with a microphone to detect frictional sound by tapping on a surface. The detected sounds were used to create a library which was then mapped to the frequency domain using Fast Fourier Transforms (FFT) for texture analysis. The features were transformed using PCA and classified using the K-Nearest-Neighbours algorithm. A biomimetic finger was used in [71] to extract texture, compressibility, and thermal data from an object. PCA was used on the data to extract features. 7 different classifiers were used to test material recognition and it was found that a two-stage Support Vector Machine performed the best. A texture sensor alongside a microphone was used for texture recognition by [72]. FFT was used on the raw data to create a spectrogram of 2049 frequencies which was represented as a feature vector and was classified using Self-Organising Maps. A Polyvinylidene Flouride (PVDF) embedded silicone finger with randomly distributed strain gauges was used in [73, 74] for texture recognition. The regions where the finger touches the surface were segmented and converted to the frequency domain using FFT. The Fourier coefficients were used as features to classify textures using a Naive Bayes classifier. By mimicking

how humans touch objects to identify texture, a robotic fingertip was used to slide over objects to identify their texture in [75, 76]. In [75], the fingertip was knitted with electro-conductive tension sensitive yarns and it was found that the Discrete Wavelet Transformation gave the best classification accuracy.

In [76], a dynamic friction model was introduced which performed the best with the Naive Bayes classifier. Using humanoid robot hands, the vibrations that occurred from sliding along the hands was captured using accelerometers and used as features to learn textures in [77, 78]. The heightmap obtained from a GelSight sensor was used in [79] for texture classification. The heightmap was considered as an image and Local Binary Pattern (LBP) histograms were extracted as features. A multiscale Gaussian pyramid was used to reduce the dimensions of the image before applying LBP. Classification was done by comparing the similarity between images using Hellinger distance. By using a highly-discriminative tactile probe which was sensitive to surface texture, [80] showed that unsupervised texture recognition was possible. Using the triple-axis accelerometer tactile probe proposed in [81], textures were clustered using the "Expectation-Maximization" algorithm combined with the "Minimum Description Length" principle.

Deep Learning Methods

Neural Networks were used for texture classification by using the time series data obtained from a pressure sensor and a 6-axis accelerometer in [82, 83]. A Convolutional Neural Network was used in [82] and a Recurrent Neural Network was used in [83] to analyze the data which showed similar results. A novel neuromorphic optical tactile sensor called "NeuroTac" was proposed by [84] for texture classification. 4 spike train encoding methods were used and it was found that the temporal encoding method performed the best. Texture classification was done using a K-Nearest-Neighbour classifier.

Similar to pose estimation, texture recognition requires a prior estimate of the object. Using the work done in this thesis, once the surface is estimated, the contact points collected could be used to program the robot to follow the contact points and then use the data of the pressure sensor to obtain an understanding of the texture of the surface.

Combination of Vision and Touch

Even though tactile sensing is very successful in texture recognition, there is still room to improve it by combining it with vision. For example, with tactile sensing, you cannot realize the color of the surface.

A joint CNN was used in [85] to integrate vision and tactile images to recognize the physical properties of fabrics. A camera was used to obtain color images and depth images and a GelSight sensor was used to obtain the tactile images. The joint CNN was created using these 3 modalities and classification was done by calculating the similarity using a distance measure.

A method to fuse visual and tactile data for texture recognition was proposed in [86, 87]. Two sets of features were learned from camera image and from a GelSight sensor using deep neural networks. These features were fused together using a novel method called "Deep Maximum Covariance Analysis" (DMCA) which greatly reduced the dimensionality of the features.

By using a Kinect Camera and a GelSight sensor, a method to train a robot to explore unknown clothes autonomously was proposed by [88]. The Kinect sensor was used to guide the robot to actively select points on the clothes to touch. Two neural networks were created; one to select the points on the clothes and the other to estimate the properties of the clothes that were touched using the GelSight sensor.

2.5 Discussion

Various methods used for shape perception, pose estimation, and texture recognition have been studied in the above chapters.

Most Shape Perception methods use local methods and cannot estimate the global shape of objects. The global methods used require a large number of contact points or do not represent the tactile features in 3D. These methods also assume that prior knowledge of the object is known so that the points can be collected safely. In this thesis, we propose a global estimation method that does not require the surface or its environment to be known. Using this estimation, it is possible to perform Pose Estimation and Texture Recognition on these surfaces. The literature has also shown that it is possible to combine vision and tactile sensing to create combined models.

Extending our work to incorporate vision is for future work.

Chapter 3

Experimental Setup

3.1 Tactile Perception 19
3.2 Tactile Sensing Module 20
3.2.1 Structure 20
3.2.2 MARG Sensor 21
3.2.3 Orientation Representation 22
3.2.4 Estimating Contact Orientation 24
3.3 Robotic Manipulator 27
3.3.1 Manipulator 27
3.3.2 Software Used 28
3.3.3 Kinematics 29
3.3.4 Transformation between Module and Base-Link Frames 32

3.1 Tactile Perception

Tactile Perception refers to the use of tactile sensing to gain an understanding of the environment or the objects being touched. It is needed to estimate features such as hardness and texture and can be used to the geometric features of touched objects which is essential is tasks such as grasping and manipulation. In real-life scenarios, robots are often expected to operate in unstructured environments. Robots need to operate safely without causing harm to themselves or the objects in the environment

while taking into account environmental uncertainties and factors such as the fragility of the objects.

In this context, we propose a method to obtain the estimate of unknown surfaces in an unstructured environment. We use a multi-modal tactile sensing module alongside a robotic manipulator for this purpose. The tactile sensing module is a compliant structure that consists of a MARG (Magnetic, Angular Rate, Gravity) sensor and a pressure sensor. Using rigid sensors the angle of approach must be normal to the surface of the object being explored which is impossible to do if the object is unknown. Using the tactile sensing module used in this thesis, we are able to approach the surface with a pre-defined angle for the end-effector which is essential in exploring unknown surfaces while also accounting for unexpected disturbances. We use the gripper of the robotic manipulator to hold the sensing module and we collect orientation data obtained from the MARG sensor and the position data from the end effector of the robot. Using this data we calculate control points and estimate the surface using Bezier Curves which is explained in Chapter 5. The tactile sensing module and the robotic manipulator used in this thesis are explored in this chapter.

3.2 Tactile Sensing Module

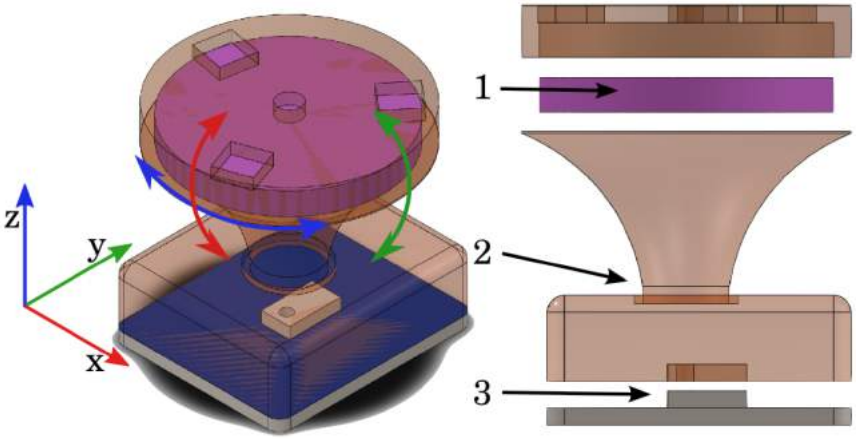


Figure 3.1: Sensing Module Figure [1], 1-MARG, 2-Flexible Structure, 3-Barometer

3.2.1 Structure

The bio-inspired multi-modal sensing module proposed in [89, 90] is used in this work. It is comprised of a STMicroelectronics© LSM9DS0 MARG (Magnetic, An-

gular Rate, Gravity) sensor and a Freescale Semiconductor© MPL115A2 Barometer pressure sensor. The MARG consists of a triple-axis accelerometer, a triple-axis gyroscope, and a triple-axis magnetometer. These sensors are embedded in a Smooth-On© VytaFlex© Shore 20 Hardness flexible polyurethane and its structure is shown in Figure 3.1.

This sensing structure was inspired by the tactile perception of human skin. The human skin contains mechanoreceptors that help humans identify the features (texture, elasticity, topology) of the touched object. There are 4 types of mechanoreceptors; Meissner's corpuscles, Merkel disks, Pacinian, and Ruffini corpuscles. Merkel disks and Ruffini corpuscles are slow adapting receptors which means they are good at detecting constant stimuli such as pressure. Meissner's corpuscles and Pacinian corpuscles are fast adapting receptors and can detect short pulses such as the initial touch. These mechanoreceptors work together in synergy to correctly percept the different stimuli that come into contact with the human in various tasks such as touching, grasping, or gliding.

The components of the tactile sensing module adapt the functions of these mechanoreceptors. The MARG is a shallow sensor that measures vibrations by emulating the function of Merkel disks and Meissner's corpuscles. The barometer is a deep pressure sensor that measures deep pressure and pressure changes by emulating the function of Pacinian corpuscles. The hardness allows for a compliant pyramidal structure that connects the shallow sensors and the deep sensors. This structure mimics the intersecting area between the receptive fields of a Merkel disk, a Ruffini corpuscle, a Meissner's corpuscle, and a Pacinian corpuscle.

3.2.2 MARG Sensor

The 9-DOF (Degree-of-Freedom) MARG Sensor contains a 3-DOF Accelerometer, 3-DOF Gyroscope, and a 3-DOF Magnetometer.

Accelerometer

The module contains a triple-axis accelerometer which is an inertial-frame sensor that measures the acceleration of all 3 orthogonal axes. If the device is laying on a surface, the acceleration in the z-direction is the negative value of gravity. If the device is falling downwards towards the ground, the acceleration in the z-direction is

$0m/s^2$. According to Newton's Second law of motion, $F = ma$ where F is the force in Newtons, m is the mass of the body and a is the acceleration. Accelerometers use a force-detection method to measure the Force and divide it by the mass of the sensor to obtain the acceleration $a = F/m$.

Gyroscope

The module contains a triple-axis gyroscope which is a sensor that measures the angular velocity of all 3 orthogonal axes. The main advantage of a gyroscope is that it can measure rotation which the accelerometer cannot. A gyroscope contains a vibrating proof of mass and when the sensor is rotated around an axis, the generated force can be measured by using the Coriolis force. The size of the generated Coriolis force is proportional to the rate of rotation.

Magnetometer

The module contains a triple-axis magnetometer is a sensor that measures the flux of earth's magnetic field. Magnetometers are sensitive to outside magnetic influence which can be divided into hard-iron distortions and soft-iron distortions. Hard-iron sources are the magnetic sources that are in the fixed frame of the magnetometer and these only cause a fixed offset in the measurement. Soft-iron sources are outside magnetic sources that cause distortions in the magnetic field which causes errors in the measurement.

3.2.3 Orientation Representation

The data obtained from the MARG sensor is in the form of a quaternion which represents a rotation in 4 dimensions. Quaternions are more accurate and are easier to manipulate than Euler angle rotation matrices. They are also more efficient compared to computationally expensive matrix multiplications.

A quaternion is defined by a scalar q_w and 3 imaginary vector components $q_x i$, $q_y j$ and $q_z k$.

$$q = q_w + q_x i + q_y j + q_z k \tag{3.1}$$

which has the following properties,

$$ii = jj = kk = -1 \tag{3.2}$$

$$ij = -ji = k \tag{3.3}$$

$$jk = -kj = i \tag{3.4}$$

$$ki = -ik = j \tag{3.5}$$

To describe a quaternion with relative frames, a notation using leading superscripts and subscripts is used. The frame being described is denoted using the leading subscript and the frame to which it is referenced is denoted using the leading superscript.

$${}^{A}_{B}q = q_w + q_x i + q_y j + q_z k \tag{3.6}$$

The above equation describes the orientation of frame B with reference to frame A.

The conjugate quaternion swaps the relative frames and is denoted below.

$${}^{A}_{B}q^* = {}^{B}_{A}q = q_w + -q_x i + -q_y j + -q_z k \tag{3.7}$$

Quaternion multiplication of two quaternions q^1 and q^2 can be expressed as follows,

$$q^1 \otimes q^2 = \begin{bmatrix} q^1_w & -q^1_x & -q^1_y & -q^1_z \\ q^1_x & q^1_2 & -q^1_z & q^1_y \\ q^1_y & q^1_z & q^1_w & -q^1_x \\ q^1_z & -q^1_y & q^1_x & q^1_z \end{bmatrix} \begin{bmatrix} q^2_w \\ q^2_x \\ q^2_y \\ q^2_z \end{bmatrix} \tag{3.8}$$

3.2.4 Estimating Contact Orientation

We use the Madgwick filter [91] on the data obtained by the MARG to estimate an orientation in quaternion form. The initial orientation estimation of the filter is based only on gravity and the angular rate data obtained by the MARG's accelerometer and gyroscope. The Madgwick filter fuses the data from the accelerometer and gyroscope by using gradient descent to optimize a quaternion that orients the data from the accelerometer to a known reference of gravity. This quaternion is then weighted and integrated with the data obtained from the gyroscope and the initial estimate which compensates for the gyroscope drift.

Angular Orientation Estimate

The triple-axis gyroscope measures the angular velocity about the x, y and z axes where the angular rate is defined as ω_x, ω_y and ω_z respectively.

These parameters can be arranged into a vector ${}^{S}\omega$.

$$ {}^{s}\omega = [0, \omega_x, \omega_y, \omega_z] \tag{3.9} $$

The quaternion that describes the rate of change of orientation of the earth frame relative to the sensor frame ${}^{S}_{E}\dot{q}$ can be described as follows.

$$ {}^{S}_{E}\dot{q} = (1/2){}^{S}_{E}\hat{q} \otimes {}^{s}\omega \tag{3.10} $$

The quaternion that describes the rate of change of orientation of the earth frame relative to the sensor frame at time t ${}^{S}_{E}q_{\omega,t}$ can be described by integrating ${}^{S}_{E}\dot{q}_{\omega,t}$.

$$ {}^{S}_{E}\dot{q}_{\omega,t} = (1/2){}^{S}_{E}\hat{q}_{est,t-1} \otimes {}^{s}\omega_t \tag{3.11} $$

$$ {}^{S}_{E}q_{\omega,t} = {}^{S}_{E}\hat{q}_{est,t-1} + {}^{S}_{E}\dot{q}_{\omega,t}\Delta t \tag{3.12} $$

where Δt is the sampling period and ${}^{S}_{E}\hat{q}_{est,t-1}$ is the previous estimate of orientation.

The subscript ω indicates that the quaternion is estimated using angular rate; i.e using the gyroscope.

Vector Orientation Estimate

To obtain an initial orientation, it is assumed that the triple-axis accelerometer only measures gravity and the triple-axis magnetometer only measures the earth's magnetic field.

Consider ${}^{S}_{E}\hat{q}$ to be the orientation of the sensor, ${}^{E}\hat{d}$ to be the reference direction of the field in the earth frame and ${}^{S}\hat{s}$ to be the measured direction of the field in the sensor frame.

$$ {}^{S}_{E}\hat{q} = [q_1, q_2, q_3, q_4] \tag{3.13} $$

$$ {}^{E}\hat{d} = [0, d_x, d_y, d_z] \tag{3.14} $$

$$ {}^{S}\hat{s} = [0, s_x, s_y, s_z] \tag{3.15} $$

An optimization problem is proposed where ${}^{S}_{E}\hat{q}$ can be found as the solution.

$$ \min f({}^{S}_{E}\hat{q}, {}^{E}\hat{d}, {}^{S}\hat{s}) \tag{3.16} $$

The objective function f is defined as follows,

$$ f({}^{S}_{E}\hat{q}, {}^{E}\hat{d}, {}^{S}\hat{s}) = {}^{S}_{E}\hat{q}^{*} \otimes {}^{E}\hat{d} \otimes {}^{S}_{E}\hat{q} - {}^{S}\hat{s} \tag{3.17} $$

Using gradient descent, this function can be optimized which gives ${}^{S}_{E}q_{\Delta,t}$ at time t and the previous estimate of orientation ${}^{S}_{E}\hat{q}_{est,t-1}$.

$$ {}^{S}_{E}q_{\Delta,t} = {}^{S}_{E}\hat{q}_{est,t-1} - \mu_t(\frac{\Delta f}{\Delta \| f \|}) \tag{3.18} $$

$$\mu_t = \alpha \| {}^S_E\dot{q}_{\omega,t} \| \Delta t, \alpha > 1 \tag{3.19}$$

Δt is the time period, ${}^S_E\dot{q}_{\omega,t}$ is the angular rate quaternion measured by the gyroscope, and α is a variable which accounts for noise in the accelerometer and magnetometer measurements.

Fusion Orientation Estimate

The madgwick filter outputs a fusion of the vector orientation estimate ${}^S_E q_{\Delta,t}$ and the angular orientation estimate ${}^S_E q_{\omega,t}$ as the fusion quaternion estimate ${}^S_E\hat{q}_{est,t}$.

$${}^S_E\hat{q}_{est,t} = \gamma_t {}^S_E q_{\Delta,t} + (1-\gamma_t) {}^S_E q_{\omega,t}, 0 <= \gamma_t <= 1 \tag{3.20}$$

where γ_t and $1-\gamma_t$ are the weights applied to the orientation calculations.

The contact orientation estimated from using the Madgwick filter on the MARG data is ${}^S_E\hat{q}_{est,t}$

Calibration

After the sensor is initialized, we keep the sensor still for 10 seconds to remove the level 0 bias. However, the gyroscope will still drift over time because of motion and temperature. This needs to be accounted for in the orientation estimate.

If the angular measurement is ${}^S\omega$, the bias is ${}^S\omega_b$ and the compensated measurement is ${}^S\omega_c$.

$${}^S\omega_{c,t} = {}^S\omega_t - {}^S\omega_{b,t} \tag{3.21}$$

where

$${}^S\omega_{b,t} = \zeta \sum_t {}^S\omega_{\epsilon,t} \Delta t \tag{3.22}$$

where ζ is the gain and,

$$ {}^{S}\omega_{\epsilon,t} = 2 {}^{S}_{E}\hat{q}^{*}_{est,t-1} \otimes {}^{S}_{E}\dot{\hat{q}}_{\epsilon,t} \tag{3.23} $$

The compensated measurement ${}^{S}\omega_{c}$ can be used instead of the angular measurement ${}^{S}\omega$.

3.3 Robotic Manipulator

3.3.1 Manipulator

The robotic manipulator used in this work is the OpenMANIPULATOR-X RM-X52-TNM. It contains 5 Dynamixel actuators, with 4 of them being joints and the last one being the gripper as seen in Figure 3.2. Joint 1 allows the robot to move around the base while the other joints can only move in the plane defined by the position of joint 1. This allows for 4 Degrees of Freedom (DOF).

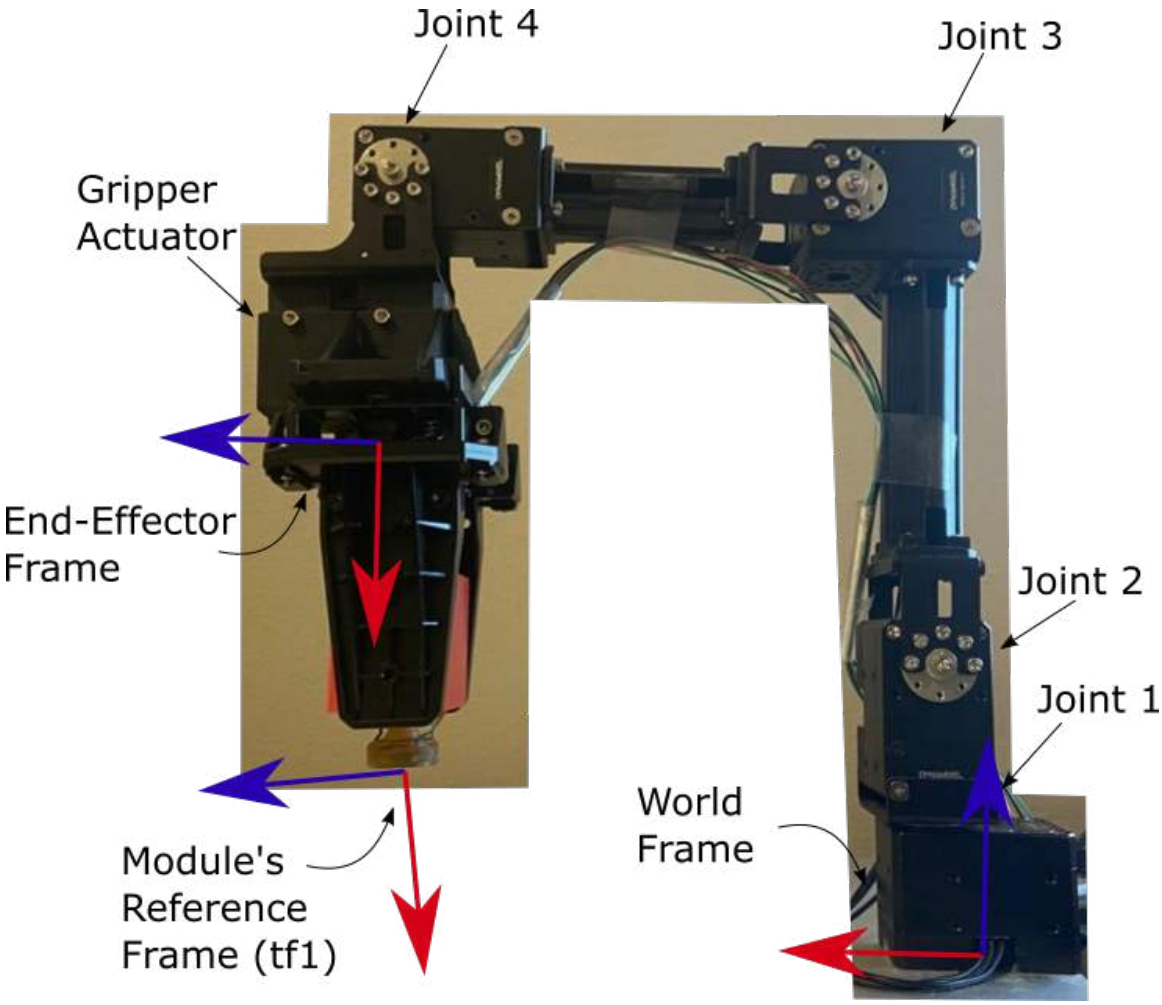


Figure 3.2: Robotic Manipulator with the Sensing Module

3.3.2 Software Used

ROS

To control the robot and integrate it with the sensing module to perform experiments, we used Robot Operating System (ROS) on a Linux computer. ROS has many useful libraries that were made use of in this thesis. ROS is an integrated platform in which robotic systems can be controlled. The main components of ROS are nodes, topics, services, and messages. The major functionality of the robot is divided and stored as nodes. Each node is typically a process that performs a task. These nodes communicate with each other by sending and receiving messages. A message is a standard data structure in where primitive data types are supported. Topics are named buses to which nodes can subscribe or publish to. There can be multiple

publishers and subscribers concurrently on a single topic. A single node can also publish or subscribe to multiple topics. On the other hand, services are one on one. A node can offer a service and other nodes may request an action on that service. Information about the action can also be received.

Rviz

Rviz is a ROS library that was used in this thesis to perform experiments. Rviz allows 3D visualization of the robot and the robot model can be visualized in Rviz using a URDF (Unified Robot Description Format) file. The other main use of Rviz is that the published topics can be visualized. In our case, the data of our MARG sensor was visualized in Rviz along with the robot.

TF

The Transformation Frames (TF) library was used to align the robot with the sensing module and also for visualization purposes. TF is a core ROS library that keeps track of coordinate frames. This library is helps describe the pose of an object or the pose of a robot link in space. Every link has a coordinate frame and they are connected to each other. Thus, any link can be described with respect to another connected reference frame. Tf allows to obtain these poses easily and it can also be visualized in Rviz.

3.3.3 Kinematics

Kinematics deals with obtaining the geometrical representation of a coordinate frame of any link on a robot with respect to a fixed coordinate frame. On a robotic manipulator, this coordinate frame is usually the tip of the robot which is the end-effector and the fixed coordinate frame is the base link. Forward kinematics is a method to obtain the pose of the end_effector by utilizing the angles of the joints of the robot. To explain the forward kinematics of the manipulator used in this project, Denavit-Hartenberg (DH) notation is used. Each link is described by two angles and two distance parameters. The entire structure of the manipulator can be defined using this notation once all the joints and link lengths are described.

The kinematic chain of n links is given by,

$$^{0}T_{n} = \prod_{i=1}^{n} {}^{i-1}T_{i}(\theta i) \tag{3.24}$$

where

$$^{i-1}T_{i} = [Z_{i}][X_{i}] \tag{3.25}$$

where

$$[Z_{i}] = Trans_{Z_{i}}(d_{i})Rot_{Z_{i}}(\theta i) \tag{3.26}$$

$$[X_{i}] = Trans_{X_{i}}(a_{i,i+1})Rot_{X_{i}}(\alpha_{i,i+1}) \tag{3.27}$$

where d_i, θi, $a_{i,i+1}$, $\alpha_{i,i+1}$ are the DH parameters.

The matrices associated with these are

$$Trans_{Z_{i}}(d_{i}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.28}$$

$$Rot_{Z_{i}}(\theta i) = \begin{bmatrix} \cos\theta i & -\sin\theta i & 0 & 0 \\ \sin\theta i & \cos\theta i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.29}$$

$$Trans_{X_{i}}(a_{i,i+1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i,i+1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.30}$$

$$Rot_{X_i}(\alpha_{i,i+1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_{i,i+1} & i\sin\alpha_{i,i+1} & 0 \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.31}$$

Using the above notation, the Kinematics from the world to the end_effector of our robot can be found.

$$^{e}T_{w} = {}^{w}T_{l1} * {}^{l1}T_{l2} * {}^{l2}T_{l3} * {}^{l3}T_{l4} * {}^{l4}T_{l5} * {}^{l5}T_{e} \tag{3.32}$$

Here, w, e, l1, l2, l3, l4, l5 refer to the world, end_effector, link1, link2, link3, link4, link5 respectively.

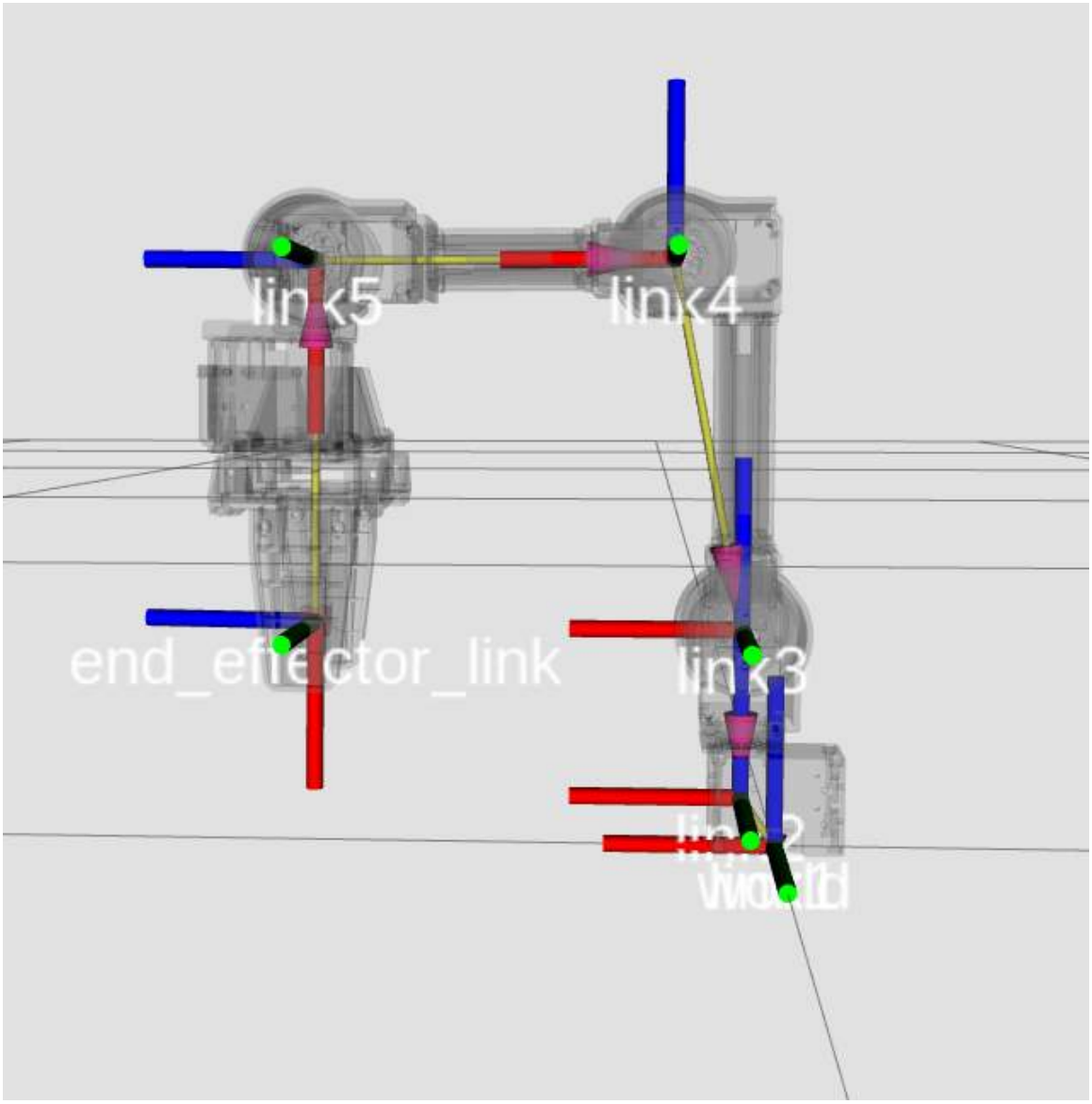


Figure 3.3: OpenMANIPULATOR-X Kinematic Chain

3.3.4 Transformation between Module and Base-Link Frames

To obtain the pose of the contact points, the transformation between the quaternion estimated from the MARG system and the base-link reference frame must be found. This is done by leveraging the forward kinematics of the robot and the quaternion representation from the MARG.

First, the Madgwick filter initiates the estimation.

$$^{S}q = [^{S}q_w, ^{S}q_x, ^{S}q_y, ^{S}q_z] \tag{3.33}$$

Next, we move the robot such that the x-axis of the end_effector is aligned with the x-axis of the world frame.

Using forward kinematics, the translation t_e and orientation q^e of the end_effector with respect to the base-link frame can be found.

$$t_e = [x_e, y_e, z_e] \tag{3.34}$$

$$^{E}q = [^{E}q_w, ^{E}q_x, ^{E}q_y, ^{E}q_z] \tag{3.35}$$

As the gripper is used to hold the sensing module, we are able to calculate the translation of the sensing module t_s with respect to the world frame of the robot.

$$t_s = [x_e + 0.02, y_e, z_e] = [x_s, y_s, z_s] \tag{3.36}$$

We take this translation and orientation of the sensor and publish a new TF broadcast **filter_orientation**.

$$filter\_orientation = [t_s, q^s] \tag{3.37}$$

To collect data using the sensor, we first need to align the sensor with the base of the robot. In our case, this is the world frame.

To align the orientation of the MARG to the world frame, we take the conjugate quaternion $^{S}q*$ of the initial MARG orientation estimate and multiply it with the orientation of **filter_orientation** where the robot is positioned so that the x-axis of the **end_effector** is aligned with **world**.

$$^{S}q* = [^{S}q_w, ^{S} - q_x, ^{S} - q_y, ^{S} - q_z] \tag{3.38}$$

We then publish this TF broadcast as **imu_world_aligned**.

$$imu\_world\_aligned = [t_s, q^s q^{s*}] \tag{3.39}$$

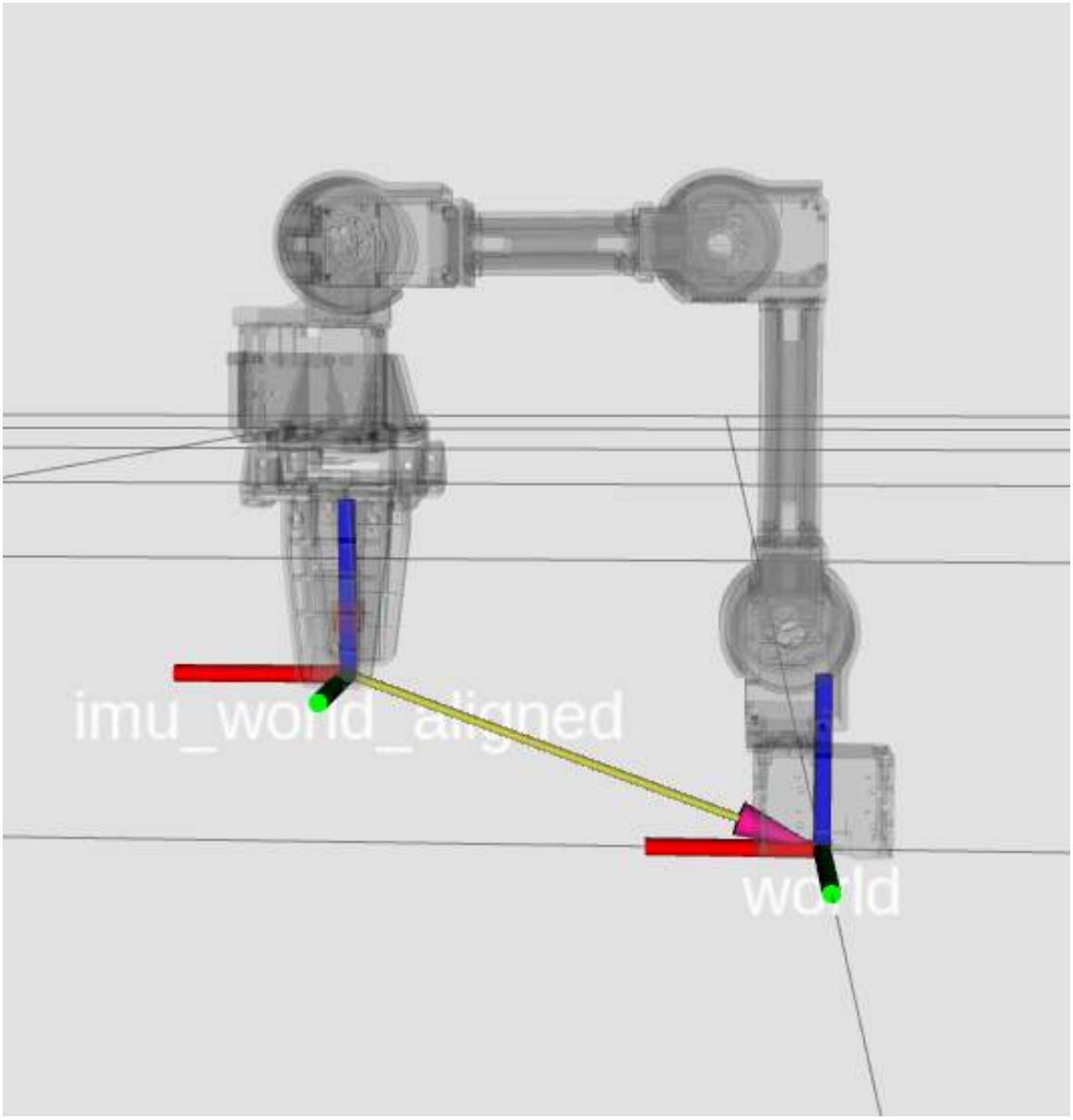


Figure 3.4: Sensor aligned TF

Chapter 4

Surface Estimation

4.1 Data Collection 35
4.2 Control Point Calculation 36
4.3 Surface Estimation 42

4.1 Data Collection

To collect a data point on an unknown surface, the manipulator moves the robotic manipulator holding the sensing module down while keeping a constant angle of approach for the end_effector. Once the sensing module makes contact with the surface, the module deforms as seen in Figure 4.1 and a data point is collected. A collected data point i has a translation and an orientation. The translation is represented as a 3d vector point $o_i = [x_i, y_i, z_i]$. The orientation is represented as a quaternion $q_i = [a_i, b_i, c_i, w_i]$ which we convert into the orthogonal normal (n_i) to the contact surface and two tangent vectors (u_i,v_i). This is represented as a matrix $V_i = [n_i, u_i, v_i]$. The tangent plane P_i of this contact point is defined by o_i and n_i.

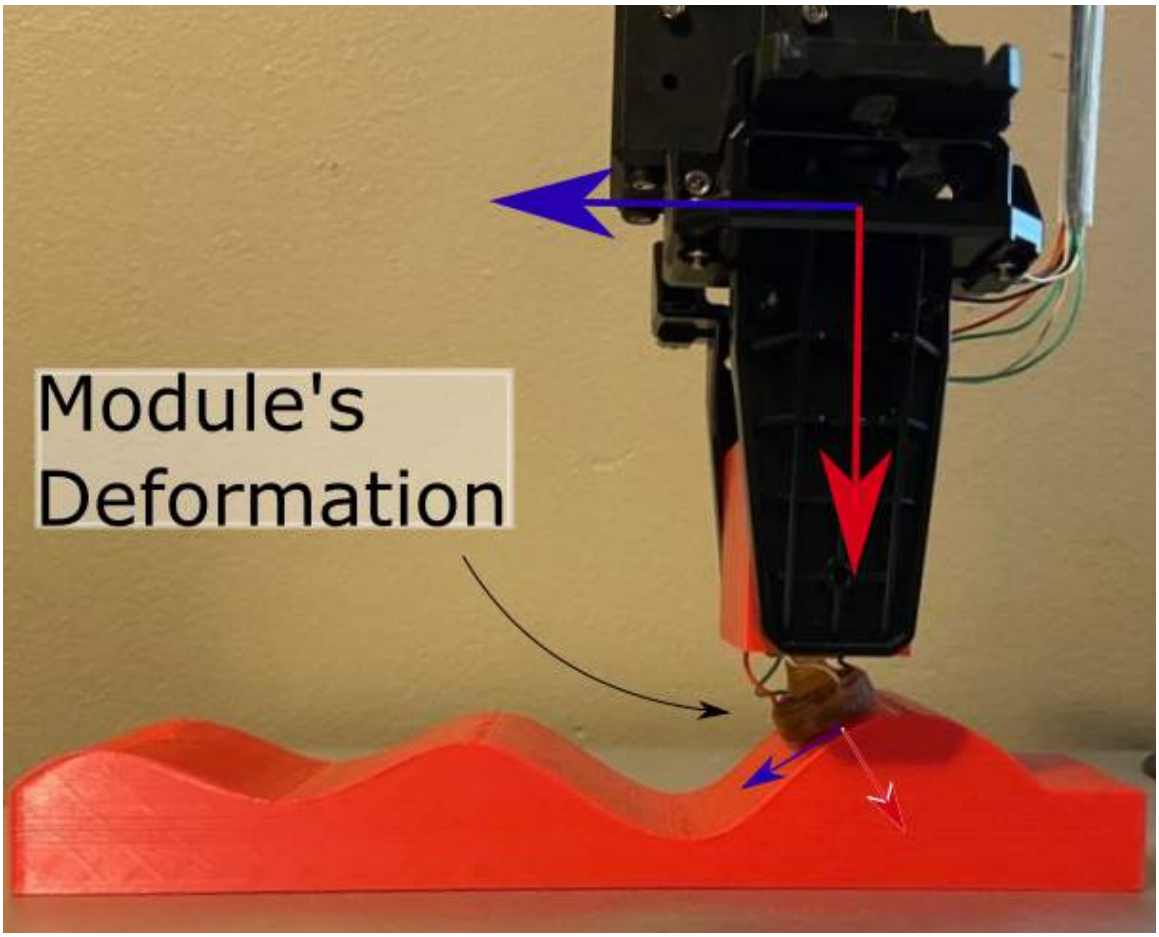


Figure 4.1: Data Collection

4.2 Control Point Calculation

Once 2 data points are collected, we calculate a control point between them that indicates the direction of the surface curvature. By calculating control points, we reduce the number of probes needed to estimate a surface. These control points are calculated by leveraging the surface normals and the knowledge of geometry.

Given 2 contact points $o_i = [x_i, y_i, z_i]$ and $o_j = [x_j, y_j, z_j]$ and their respective reference frames denoted by the matrices $V_i = [n_i, u_i, v_i]$ and $V_j = [n_j, u_j, v_j]$, our objective is to calculate a control point c_{ij}. To calculate this control point, the tangents v_i and v_j must be aligned without reorienting the planes that describe the contact. Figure 4.2, shows the contact points o_i and o_j respective reference frames.

The plane P_i is defined by the point $o_i = [x_i, y_i, z_i]$ and the normal $n_i = [a_i, b_i, c_i]$

and the plane P_j is defined by the point $o_j = [x_j, y_j, z_j]$ and the normal $n_j = [a_j, b_j, c_j]$.

To calculate c_{ij}, first, the point o_i must be projected onto the plane P_j. Let's call this projected point o_i'.

Consider a vector γ_i drawn between o_i and o_j which is on the plane P_j. This vector can be written as

$$\gamma_i = o_i - o_j \tag{4.1}$$

Expanding this into coefficients,

$$\gamma_i = [(x_i - x_j), (y_i - y_j), (z_i - z_j)] \tag{4.2}$$

We can get the scalar distance from o_i to the plane P_j along it's normal n_j by taking the dot product of γ_i and n_j.

$$dist = (x_i - x_j) * a_j + (y_i - y_j) * b_j + (z_i - z_j) * c_j \tag{4.3}$$

To get the projected point o_i', we multiply the normal n_j by the distance and subtract this vector from o_i

$$o_i' = o_i - dist * n_j \tag{4.4}$$

We draw a vector ϕ_j between o_i' and o_j to calculate the angle θ_j between v_j and ϕ_j. We then rotate v_j around the normal n_j by the angle θ_j to align itself to ϕ_j, while keeping it on P_j.

To perform this rotation, we take the rotation matrix R_j of the vector v_j and the angle θ_j.

$$R_j = \begin{bmatrix} t*x_j*x_j + c & t*x_j*y_j - z_j*s & t*x_j*z_j + y_j*s \\ t*x_j*y_j + z_j*s & t*y_j*y_j + c & t*y_j*z_j - x_j*s \\ t*x_j*z_j - y_j*s & t*y_j*z_j + x*s & t*z_j*z_j + c \end{bmatrix}$$

Here, $v_j = [x_j, y_j, z_j]$, $c = \cos(\theta_j)$, $s = \sin(\theta_j)$ and $t = 1 - c$

The rotated vector is the dot product of the above rotation matrix R_j and the normal n_j.

$$v_j = R_j.n_j \tag{4.5}$$

We then take a point along the rotated vector ϕ_j and attach the normal n_j to this point. To find the intersection point t_i between the $(\phi_j - n_j)$ vector and the plane P_i, we use the line-plane intersection equation.

Consider a point p on the plane P_i. As o_i is also on the plane, the equation of the plane can be written as

$$n_i.(p - o_i) \tag{4.6}$$

Considering the line $(\phi_j - n_j)$, the equation of this line passing through p can be written as

$$p = n_j + u(\phi_j - n_j) \tag{4.7}$$

Then, the equation of the intersection of this line $(\phi_j - n_j$ and the plane P_i can be written as

$$n.(n_j + u(\phi_j - n_j)) - o_i) = 0 \tag{4.8}$$

$$n.(n_j + u(\phi_j - n_j) = n.o_i \tag{4.9}$$

where

$$u = (n.(p - o_j))/(n.(o_k - o_j) \tag{4.10}$$

We draw a vector ϕ_i between t_i and o_i to calculate the angle θ_i between v_i and ϕ_i. We then rotate v_i around the normal n_i by the angle θ_i to align itself to ϕ_i, while keeping it on P_i.

To perform this rotation, we take the rotation matrix R_i of the vector v_i and the angle θ_i.

$$R_i = \begin{bmatrix} t * x_i * x_i + c & t * x_i * y_i - z_i * s & t * x_i * z_i + y_i * s \\ t * x_i * y_i + z_i * s & t * y_i * y_i + c & t * y_i * z_i - x_i * s \\ t * x_i * z_i - y_i * s & t * y_i * z_i + x_i * s & t * z_i * z_i + c \end{bmatrix}$$

Here, $v_i = [x_i, y_i, z_i]$, $c = \cos(\theta_i)$, $s = \sin(\theta_i)$ and $t = 1 - c$.

The rotated vector is the dot product of the above rotation matrix R_i and the normal n_i.

$$v_i = R_i.n_i \tag{4.11}$$

Now that the tangents v_i and v_j lie on the plane determined by the points o_i, o_i', o_j, t_i, we can calculate the control point between these 2 contact reference frames as the intersection between lines defined by v_i and v_j and respective contact points o_i and o_j. This point can be found using the 3D line-line intersection equation.

The intersection of 2 lines in 3D is the shortest distance between the 2 lines.

Consider a point o_l which lies on v_i and a point o_k which lies on v_j.

We can then consider 2 lines; a defined by the 2 points o_l and o_i and b defined by the 2 points o_k and o_j.

The line a can be defined by the equation

$$p_a = o_i + mu_a(o_l - o_i) \tag{4.12}$$

The line b can be defined by the equation

$$p_b = o_j + mu_b(o_k - o_j) \tag{4.13}$$

The shortest line between a and b will be perpendicular. Therefore,

$$(p_a - p_b).(o_l - o_i) = 0 \tag{4.14}$$

$$(p_a - p_b).(o_k - o_j) = 0 \tag{4.15}$$

Expanding these equations we get,

$$(o_i - o_j + mu_a(o_l - o_i) - mu_b(o_k - o_j)).(o_l - o_i) = 0 \tag{4.16}$$

$$(o_i - o_j + mu_a(o_l - o_i) - mu_b(o_k - o_j)).(o_k - o_j) = 0 \tag{4.17}$$

Expanding in term of coordinates,

$$d_{ijli} + mu_a d_{lili} - mu_b d_{kjli} = 0 \tag{4.18}$$

$$d_{ijkj} + mu_a d_{kjli} - mu_b d_{kjkj} = 0 \tag{4.19}$$

where

$$d_{mnop} = (x_m - x_n)(x_o - x_p) + (y_m - y_n)(y_o - y_p) + (z_m)(z_n)(z_o - z_p) \tag{4.20}$$

and

$$d_m nop = d_o pmn \tag{4.21}$$

Solving for mu_a and mu_b we get,

$$mu_a = (d_{ijkj} d_{kjli} - d_{ijli} d_{kjkj})/(d_{lili} d_{kjkj} - d_{kjli} d_{kjli}) \tag{4.22}$$

$$mu_b = (d_{ijkj} + muad_{kjli})/d4343 \tag{4.23}$$

Finally, the equation of the intersection point in 3D can be obtained by substituting mu_a and mu_b in Equation 4.18 or Equation 4.19.

An illustration of this process can be seen in Figure 4.2.

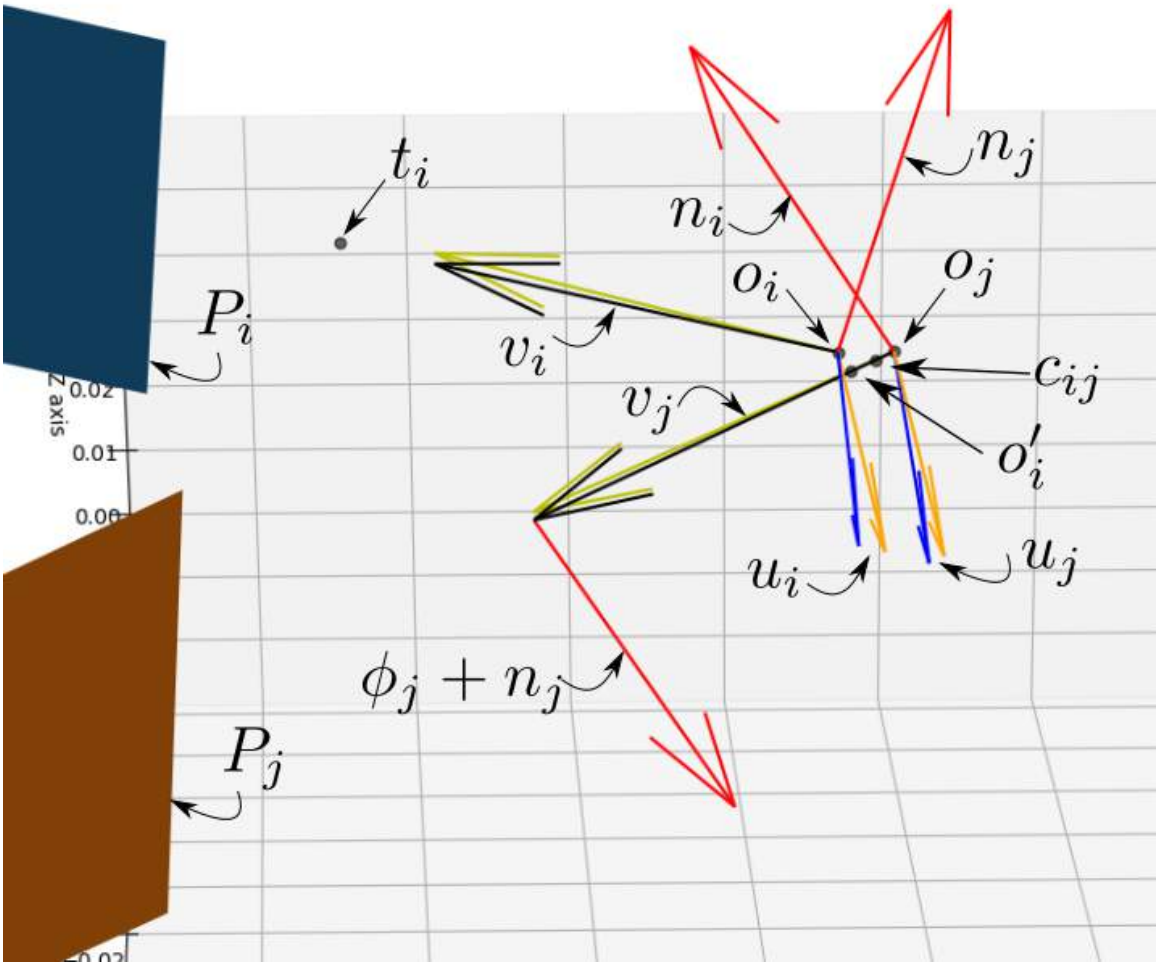


Figure 4.2: Control Point Calculation

4.3 Surface Estimation

To reconstruct the surface, we use Bezier surfaces [92]. As Bezier Surfaces rely on control points to better approximate its shape, we use a mixture of contact points and control points to reconstruct the surface. As there isn't any meaningful way to differentiate between contact points and control points, we cannot reconstruct the entire surface at once. Instead, we create 3*3 surface patches that make up the full surface.

Consider 4 contact points o_i, o_j, o_k and o_l. Using the method specified above, we can obtain the control points c_{ij}, c_{jk}, c_{lk} and c_{il}.

To obtain the control point in the middle, we have a few options.

1. We can take the mean of the 2 control points c_{il} and c_{jk} which can be annotated as c_{iljk}.
2. We can take the mean of the 2 control points c_{ij} and c_{lk} which can be annotated as c_{ijlk}.
3. We can calculate a new control point c_{jl} by taking the diagonal contact points o_j and o_l.
4. We can calculate a new control point c_{ik} by taking the diagonal contact points o_i and o_k.

These 4 methods give similar results and therefore we consider all 4 control points and choose the control point which has the highest value on the z-axis. This leads to a conservative estimate of the surface, that can prevent robots from advancing beyond the surface limits in dynamic exploration tasks.

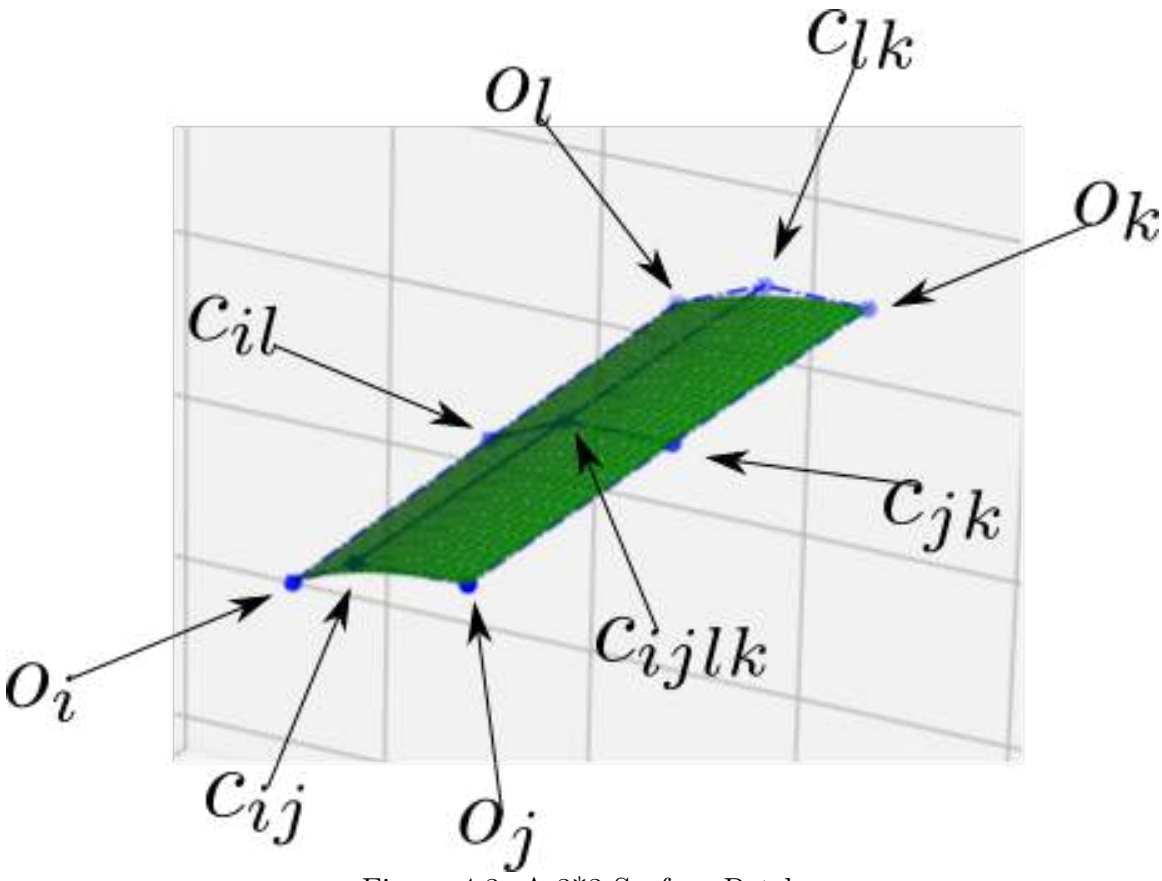


Figure 4.3: A 3*3 Surface Patch

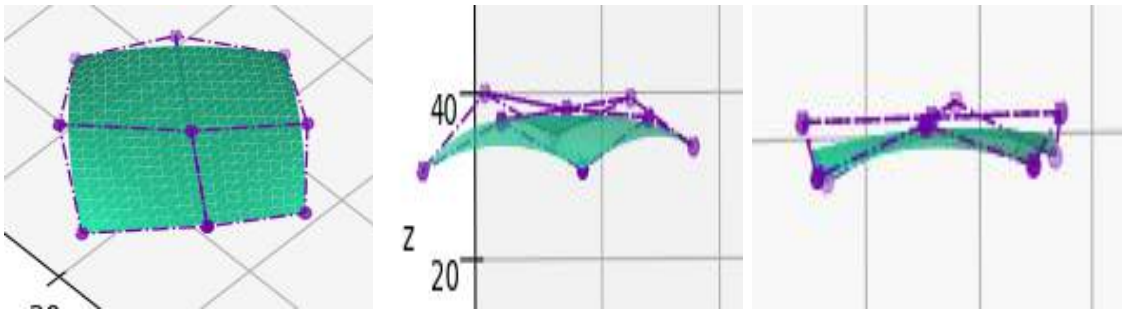
Using these 9 points, we create a surface patch in a 3*3 grid as seen in Figure 4.3. In this example, the highest value in the z-axis is the control point c_{ijlk}.

To further elaborate on the notion of taking the highest z-axis value, a small test was done on 2 extreme cases where 4 points were taken around a convex surface and around a concave surface.

Consider the following convex surface shown below in Figure 4.4. The 4 black dots represent where the contact points were taken on the surface.

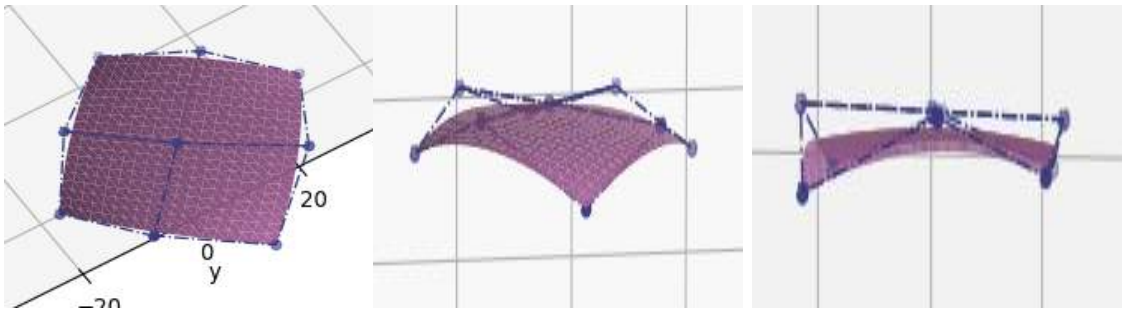
Figure 4.4: Convex Surface

The Bezier surfaces obtained using the above 4 methods for calculating the mid control point are shown below. Each figure contains 3 images; the first image is the view from the top, the second and third images are side views with some rotation.



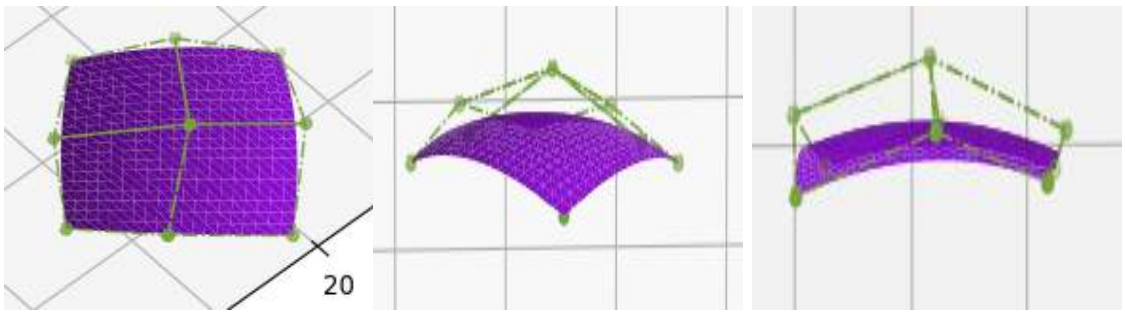
(a) View 1 (b) View 2 (c) View 3

Figure 4.5: Convex Surface - Method 1



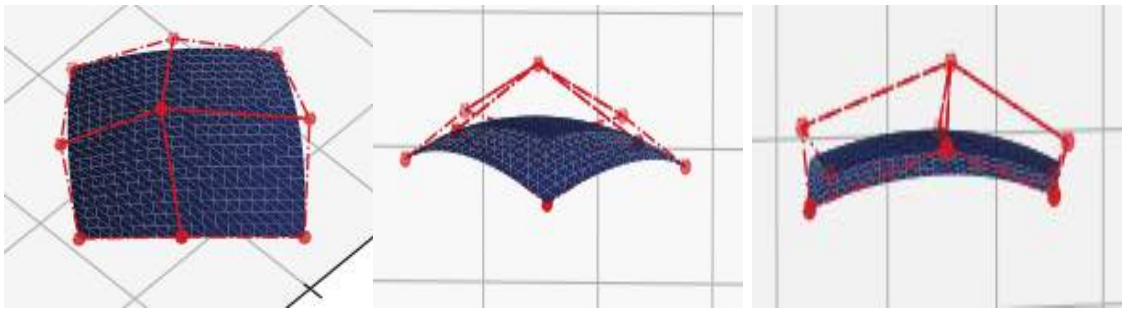
(a) View 1 (b) View 2 (c) View 3

Figure 4.6: Convex Surface - Method 2



(a) View 1 (b) View 2 (c) View 3

Figure 4.7: Convex Surface - Method 3



(a) View 1 (b) View 2 (c) View 3

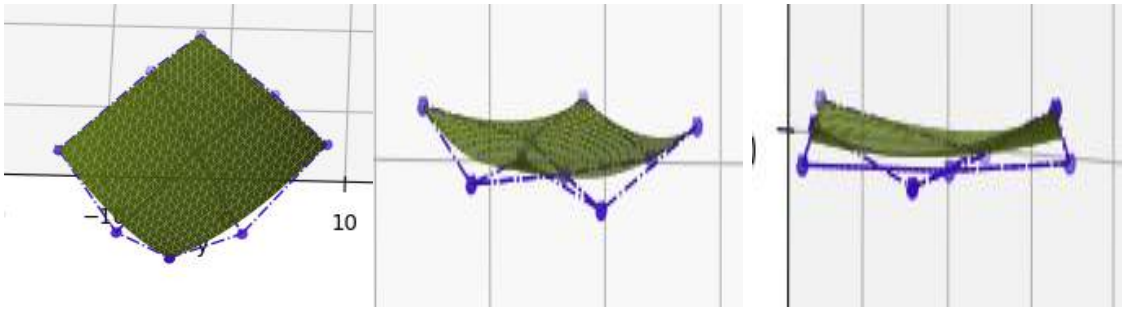
Figure 4.8: Convex Surface - Method 4

Method 1 and method 2 which is shown in Figure 4.5 and Figure 4.6 respectively give similar results when considering the z-axis of the mid control point. The z-value of Method 1 and Method 2 is visibly lower than the z-value of Method 3 and Method 4 shown in Figure 4.7 and Figure 4.8 respectively. The highest z-value is in Method 4 so we choose that control point for the estimation as it will allow the robot to operate safely when re-exploring the surface to obtain features such as texture.

Consider the following concave surface shown below in Figure 4.9. The 4 black dots represent where the contact points were taken on the surface.

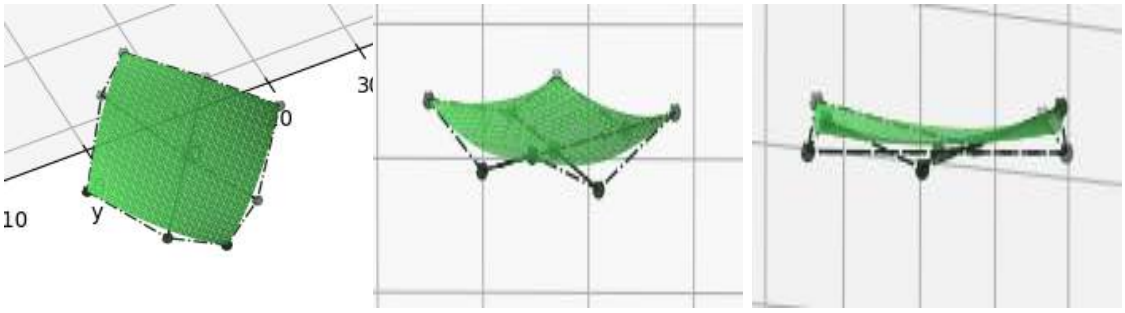
Figure 4.9: Concave Surface

The Bezier surfaces obtained using the above 4 methods are shown below. Similar to the figures above, each figure contains 3 images; the first image is the view from the top, the second and third images are side views with some rotation.



(a) View 1 (b) View 2 (c) View 3

Figure 4.10: Concave Surface - Method 1

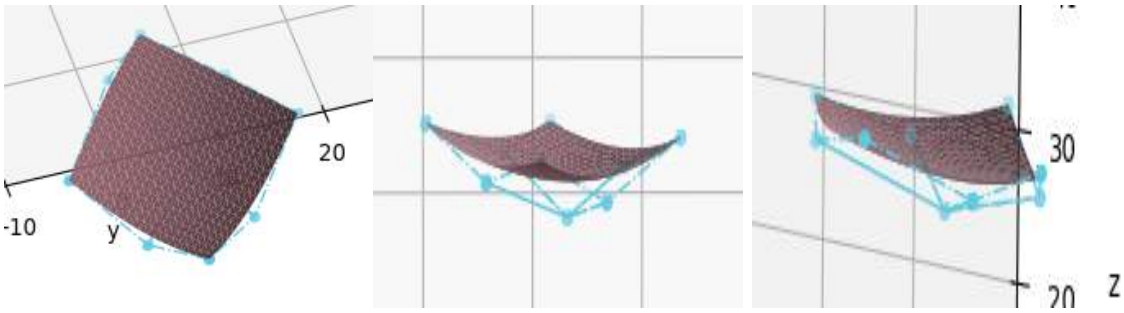


(a) View 1 (b) View 2 (c) View 3

Figure 4.11: Concave Surface - Method 2

(a) View 1 (b) View 2 (c) View 3

Figure 4.12: Concave Surface - Method 3



(a) View 1 (b) View 2 (c) View 3

Figure 4.13: Concave Surface - Method 4

In the results obtained from the concave surface, Method 1 and Method 2 represented in Figure 4.5 and Figure 4.6 respectively show higher z-values in their mid control points compared to Method 3 and Method 4 represented in Figure 4.7 and Figure 4.8 respectively. In this case, we also take the higher z-value which is Method 2 for the estimation. This leads to the robot not going beyond the limits of the surface on re-exploration which is essential for safe operation.

The results obtained from applying the above control point calculation method and the surface estimate method are discussed in the following Chapter.

Chapter 5

Results and Discussion

This section shows the results of the sensing apparatus and surface estimation method presented in this thesis.

5.1 Synthetic Surfaces 49
5.1.1 Surface 1 49
5.1.2 Surface 2 54
5.1.3 Surface 3 57
5.1.4 Surface 4 61
5.1.5 Surface 5 65
5.2 Everyday Objects 68
5.2.1 Vitamin C Bottle 68
5.2.2 Peanut Massage Ball 72
5.2.3 Hair Trimmer 75
5.2.4 Cream Bottle 79
5.2.5 Transparent Bottle 83
5.2.6 Screw 86
5.2.7 Earphone Case 90
5.2.8 Inhaler 93
5.3 Texture Exploration 98

In this section, we show the results obtained through surface estimation on 5 synthetic surfaces and 8 everyday objects. Points are collected as lines on the surfaces and each surface contains more than one such line. To create a Bezier surface, we take 2 contact points from one line and two contact points from the next line. We then calculate 4 control points between these points and a mid control point according to the methods described in the previous chapter. These 9 points make up a Bezier surface patch. Considering all the points, we estimate these Bezier surfaces patches to estimate the full surface.

For each surface estimated, we show the data collected for one line of points. For each line, we first show the data collected from the barometer, accelerometer, and gyroscope in the x-axis. Next, we show the translation of the end effector in the x-axis and the z-axis. Following that, we show the orientation in the form of Euler angles for the end-effector and the sensing module. Next, we show the pose data obtained; the orientation from the fused accelerometer and gyroscope data and the position from the kinematics of the robotic manipulator's end effector. Finally, the full estimation made up of Bezier surfaces patches is shown.

5.1 Synthetic Surfaces

5.1.1 Surface 1

Figure 5.1: Surface 1

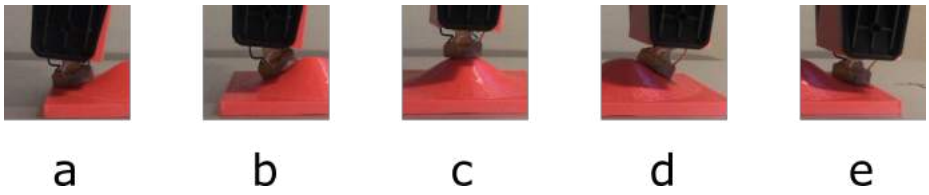


Figure 5.2: Surface 1 Contact Points

For this surface, we collected 5 lines of points with 5 points in each line. The middle line is shown in Figure 5.2.

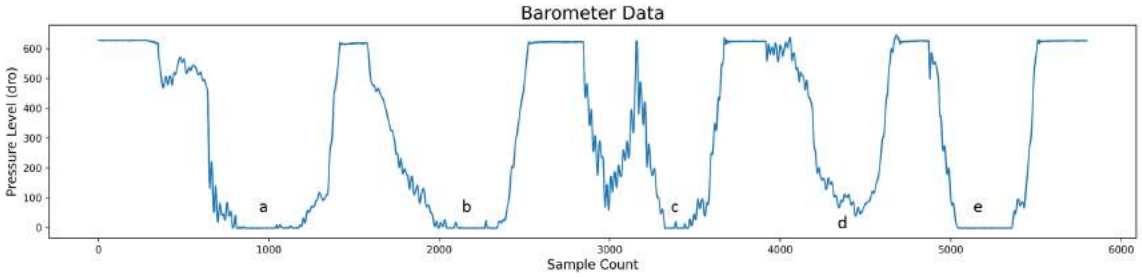


Figure 5.3: Surface 1 Barometer Graph

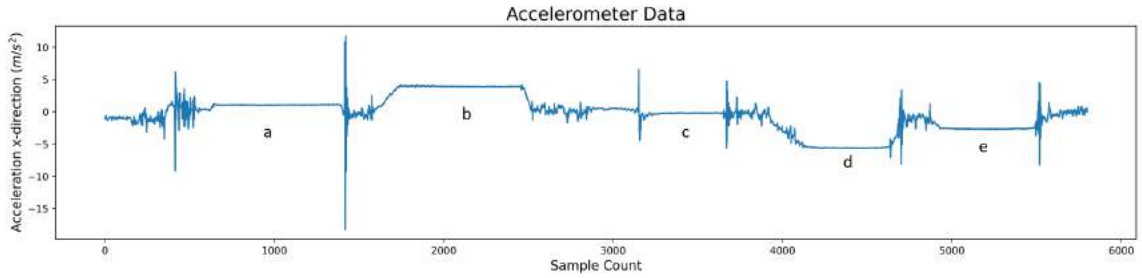


Figure 5.4: Surface 1 Accelerometer Graph

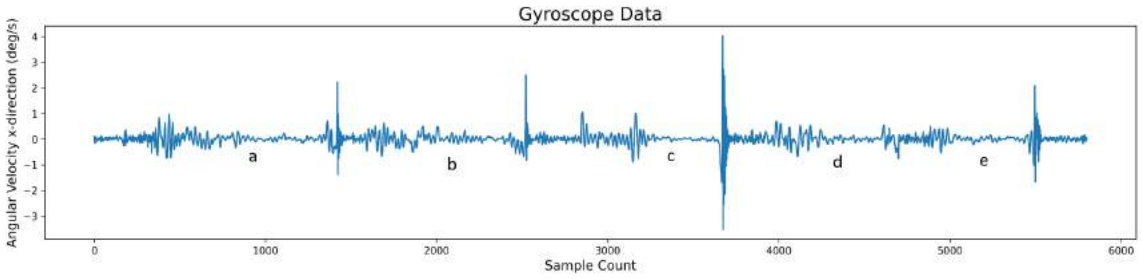


Figure 5.5: Surface 1 Gyroscope Graph

Figure 5.3 shows pressure data collected during the exploration of this middle line of the surface. The pressure is represented as a digital read-out value. The minimums

of the graph represent when the sensing module makes contact with the surface. Figure 5.4 shows the acceleration data collected from the accelerometer in meters per second and Figure 5.5 shows the angular velocity collected from the gyroscope in degrees per second during the exploration, both in the x-direction. Vibrations are seen when the sensing module is moved in between probes and the vibrations become stable when the sensing module comes into contact with the surface and stops its movement. This is reflected in both graphs.

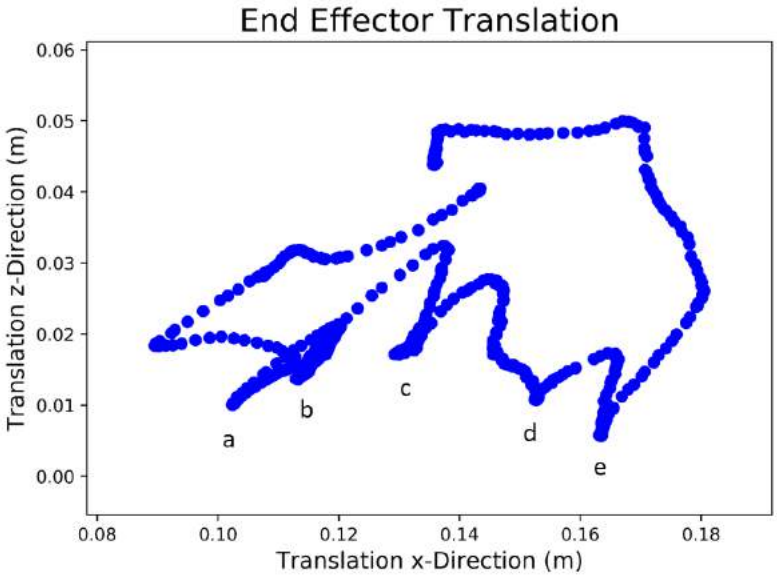


Figure 5.6: Surface 1 Translation Graph

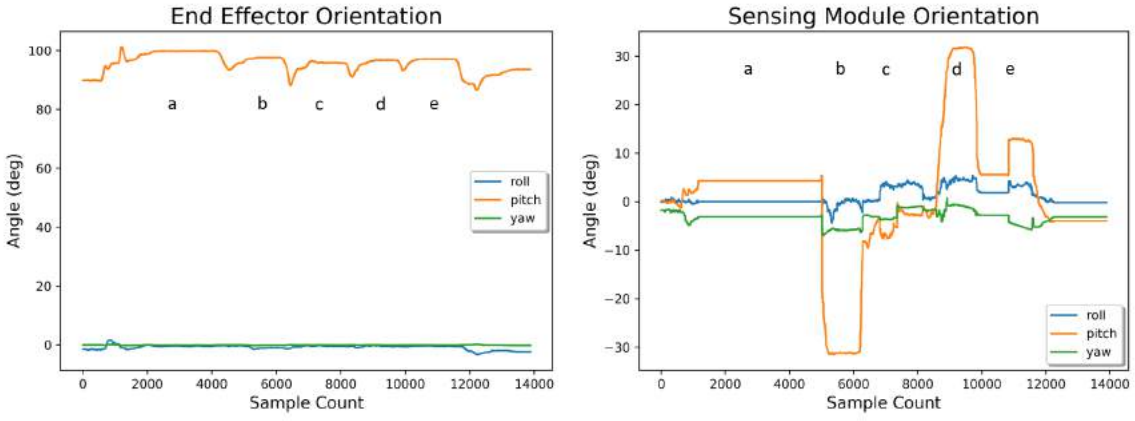


Figure 5.7: Surface 1 Orientation Graphs

Figure 5.6 shows the movement of the robotic end-effector during the exploration. The translation is shown in 2D between the x-axis and the z-axis and a clear idea of how the manipulator moves up and down to collect data can be observed. In Figure 5.6, the orientation of the end-effector and the sensing module is shown in terms of the Euler angles roll, pitch, and yaw. Here, it can be clearly seen that the pitch of the end-effector remains constant during the exploration while the pitch of the sensing module matches the normals of the surface. The pitch of the sensing module has a minimum at point B where the module deforms upwards while a maximum is seen at point D where the module deforms downwards.

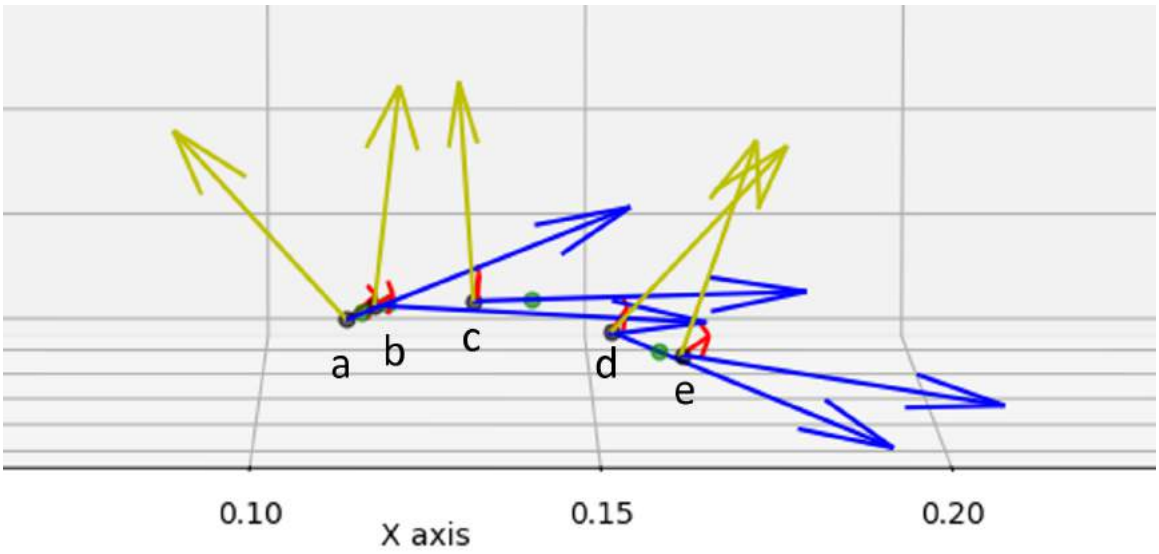


Figure 5.8: Surface 1 Markers

By fusing the data from the accelerometer and the gyroscope while also taking into account the position of the end-effector, Figure 5.8 shows the position and orientation of the collected data points. The green points between the contact points are the calculated control points.

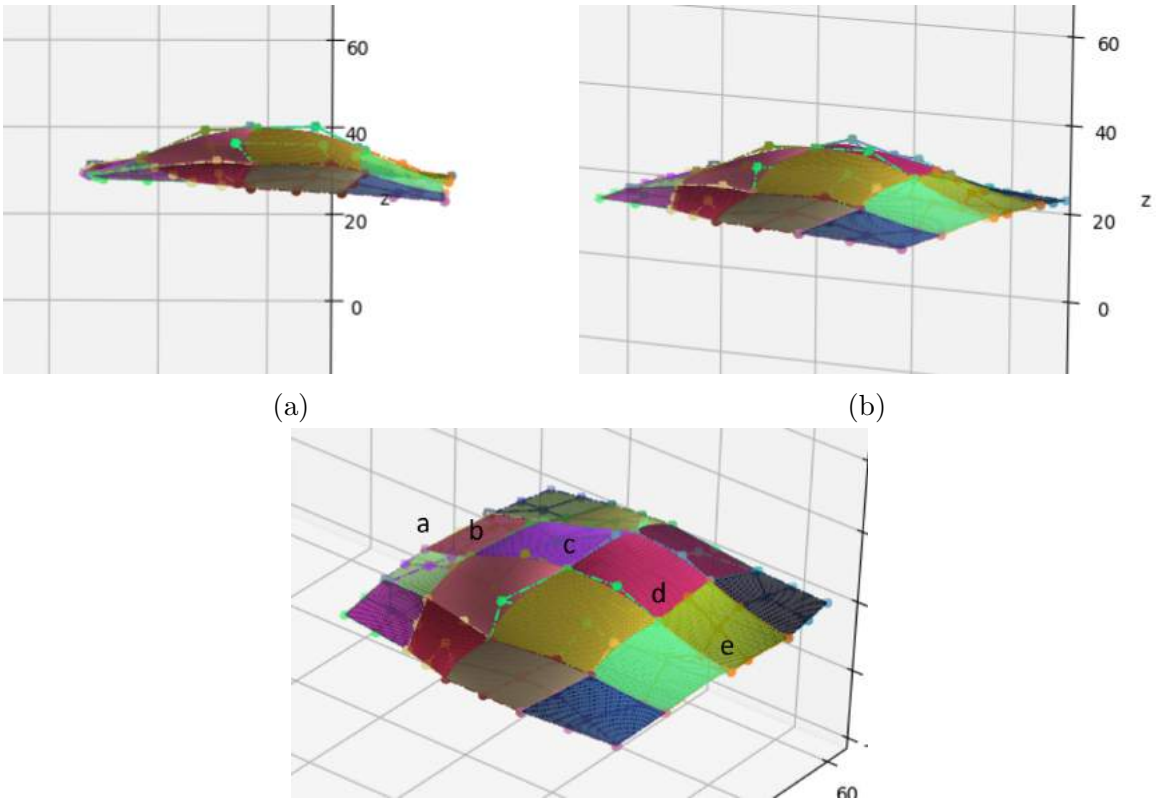


Figure 5.9: Surface 1 Estimation

Figure 5.9 shows the full estimate of the concave surface. As the full surface is made up of 5 lines of points with 5 points in each line, there will be 16 surface patches that make up the surface. Each surface patch is shown in a different color and they are made up of 4 contact points and 5 calculated control points. The lines indicate the curvature of each Bezier surface patch. The estimation moves up in the middle similar to the actual surface proving that this method is able to estimate a surface with a peak.

5.1.2 Surface 2

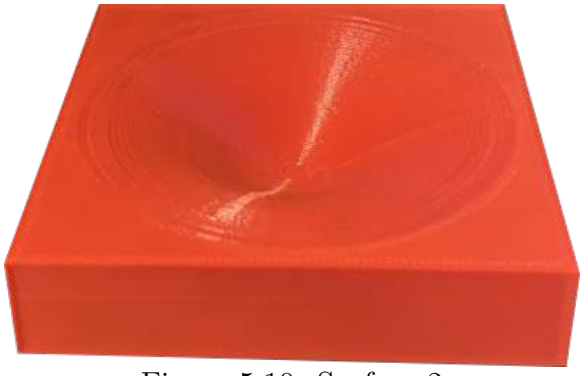


Figure 5.10: Surface 2

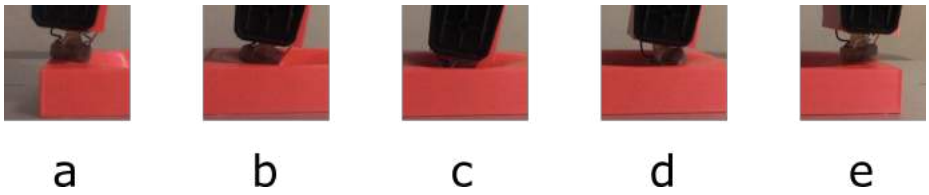


Figure 5.11: Surface 2 Contact Points

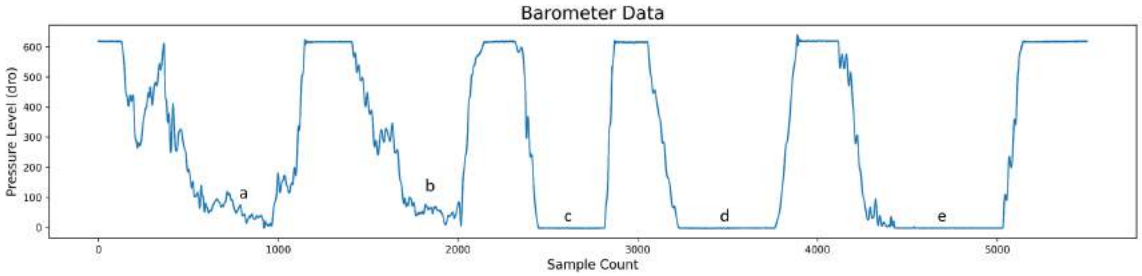


Figure 5.12: Surface 2 Barometer Graph

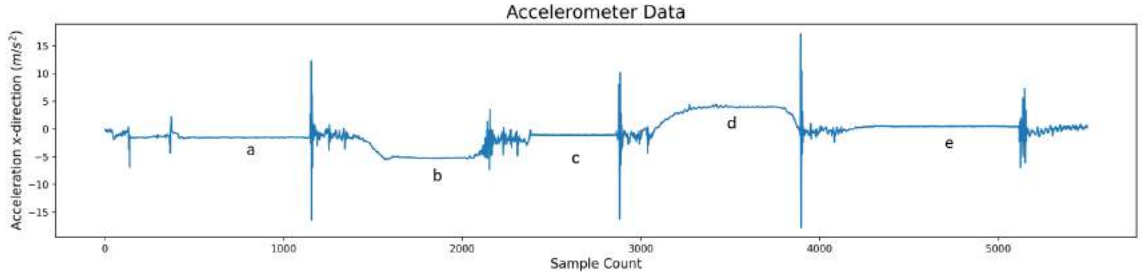


Figure 5.13: Surface 2 Accelerometer Graph

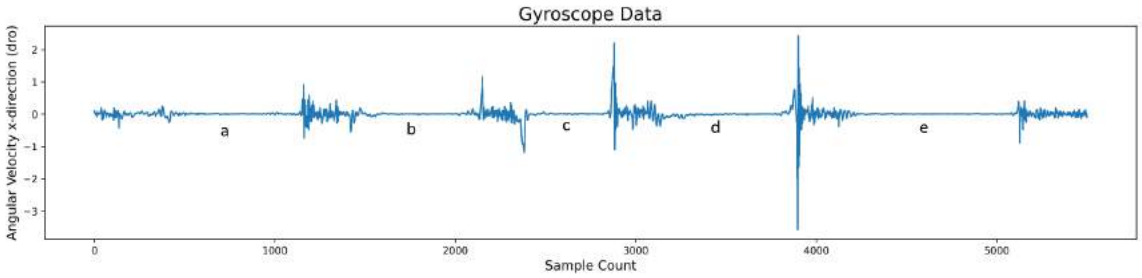


Figure 5.14: Surface 2 Gyroscope Graph

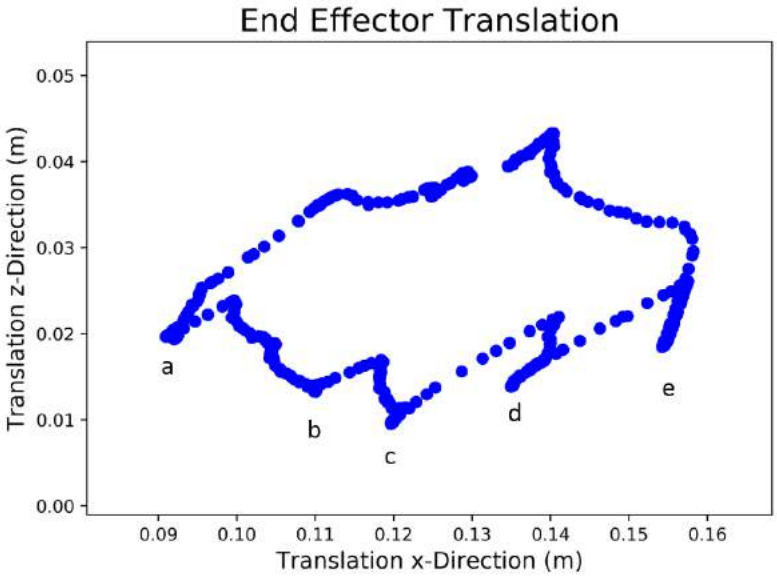


Figure 5.15: Surface 2 Translation Graph

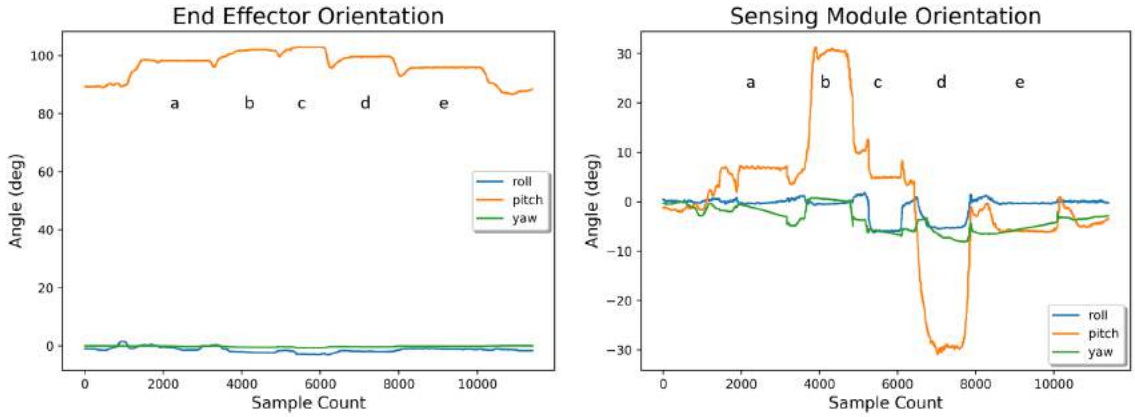


Figure 5.16: Surface 2 Orientation Graphs

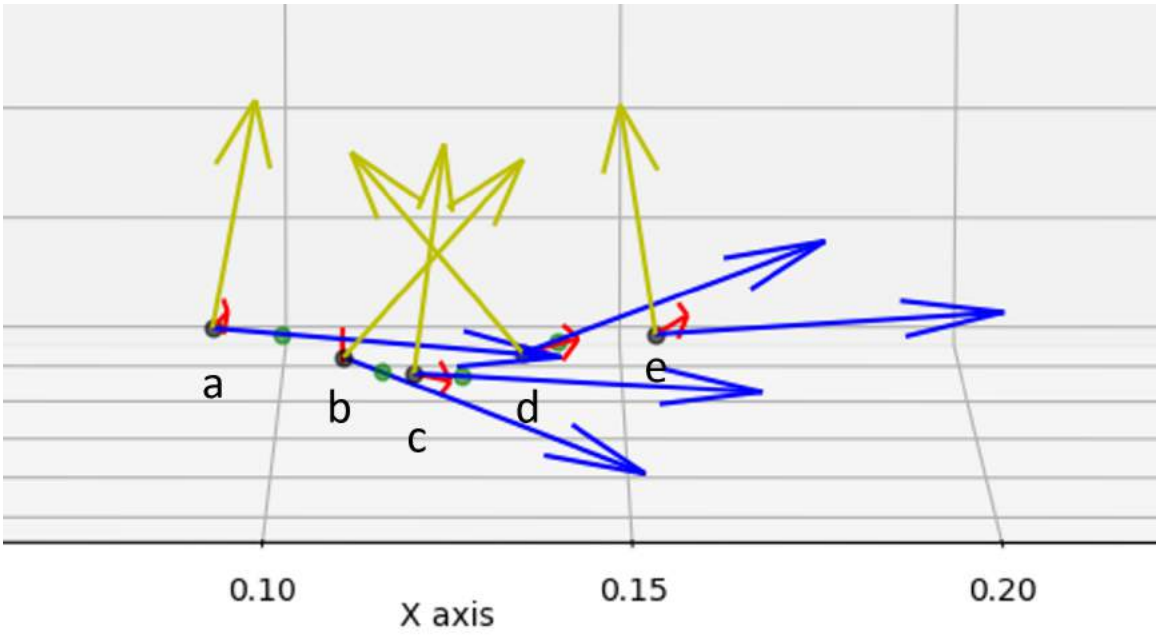
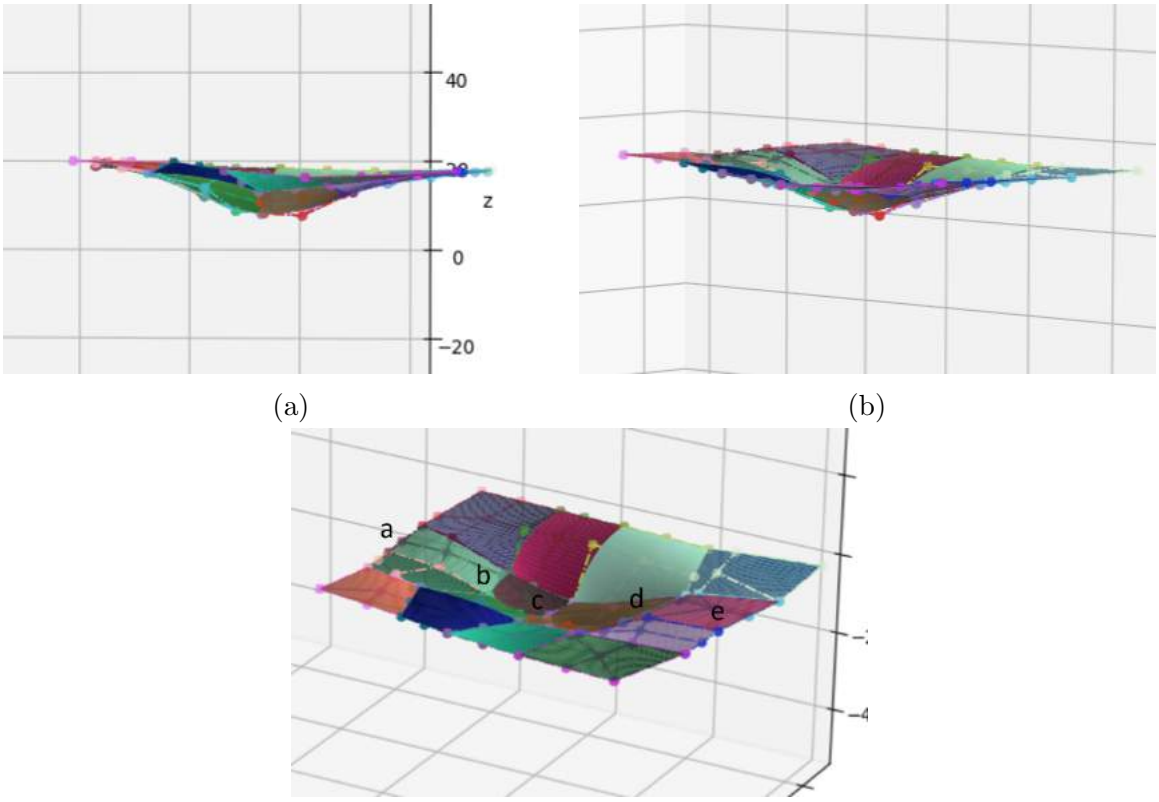


Figure 5.17: Surface 2 Markers



(a) (b)

(c)

Figure 5.18: Surface 2 Estimation

Similar to surface 1, we collected 5 lines of points with 5 points in each line for this concave surface. The middle line of points is shown in Figure 5.11. The corresponding data obtained from the barometer, accelerometer and gyroscope of the exploration of this middle line is shown in Figure 5.12, Figure 5.13 and Figure 5.14 respectively. The translation of the end-effector is shown in Figure 5.15 while the orientation of the end-effector and the sensing module is shown in Figure 5.16 and the pose of the collected data points is shown in Figure 5.17. Conversely to the convex surface, the pitch of the sensing module shows opposite peaks with point B having a maximum and point D having a minimum. The estimated surface is shown in different angles in Figure 5.18 which is quite similar to the actual surface showing that surfaces with downward peaks can be estimated using this approach.

5.1.3 Surface 3

Figure 5.19: Surface 3

Figure 5.20: Surface 3 Contact Points

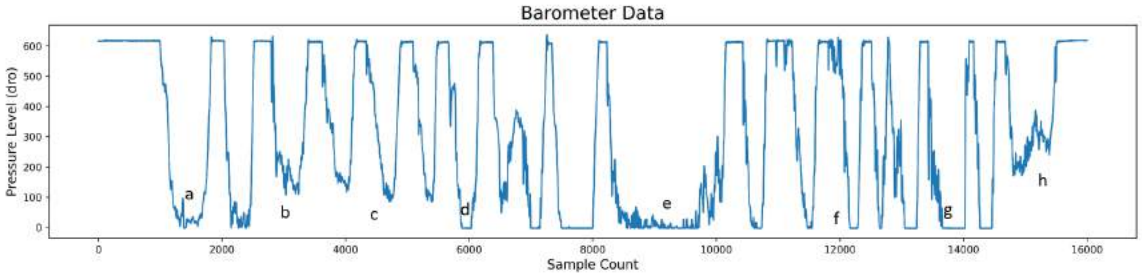


Figure 5.21: Surface 3 Barometer Graph

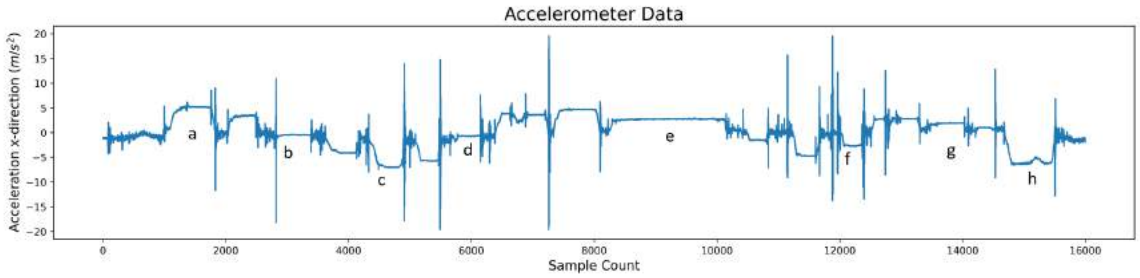


Figure 5.22: Surface 3 Accelerometer Graph

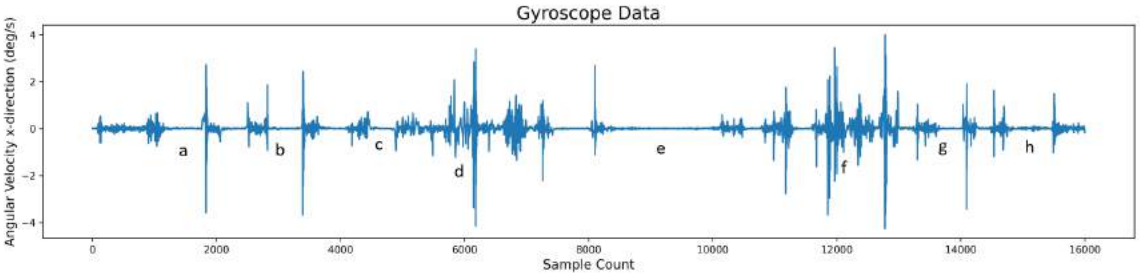


Figure 5.23: Surface 3 Gyroscope Graph

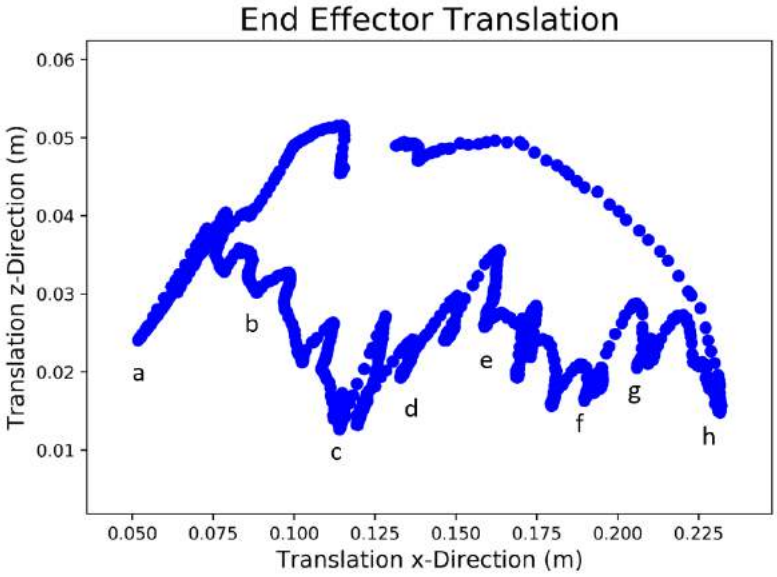


Figure 5.24: Surface 3 Translation Graph

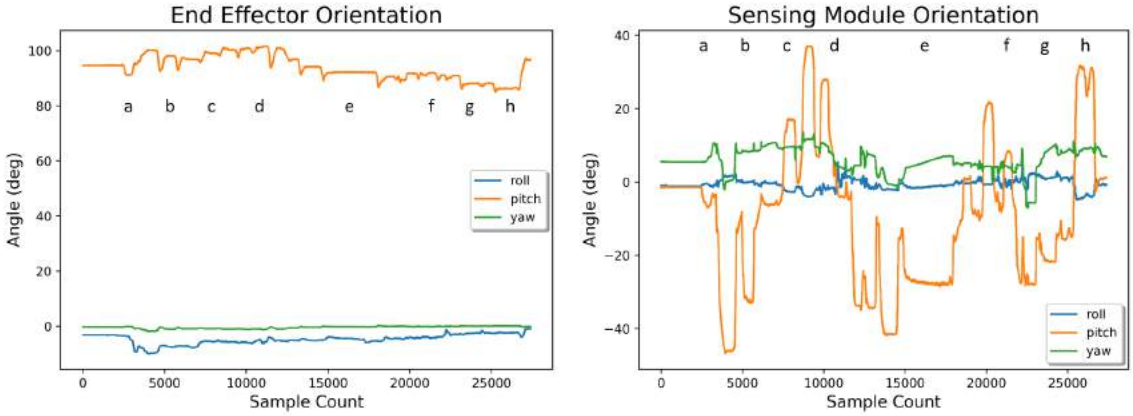


Figure 5.25: Surface 3 Orientation Graphs

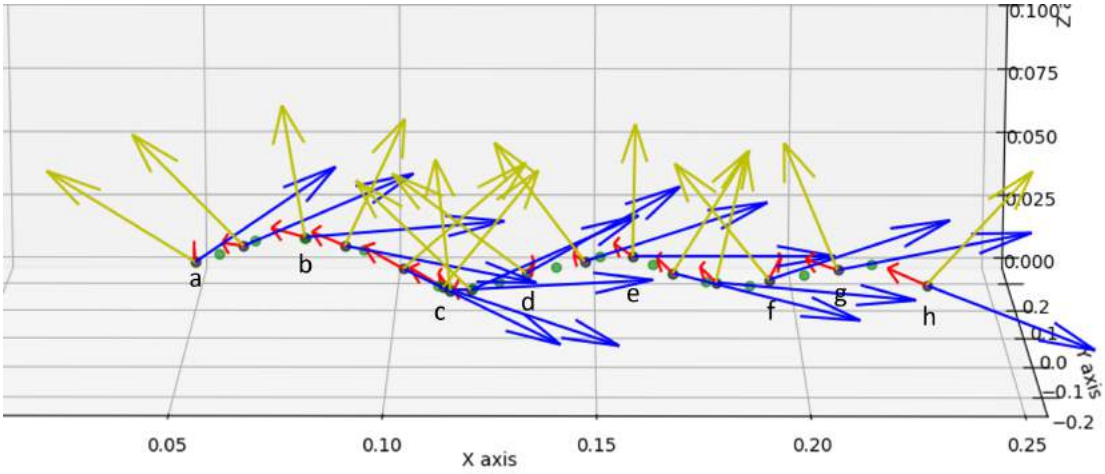
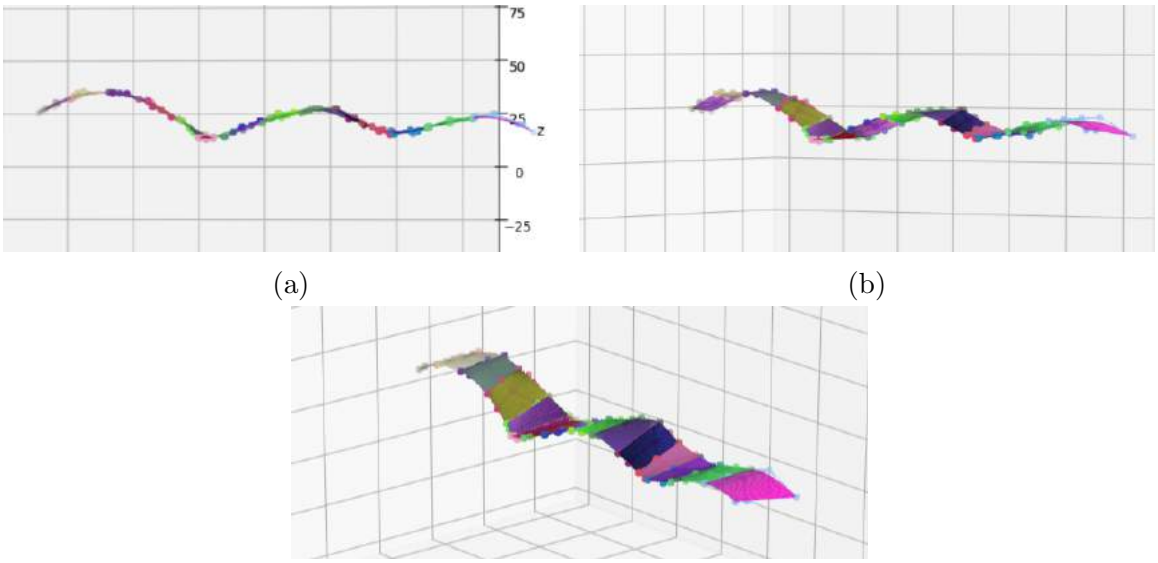


Figure 5.26: Surface 3 Markers



(a) (b)

(c)

Figure 5.27: Surface 3 Estimation

For this surface, 2 lines of points were collected with 15 points for each line. One such line which shows 8 such collected points is shown in Figure 5.20. The corresponding graphs are shown in Figure 5.21, Figure 5.22, Figure 5.23, Figure 5.24, Figure 5.25 and the graphs are annotated where the 8 specific contact points were made. In Figure 5.25, it can be seen that the end-effector does not change much while the pitch of the sensing module changes according to the deformation of the sensor along the surface. Contact points A and B have negative values for the pitch while C and D have positive values as the sensing module deforms upwards and downwards respectively. A similar pattern can be seen for the rest of the surface. The pose data along with the calculated control points are shown in Figure 5.26 and the estimation of surface 3 in 3 different angles is shown in Figure 5.27. The curves of this surface can very clearly be seen in the estimation and are almost identical to the actual surface.

5.1.4 Surface 4

Figure 5.28: Surface 4

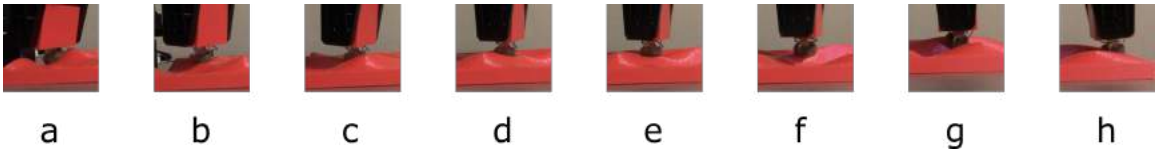


Figure 5.29: Surface 4 Contact Points

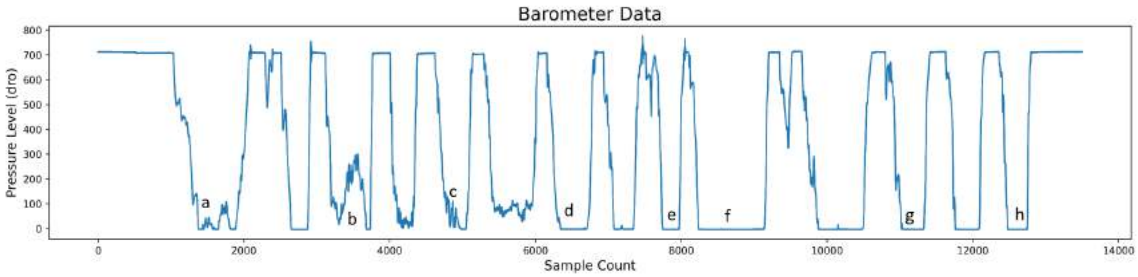


Figure 5.30: Surface 4 Barometer Graph

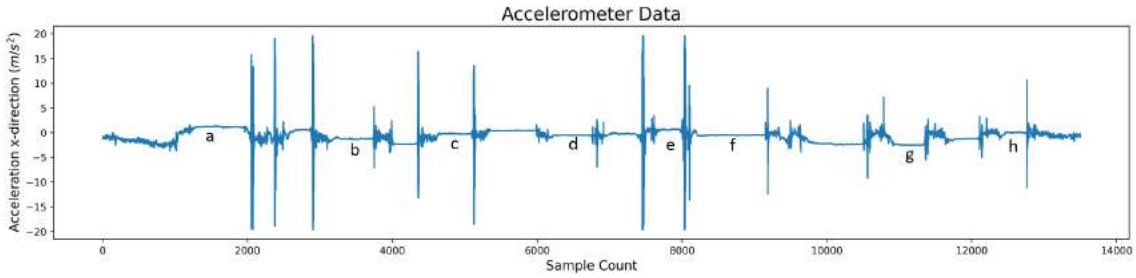


Figure 5.31: Surface 4 Accelerometer Graph

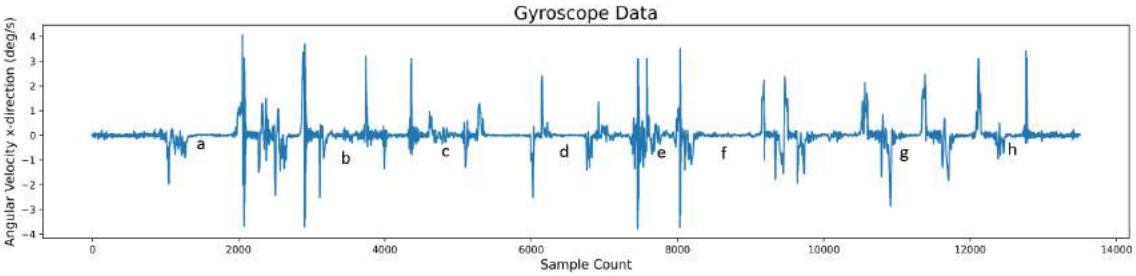


Figure 5.32: Surface 4 Gyroscope Graph

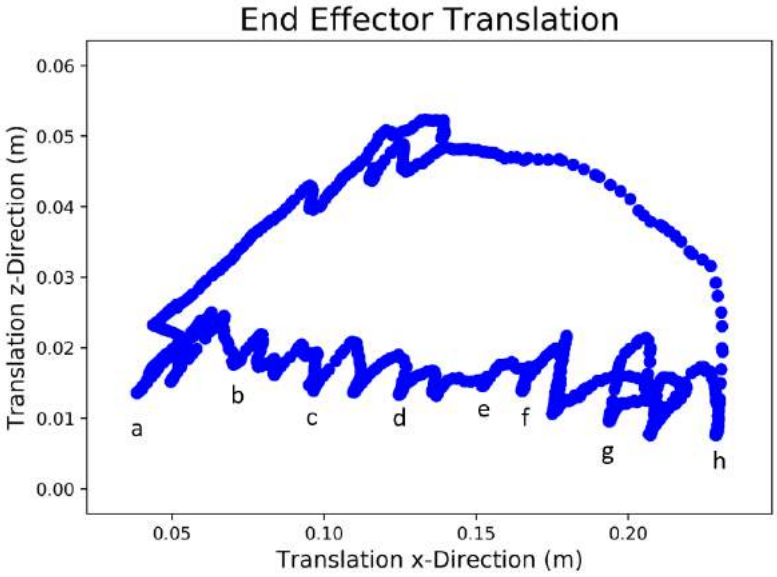


Figure 5.33: Surface 4 Translation Graph

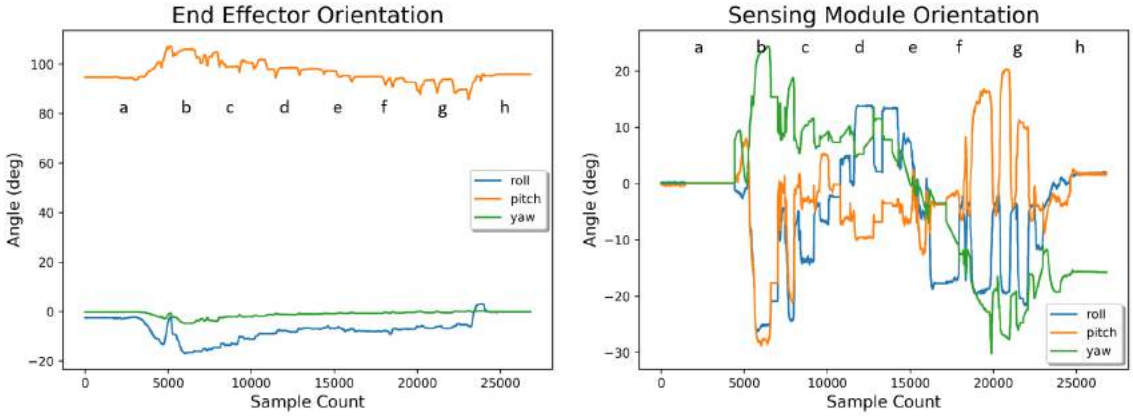


Figure 5.34: Surface 4 Orientation Graphs

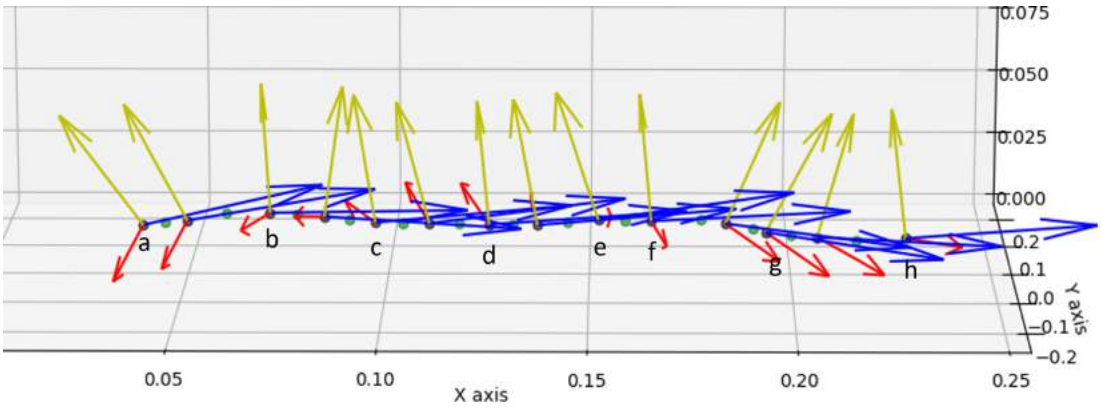
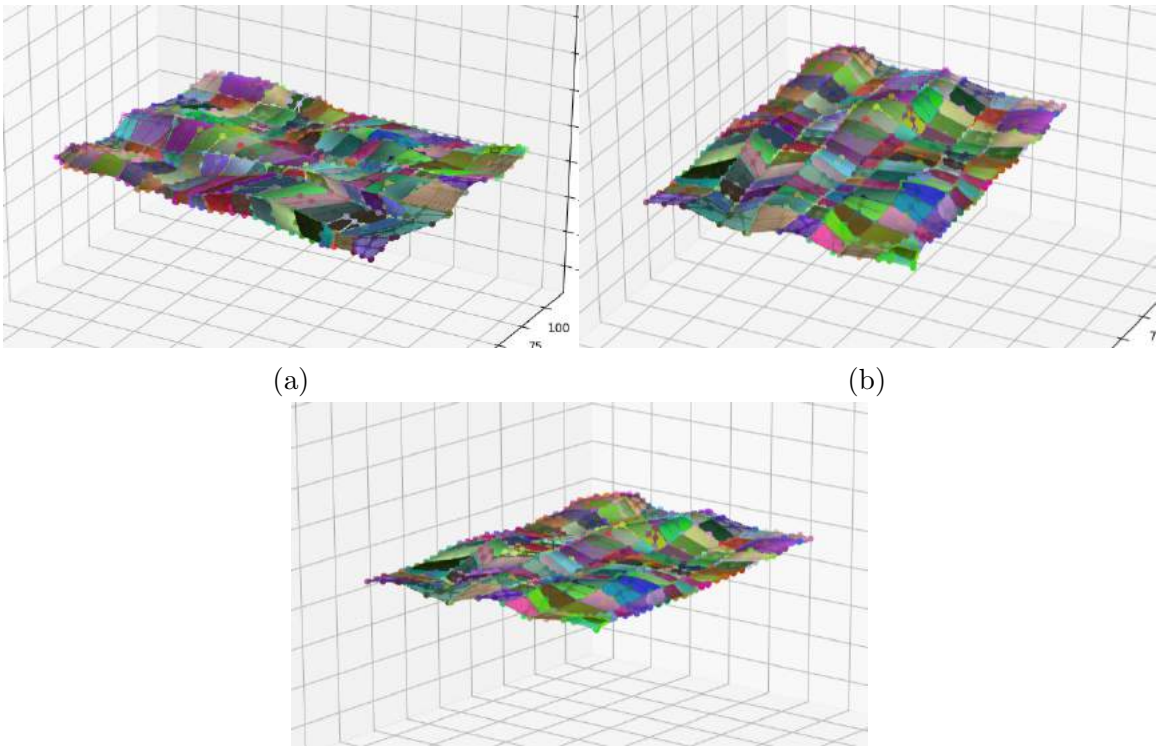


Figure 5.35: Surface 4 Markers



(a) (b)

(c)

Figure 5.36: Surface 4 Estimation

This surface is very complex with many curves and bumps. To calculate its estimate, we collected 9 lines of points with 14 points in each line. The 5th line of points collected is represented in Figure 5.29 showing 8 of the 14 points collected. The corresponding graphs are shown in Figure 5.30, Figure 5.31, Figure 5.32, Figure 5.33, Figure 5.34 and the graphs are annotated where the 8 specific contact points were made. The contact pose data along with the calculated control points are shown in Figure 5.35 and the estimation of the surface in 3 different angles is shown in Figure 5.36. For this surface, a total of 104 surface patches were used to estimate the full surface. The surface was estimated quite well considering that there were many curves with different curvatures.

5.1.5 Surface 5

Figure 5.37: Surface 5

Figure 5.38: Surface 5 Contact Points

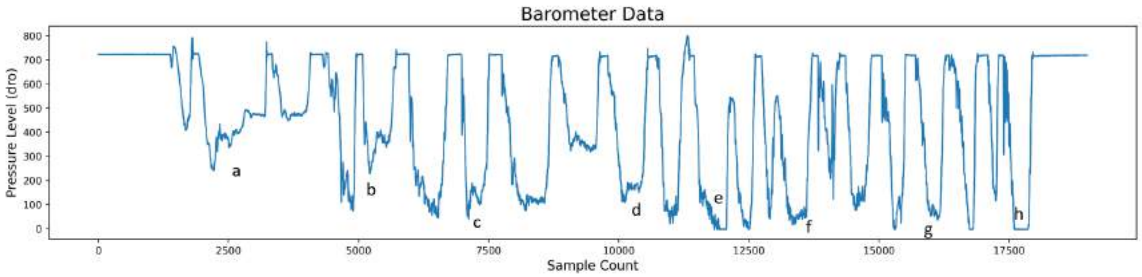


Figure 5.39: Surface 5 Barometer Graph

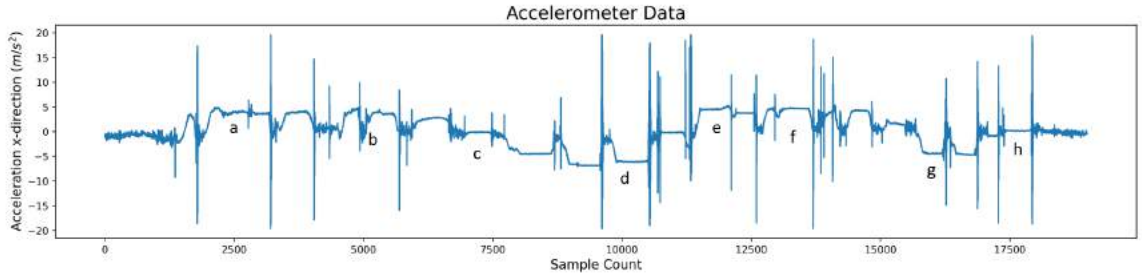


Figure 5.40: Surface 5 Accelerometer Graph

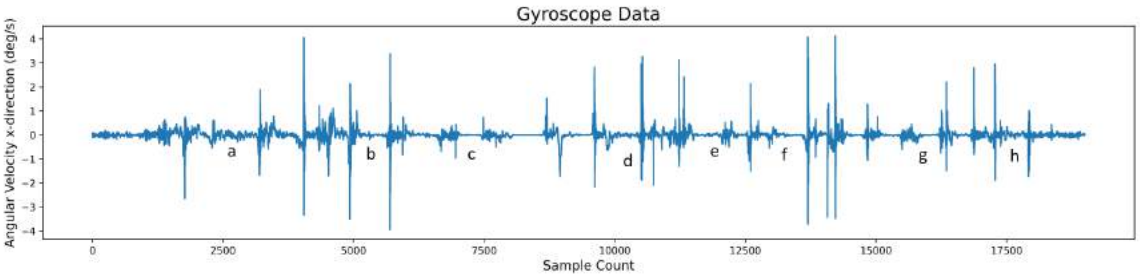


Figure 5.41: Surface 5 Gyroscope Graph

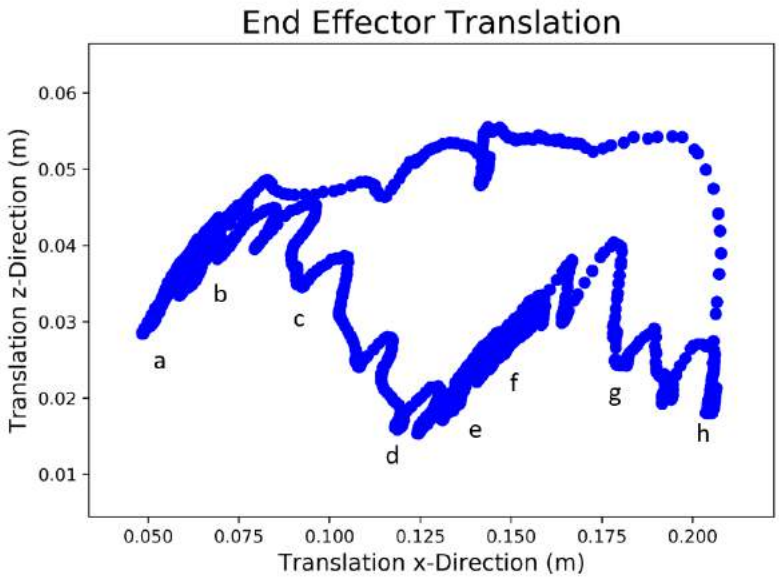


Figure 5.42: Surface 5 Translation Graph

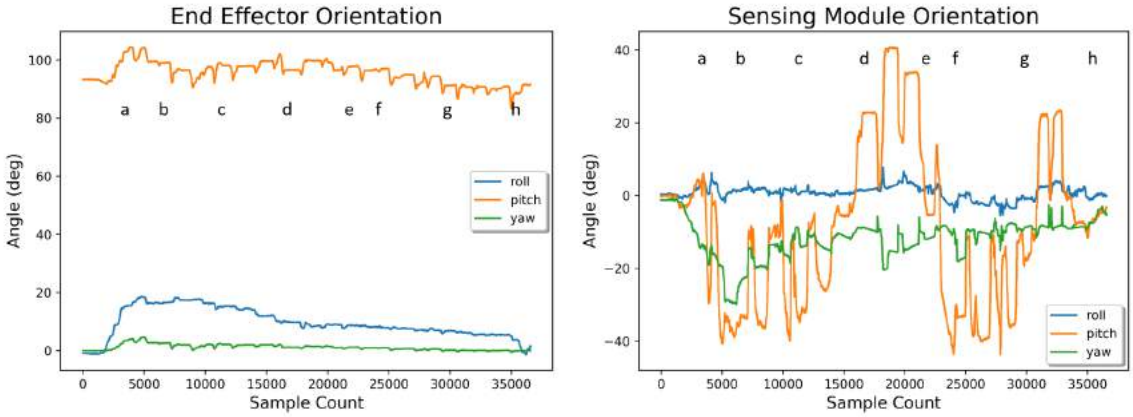


Figure 5.43: Surface 5 Orientation Graphs

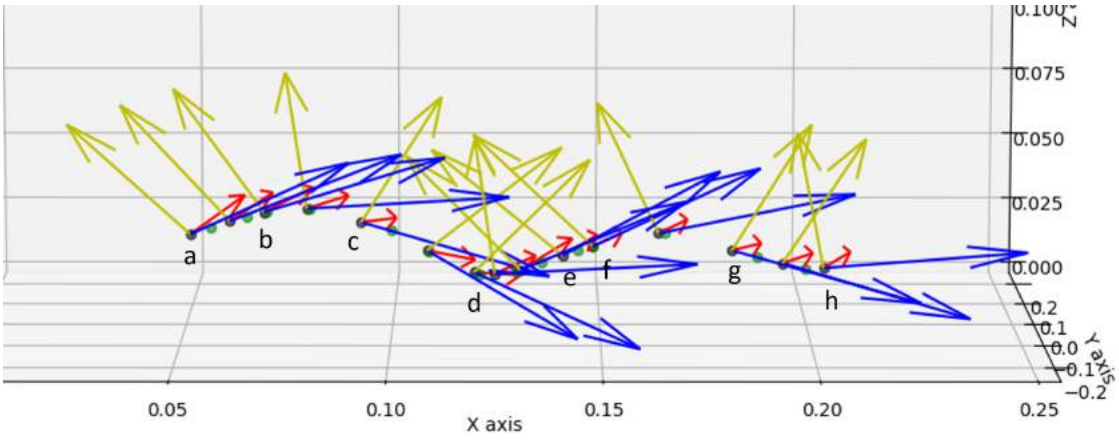
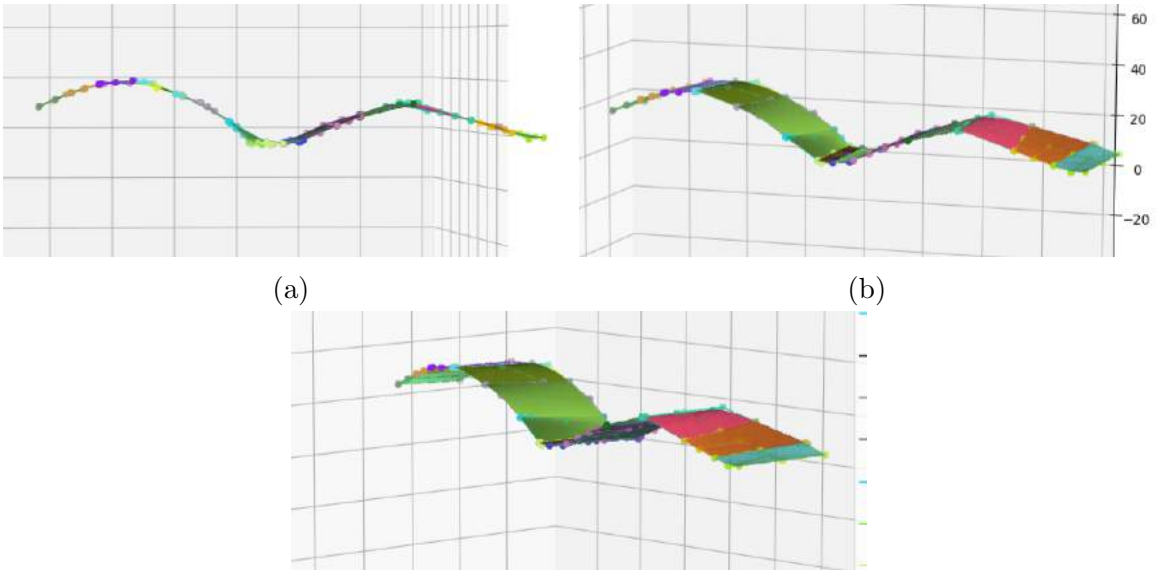


Figure 5.44: Surface 5 Markers



(a) (b)

(c)

Figure 5.45: Surface 5 Estimation

For surface 5, we collected 8 lines of points with 14 points in each line, and one line is shown in Figure 5.38 showing 8 of the 14 points collected. The corresponding graphs are shown in Figure 5.39, Figure 5.40, Figure 5.41, Figure 5.42, Figure 5.43 and the graphs are annotated where the 8 specific contact points were made. The contact pose data is shown in Figure 5.44 and the estimation of the surface in 3 different angles is shown in Figure 5.45. Surface 5 is very similar to Surface 3 and shows similar results in the graphs and estimations. Surface 5 was 3D printed with different textures which is discussed later in this chapter.

5.2 Everyday Objects

5.2.1 Vitamin C Bottle

(a) (b)

Figure 5.46: Vitamin C Bottle

Figure 5.47: Vitamin C Bottle Contact Points

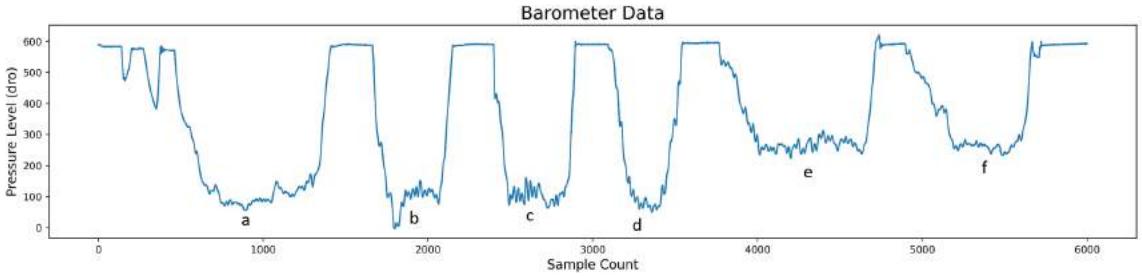


Figure 5.48: Vitamin C Bottle Barometer Graph

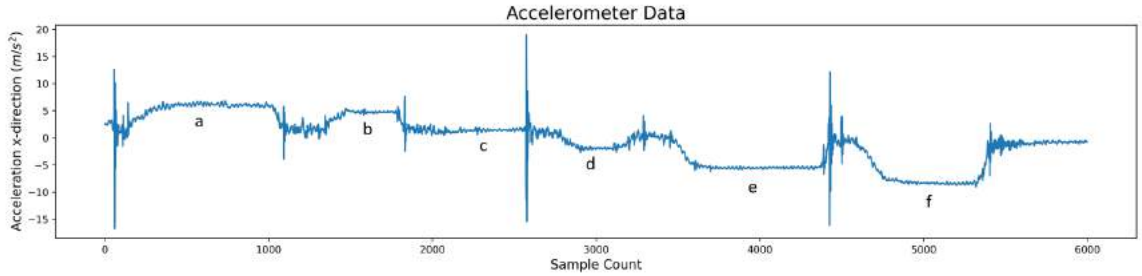


Figure 5.49: Vitamin C Bottle Accelerometer Graph

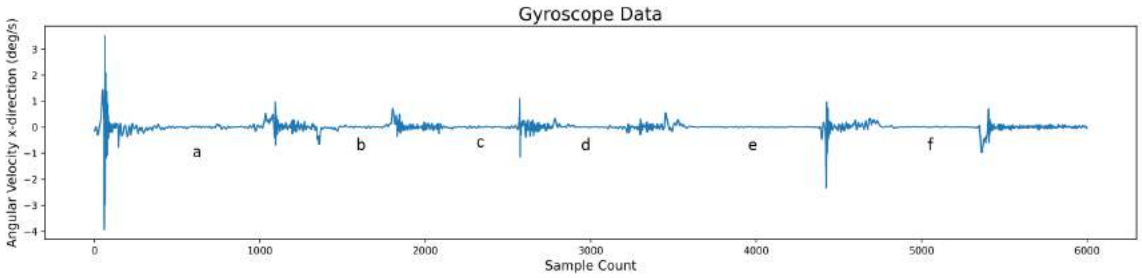


Figure 5.50: Vitamin C Bottle Gyroscope Graph

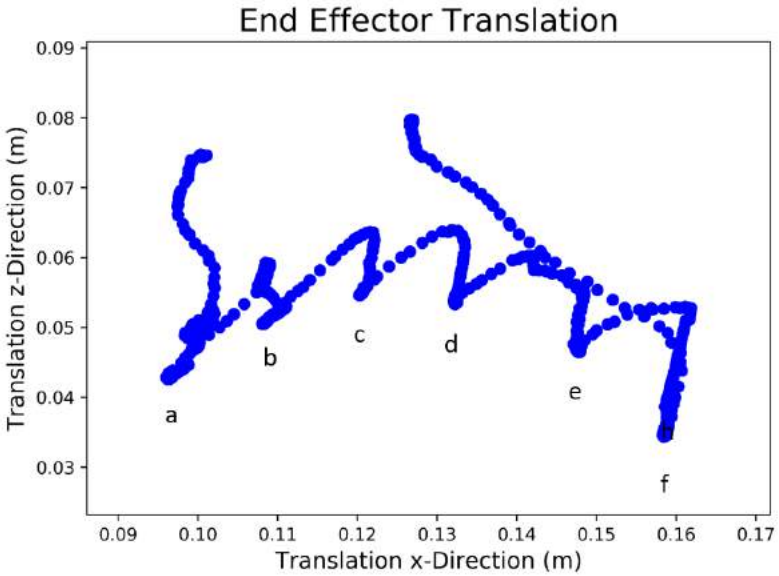


Figure 5.51: Vitamin C Bottle Translation Graph

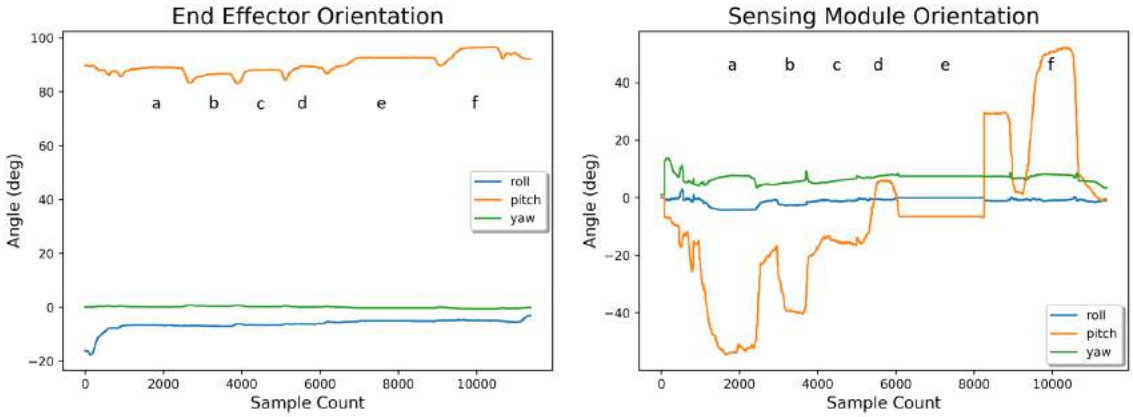


Figure 5.52: Vitamin C Bottle Orientation Graphs

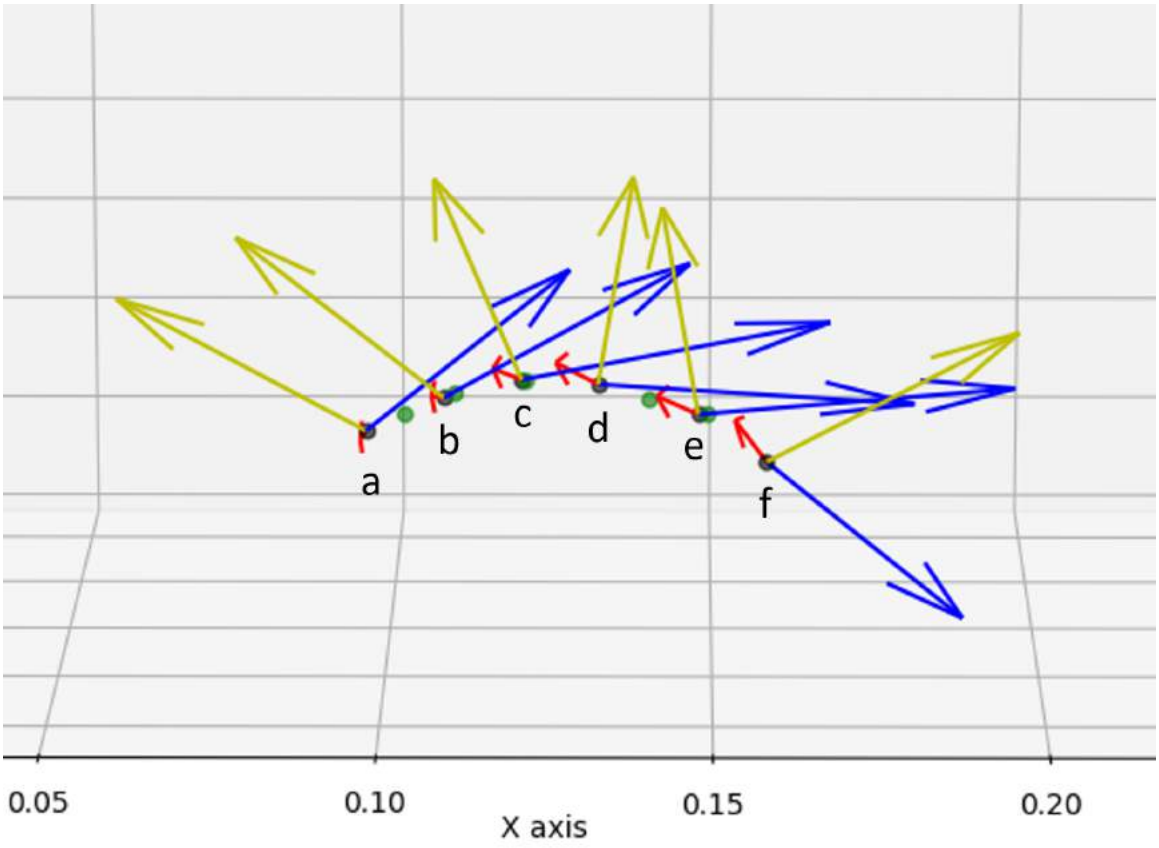
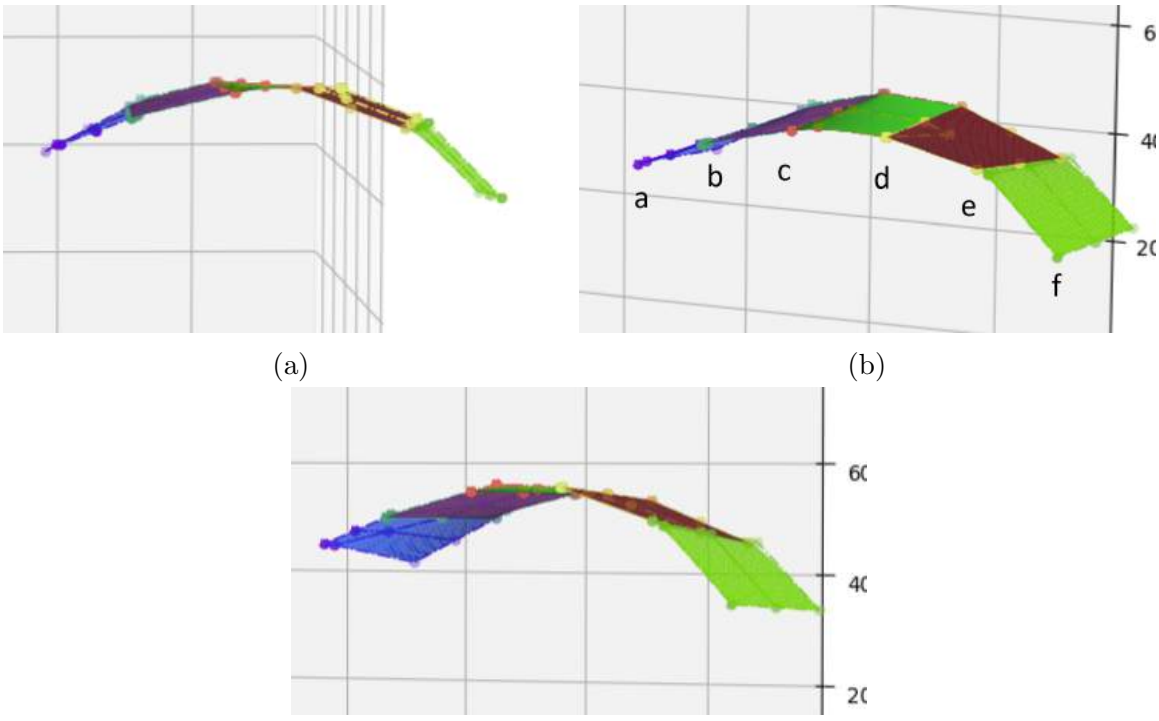


Figure 5.53: Vitamin C Bottle Markers



(a) (b)

(c)

Figure 5.54: Vitamin C Bottle Estimation

2 lines of points with 6 points for each line were collected during the exploration of the vitamin C bottle. The full object is shown in Figure 5.46 and the starting point of one such line is shown in 5.46 (a) and the ending point in 5.46 (b). The contact points of the line are shown in Figure 5.47. The corresponding data obtained from the barometer, accelerometer and gyroscope in shown in Figure 5.48, Figure 5.49 and Figure 5.50 respectively. Figure 5.51 shows the movement of the end-effector and Figure 5.52 shows the orientation changes of the end-effector and the sensing module. As the points are collected from one side to the other, it can be seen that the end-effector pitch remains fairly constant while the sensing module pitch shows minimums and then maximums in the graph. Figure 5.53 shows the pose data while Figure 5.54 shows the estimated results in 3 views. The vitamin C bottle has a round surface which has been estimated accurately. The estimation being shorter on one side reflects that the starting point on the left side was a bit higher than the ending point on the right when collecting points on the object.

5.2.2 Peanut Massage Ball

(a) (b)

Figure 5.55: Peanut Massage Ball

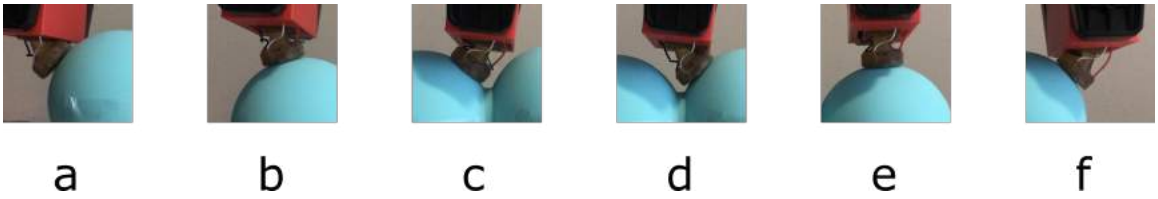


Figure 5.56: Peanut Massage Ball Contact Points

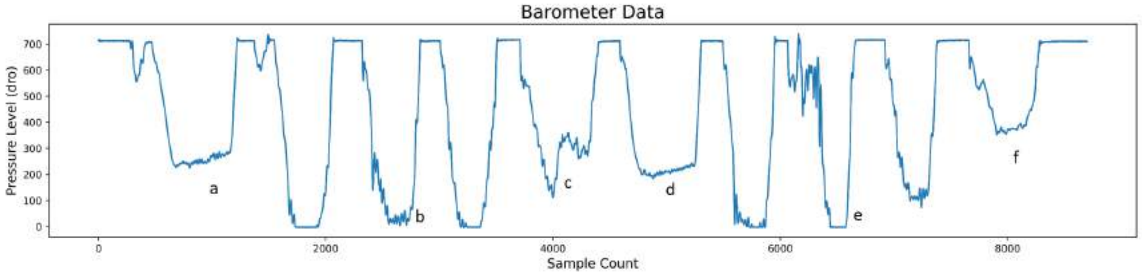


Figure 5.57: Peanut Massage Ball Barometer Graph

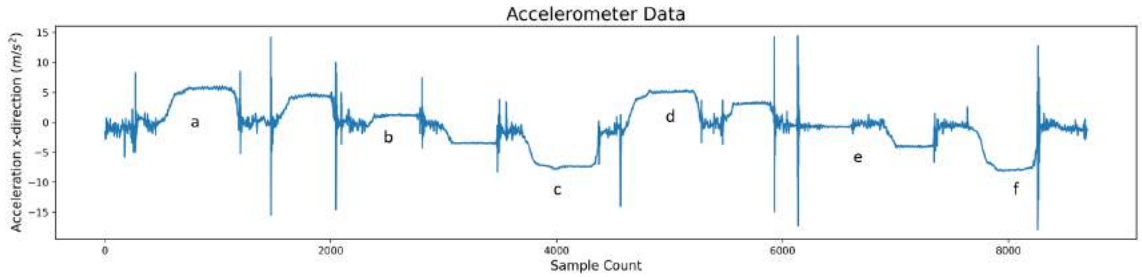


Figure 5.58: Peanut Massage Ball Accelerometer Graph

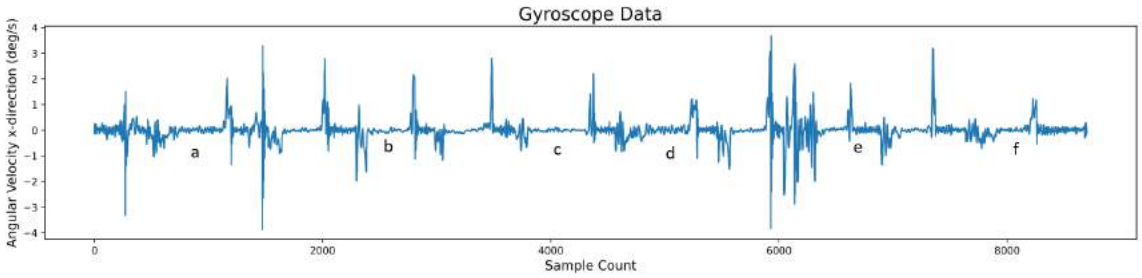


Figure 5.59: Peanut Massage Ball Gyroscope Graph

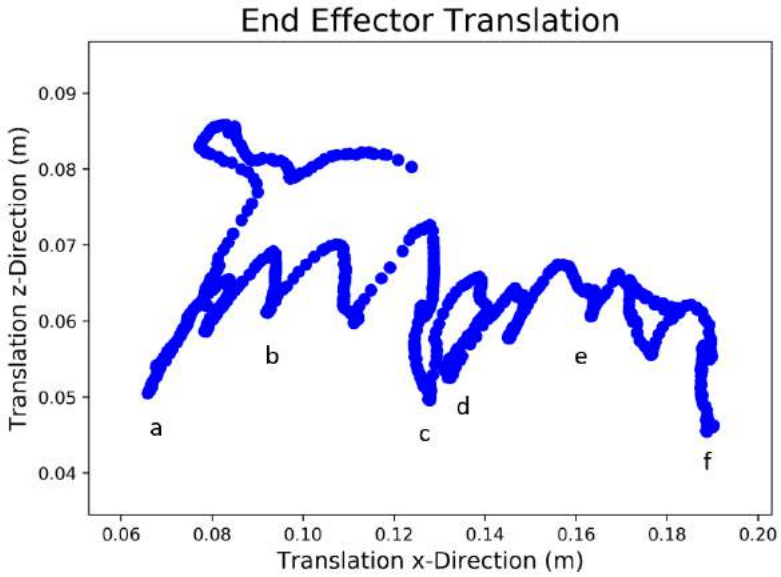


Figure 5.60: Peanut Massage Ball Translation Graph

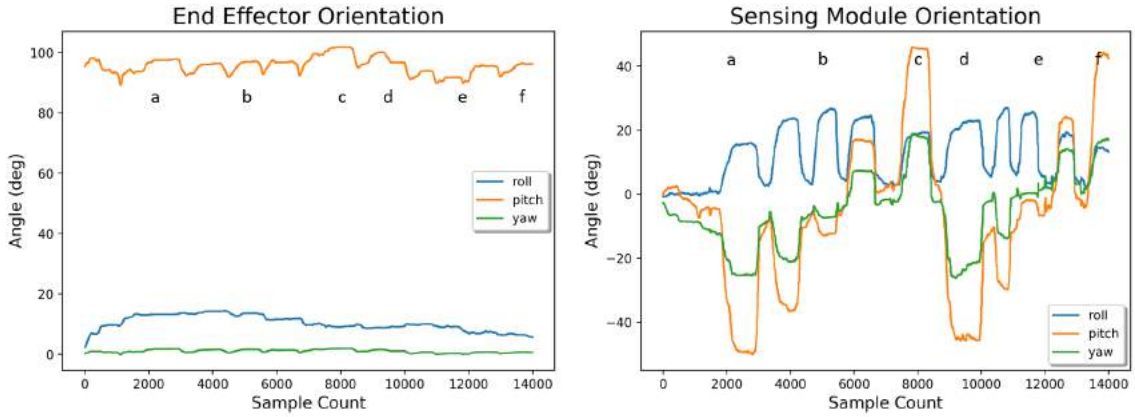


Figure 5.61: Peanut Massage Ball Orientation Graphs

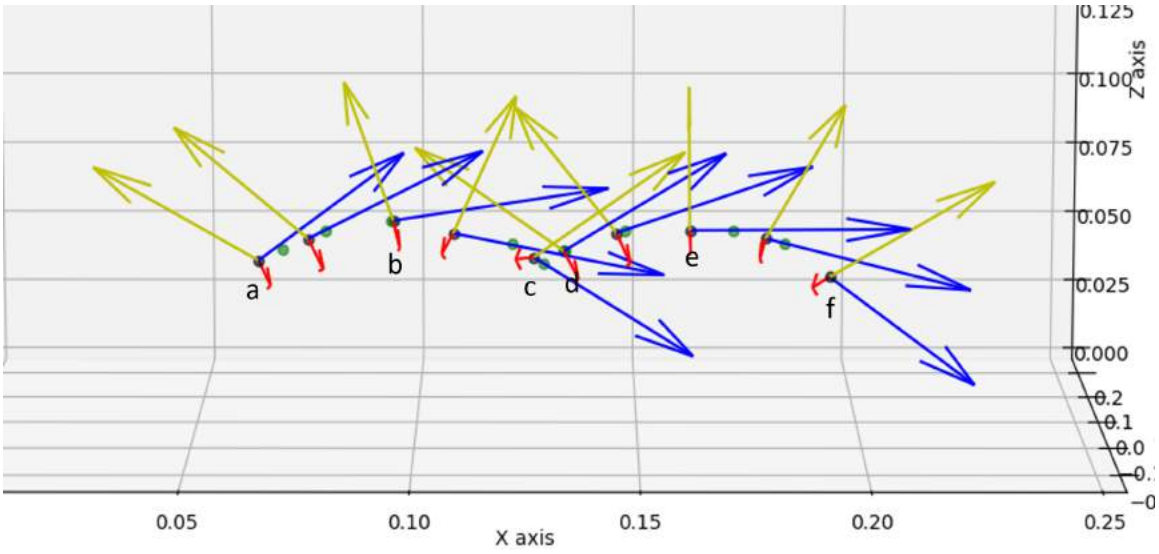
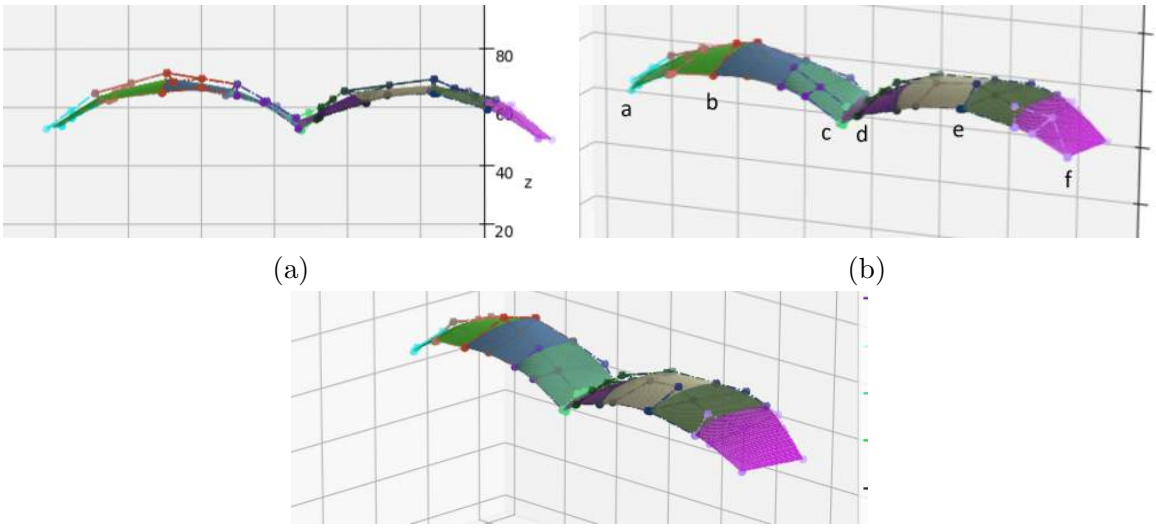


Figure 5.62: Peanut Massage Ball Markers



(a) (b)

(c)

Figure 5.63: Peanut Massage Ball Estimation

Similar to the vitamin C bottle, 2 lines of points with 6 points for each line were collected during the exploration of the peanut massage ball. The contact points of one such line are shown in Figure 5.56. The corresponding data obtained from the barometer, accelerometer and gyroscope in shown in Figure 5.57, Figure 5.58 and Figure 5.59 respectively. Figure 5.60 shows the movement of the end-effector and Figure 5.61 shows the orientation changes of the end-effector and the sensing module.

Figure 5.62 shows the pose data while Figure 5.63 shows the estimated results in 3 views. The peanut has an interesting shape with a downward peak in the middle. As seen in Figure 5.56 points are collected in a way that only the sides of the downward peak in the middle were contacted. But as seen in the estimation, the downward peak is pretty well estimated which is due to the control point calculated between the contact points. The estimation also doesn't go as deep as the actual object itself which is the desired outcome. If the slope is too deep, the sensor may get damaged on re-exploration.

5.2.3 Hair Trimmer

(a) (b)

Figure 5.64: Hair Trimmer

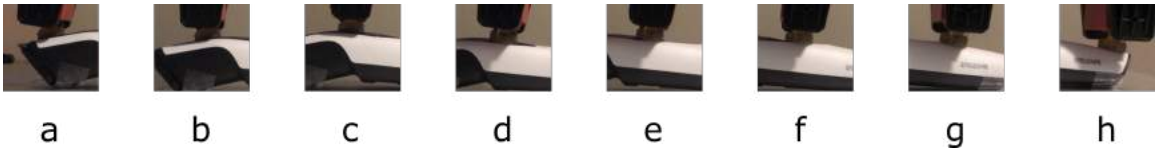


Figure 5.65: Hair Trimmer Contact Points

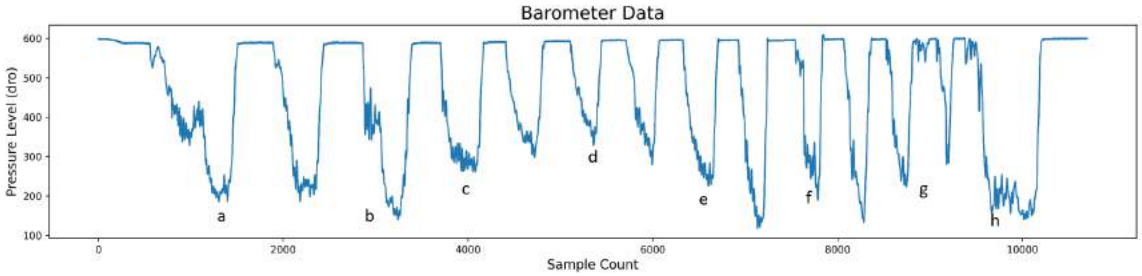


Figure 5.66: Hair Trimmer Barometer Graph

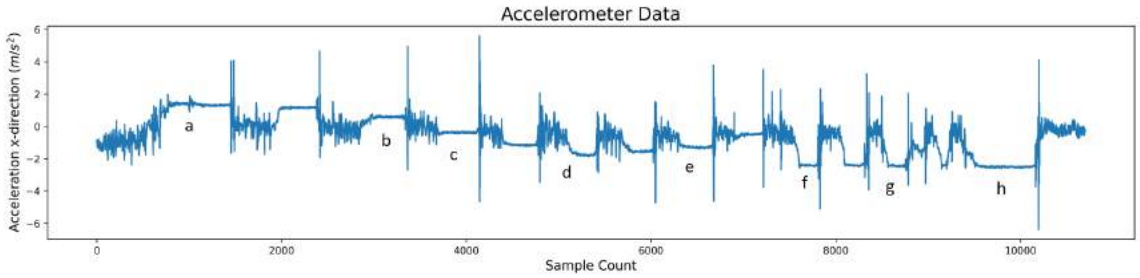


Figure 5.67: Hair Trimmer Accelerometer Graph

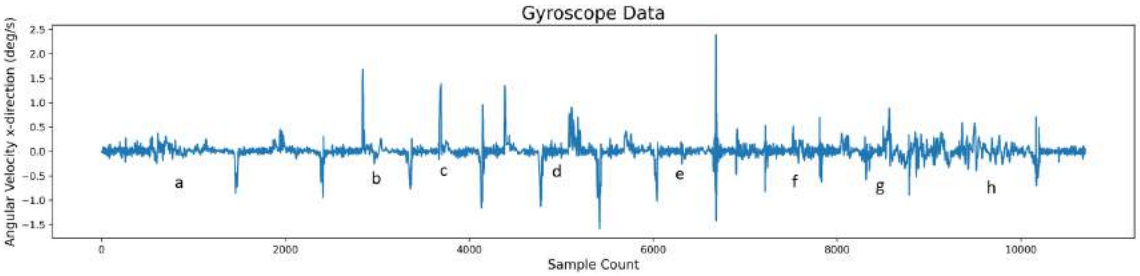


Figure 5.68: Hair Trimmer Gyroscope Graph

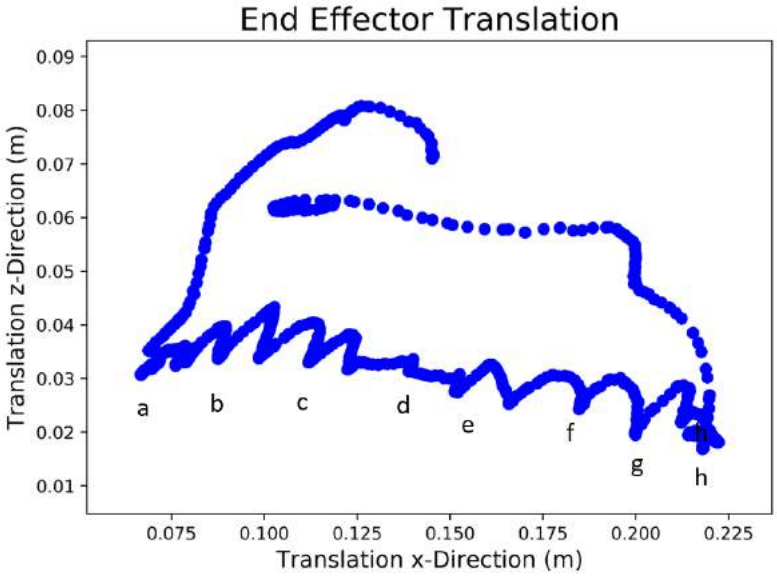


Figure 5.69: Hair Trimmer Translation Graph

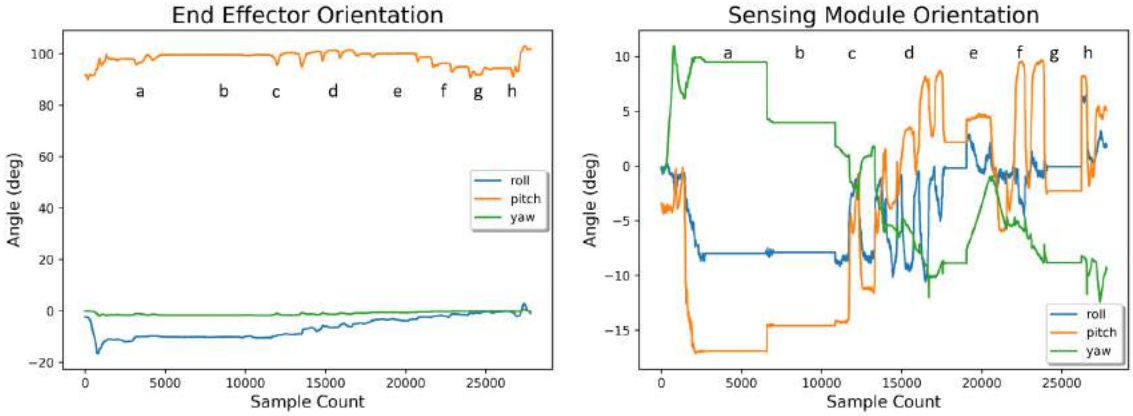


Figure 5.70: Hair Trimmer Orientation Graphs

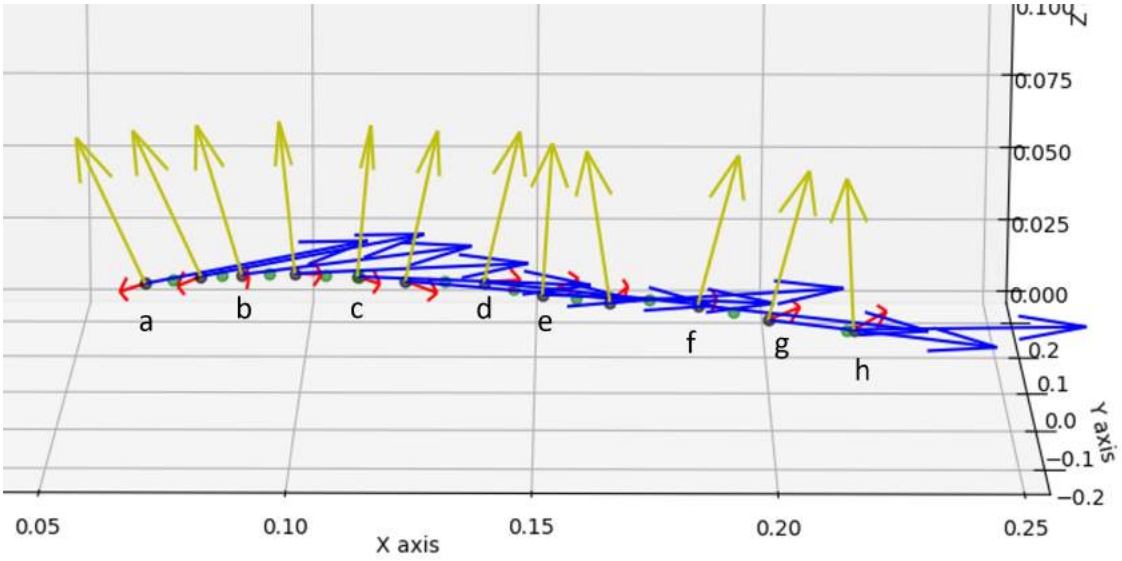
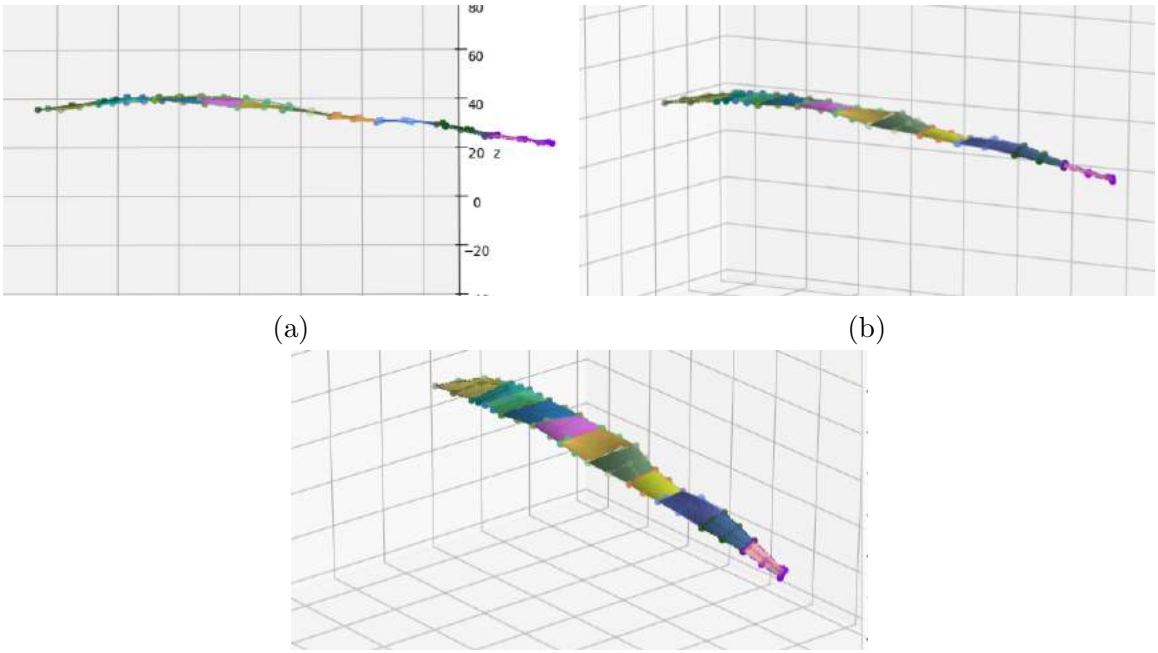


Figure 5.71: Hair Trimmer Markers



(a) (b)

(c)

Figure 5.72: Hair Trimmer Estimation

2 lines of points were collected with 12 points for each line during the exploration of the hair trimmer. Figure 5.65 shows 8 of the 12 points collected. The corresponding graphs with the annotated contact points are shown in Figure 5.66, Figure 5.67, Figure 5.68, Figure 5.69 and Figure 5.70. In Figure 5.70, it can be seen that the end-effector pitch remains relatively constant while the upward slope and the downward slope which has a slight dip in the middle which is not noticeable by vision is represented by the pitch of the sensing module. The longer minimums at contact points A and B allude to the fact that the sensor was held at those points longer. Figure 5.71 shows the collected pose of the contact points and the calculated control points while Figure 5.72 shows the estimated surface in 3 views. It can be seen that at the end of the downward slope the surface narrows which is difficult to see using vision because of the visual features of the object.

5.2.4 Cream Bottle

(a) (b)

Figure 5.73: Cream Bottle

Figure 5.74: Cream Bottle Contact Points

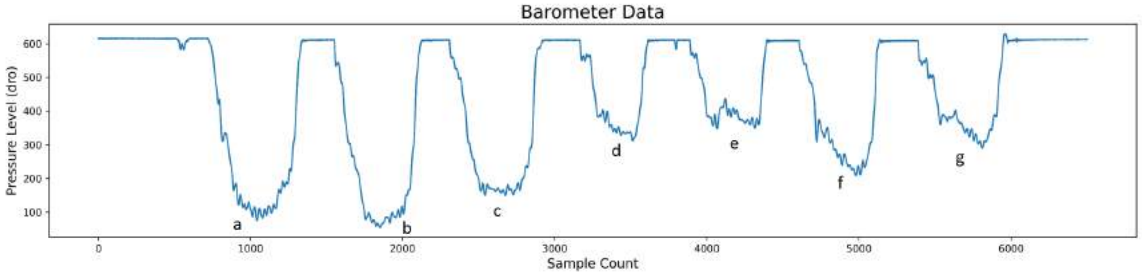


Figure 5.75: Cream Bottle Barometer Graph

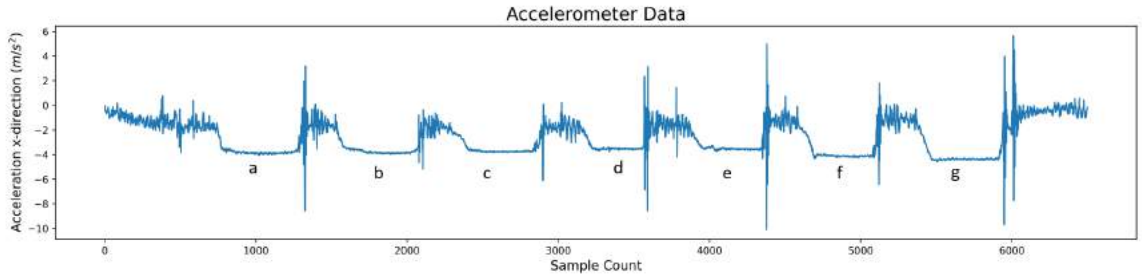


Figure 5.76: Cream Bottle Accelerometer Graph

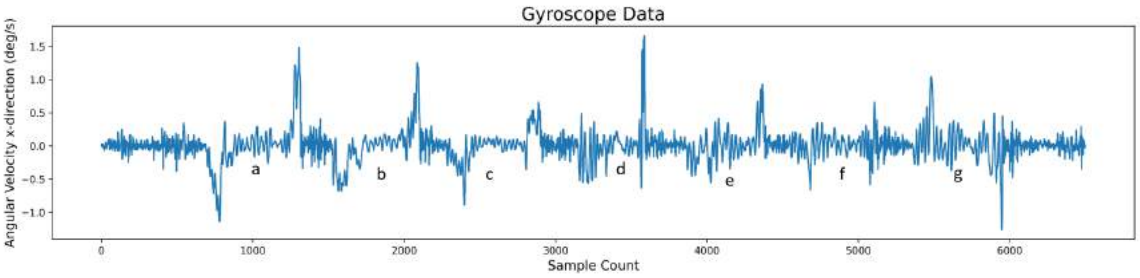


Figure 5.77: Cream Bottle Gyroscope Graph

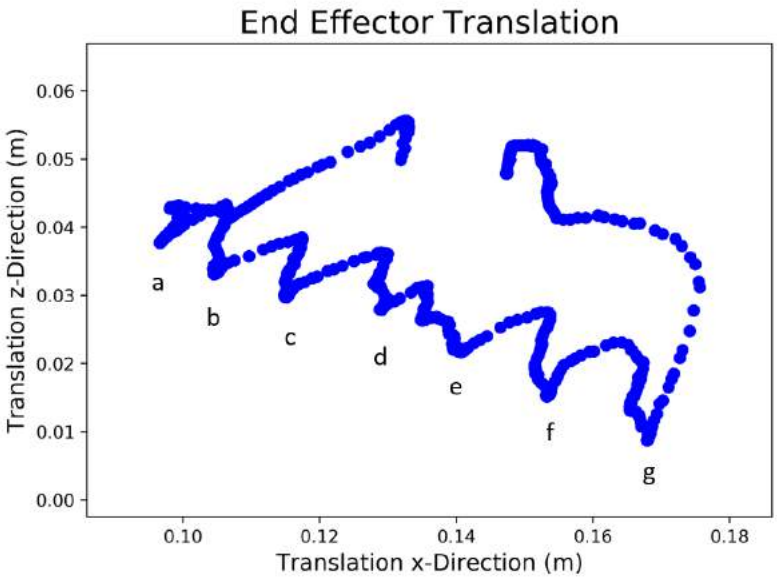


Figure 5.78: Cream Bottle Translation Graph

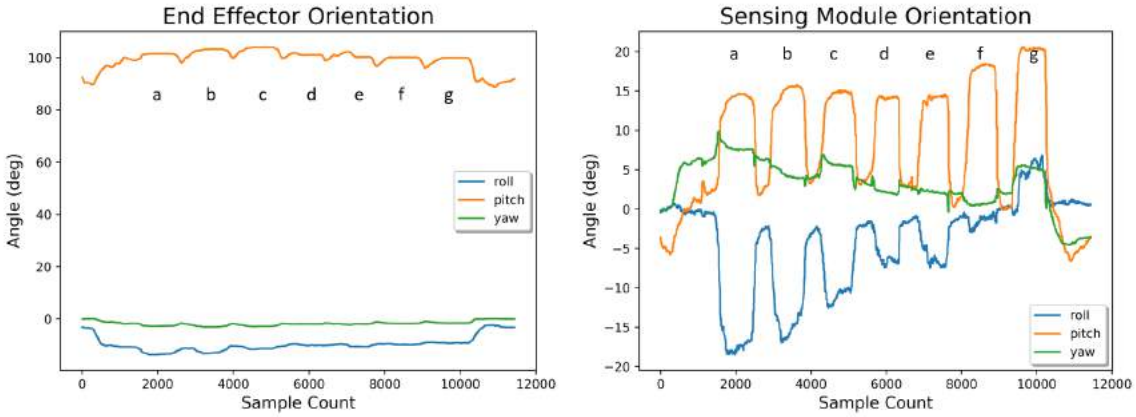


Figure 5.79: Cream Bottle Orientation Graphs

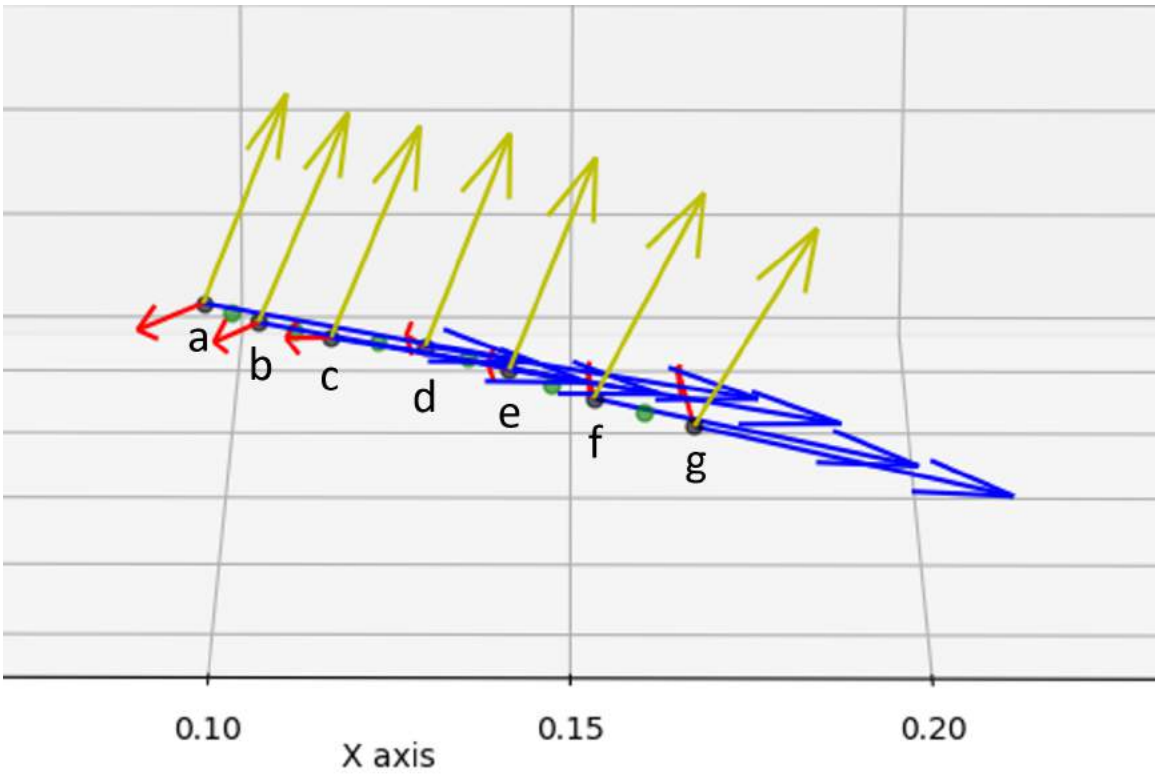
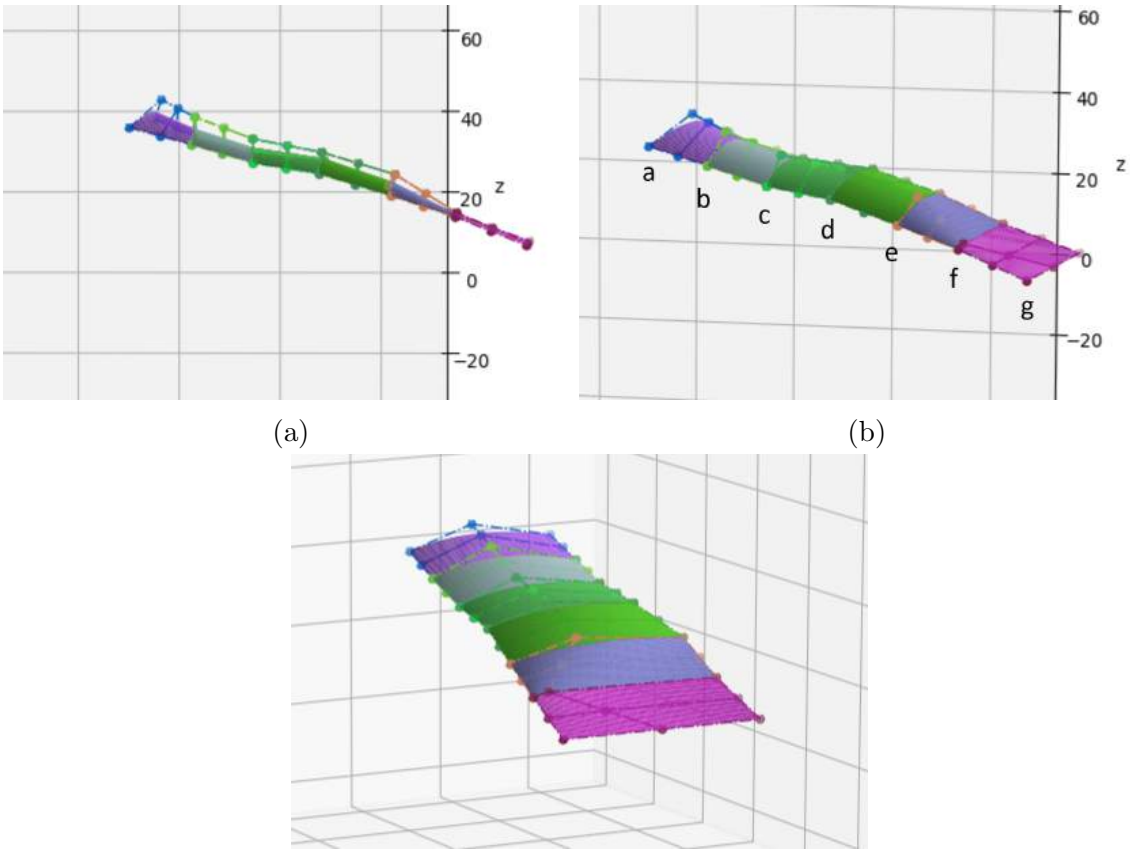


Figure 5.80: Cream Bottle Markers



(a) (b)

(c)

Figure 5.81: Cream Bottle Estimation

2 lines of points with 7 points for each line were collected during the exploration of the cream bottle. The corresponding graphs with the annotated contact points are shown in Figure 5.75, Figure 5.76, Figure 5.77, Figure 5.78 and Figure 5.79. Figure 5.80 shows the collected pose of the contact points and the calculated control points while Figure 5.81 shows the estimated surface in 3 views. The cream bottle looks to be a pretty straightforward downward slope that was clearly estimated. In fact, the cream bottle is soft and a small amount of pressure would deform its surface. Using this soft sensor, we were able to estimate its surface without the bottle deforming.

5.2.5 Transparent Bottle

(a) (b)

Figure 5.82: Transparent Bottle

Figure 5.83: Transparent Bottle Contact Points

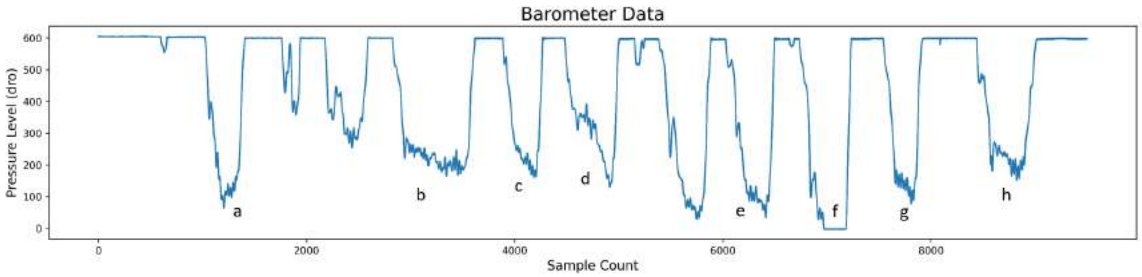


Figure 5.84: Transparent Bottle Barometer Graph

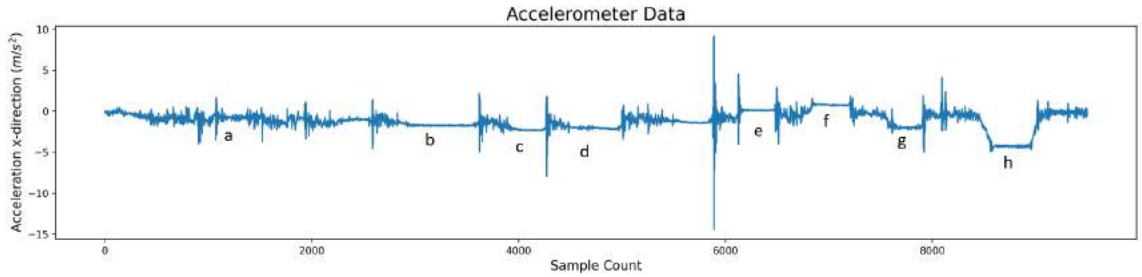


Figure 5.85: Transparent Bottle Accelerometer Graph

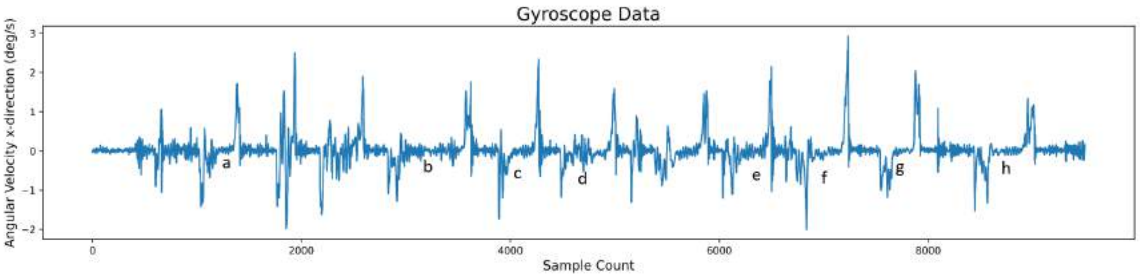


Figure 5.86: Transparent Bottle Gyroscope Graph

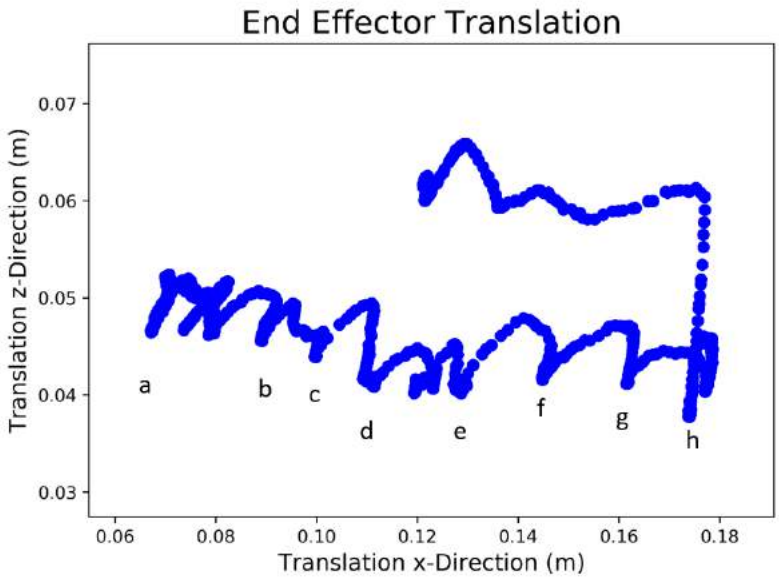


Figure 5.87: Transparent Bottle Translation Graph

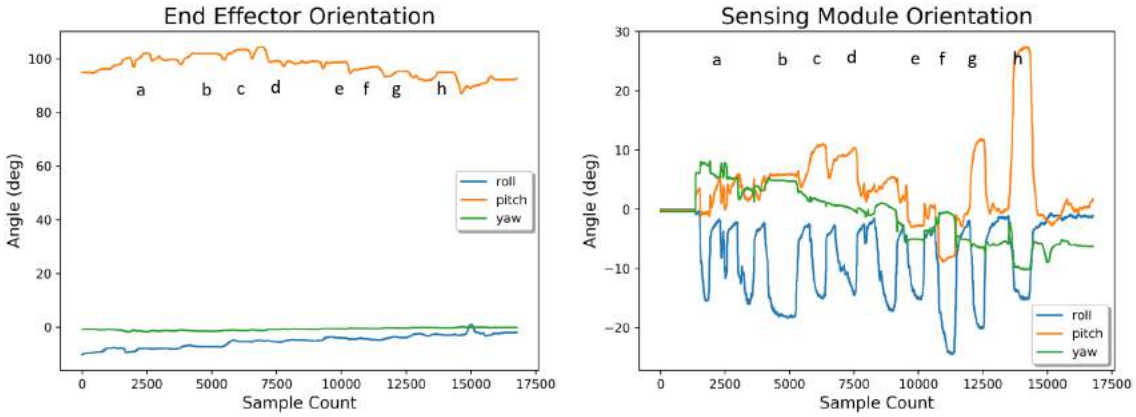


Figure 5.88: Transparent Bottle Orientation Graphs

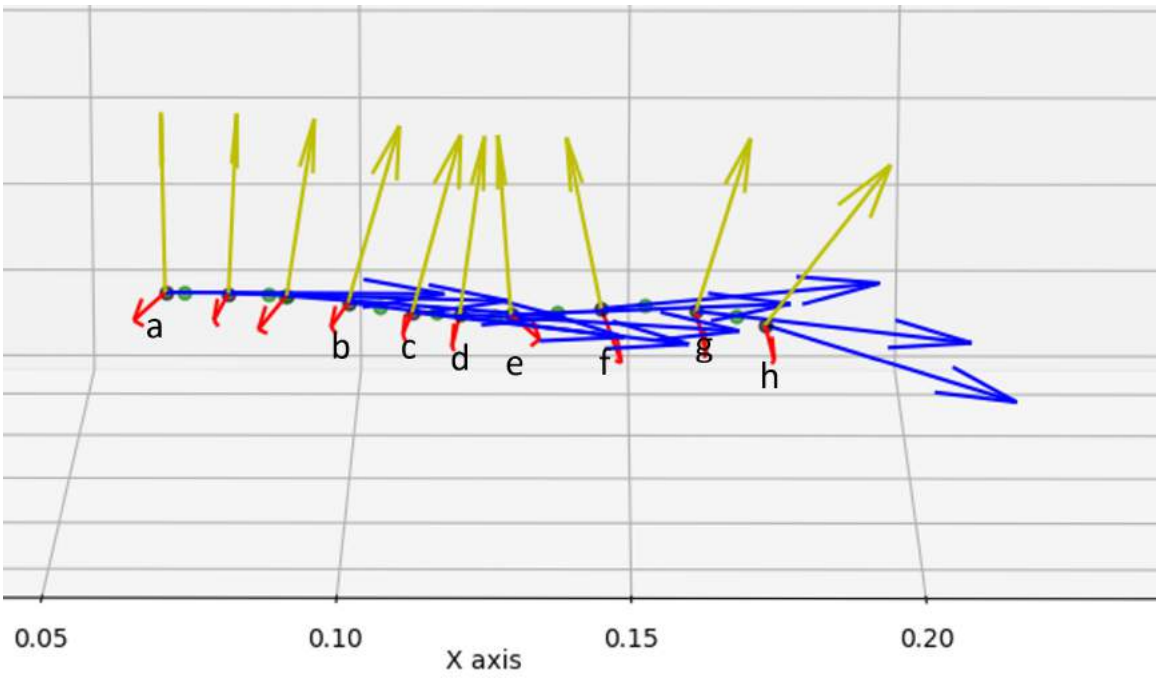
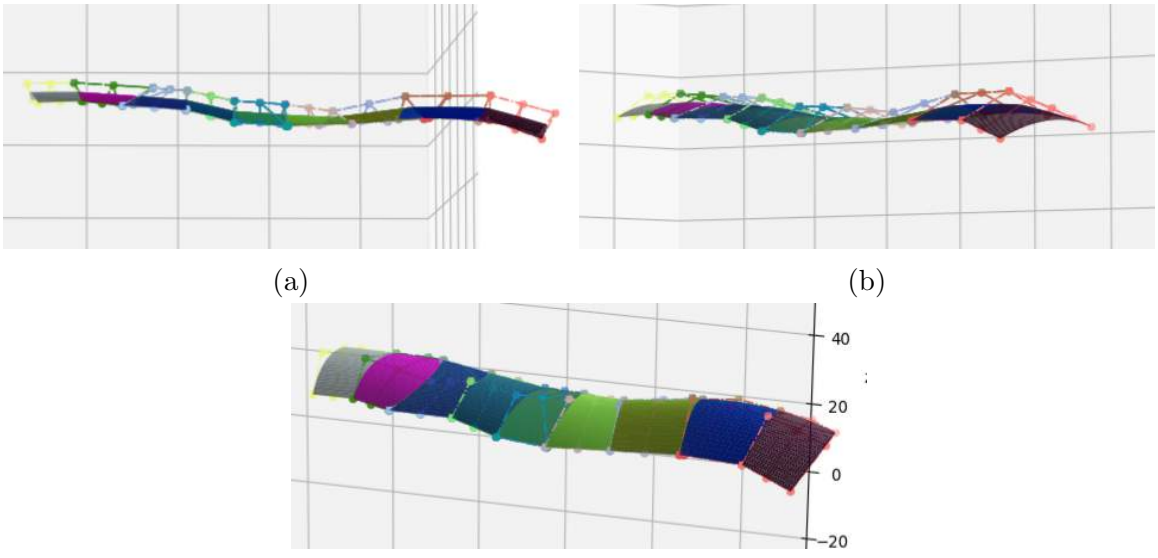


Figure 5.89: Transparent Bottle Markers



(a) (b)

(c)

Figure 5.90: Transparent Bottle Estimation

2 lines of points were collected with 10 points for each line during the exploration of the hair trimmer. Figure 5.83 shows 8 of the 10 points collected. The corresponding graphs with the annotated contact points are shown in Figure 5.84, Figure 5.85,

Figure 5.86, Figure 5.87 and Figure 5.88. The biggest deformation of the sensing module is at point H which is represented in Figure 5.88 by the pitch of the sensing module. Figure 5.89 shows the collected pose of the contact points and the calculated control points while Figure 5.90 shows the estimated surface in 3 views. The bottle is transparent and it is difficult to see the curves through vision and it would be close to impossible to reconstruct using only vision. Using our approach, the curves can be clearly seen in the estimation.

5.2.6 Screw

(a) (b)

Figure 5.91: Screw

Figure 5.92: Screw Contact Points

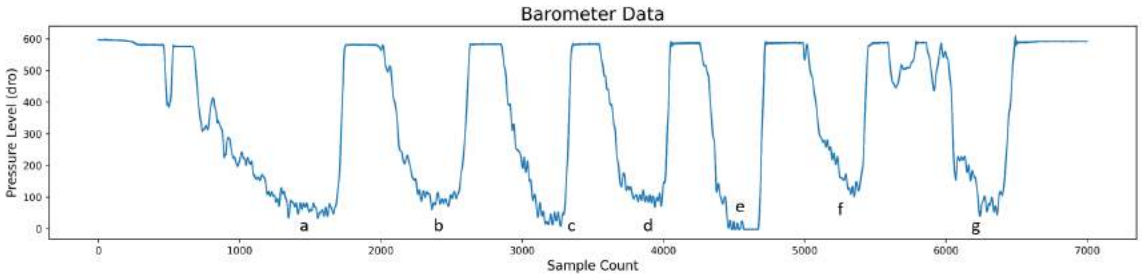


Figure 5.93: Screw Barometer Graph

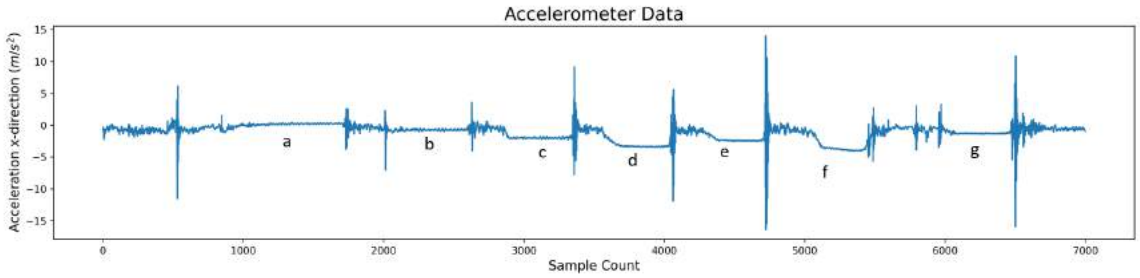


Figure 5.94: Screw Accelerometer Graph

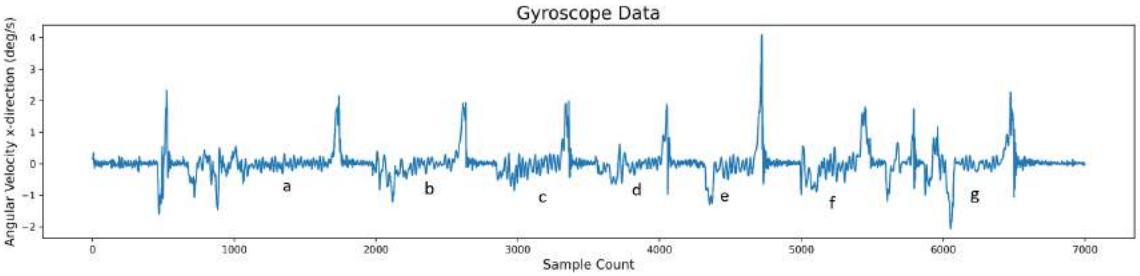


Figure 5.95: Screw Gyroscope Graph

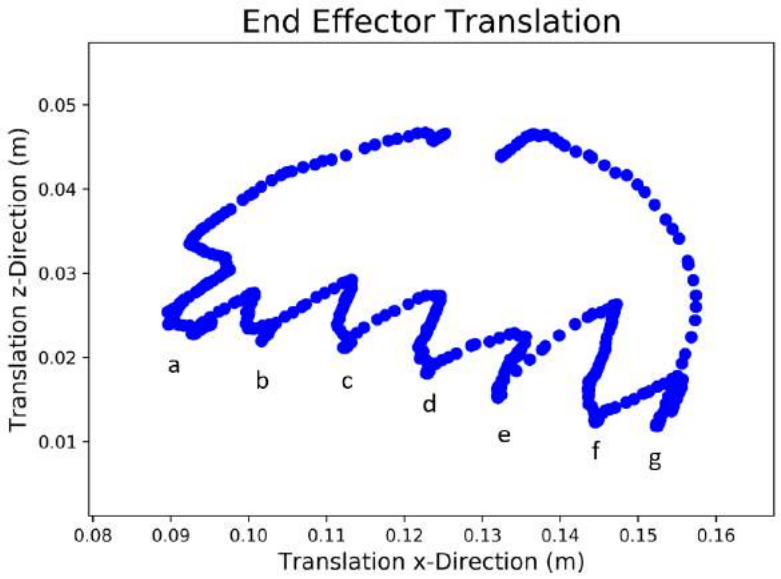


Figure 5.96: Screw Translation Graph

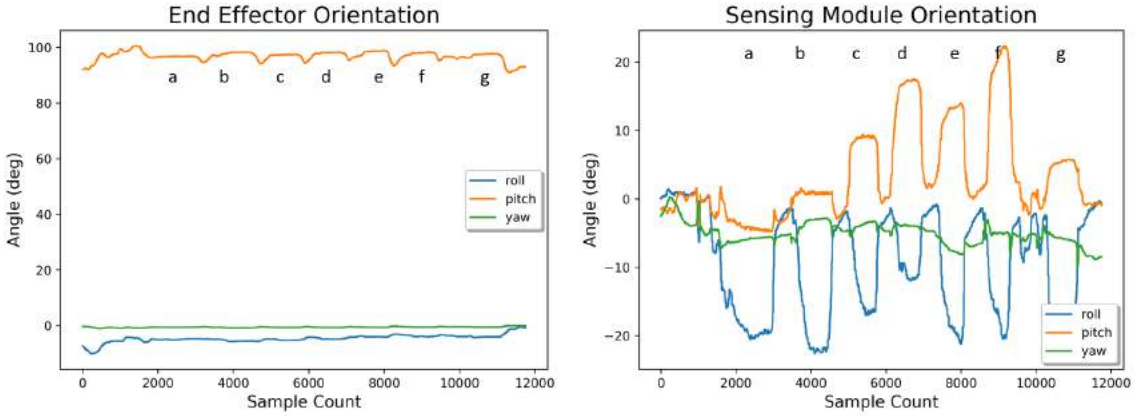


Figure 5.97: Screw Orientation Graphs

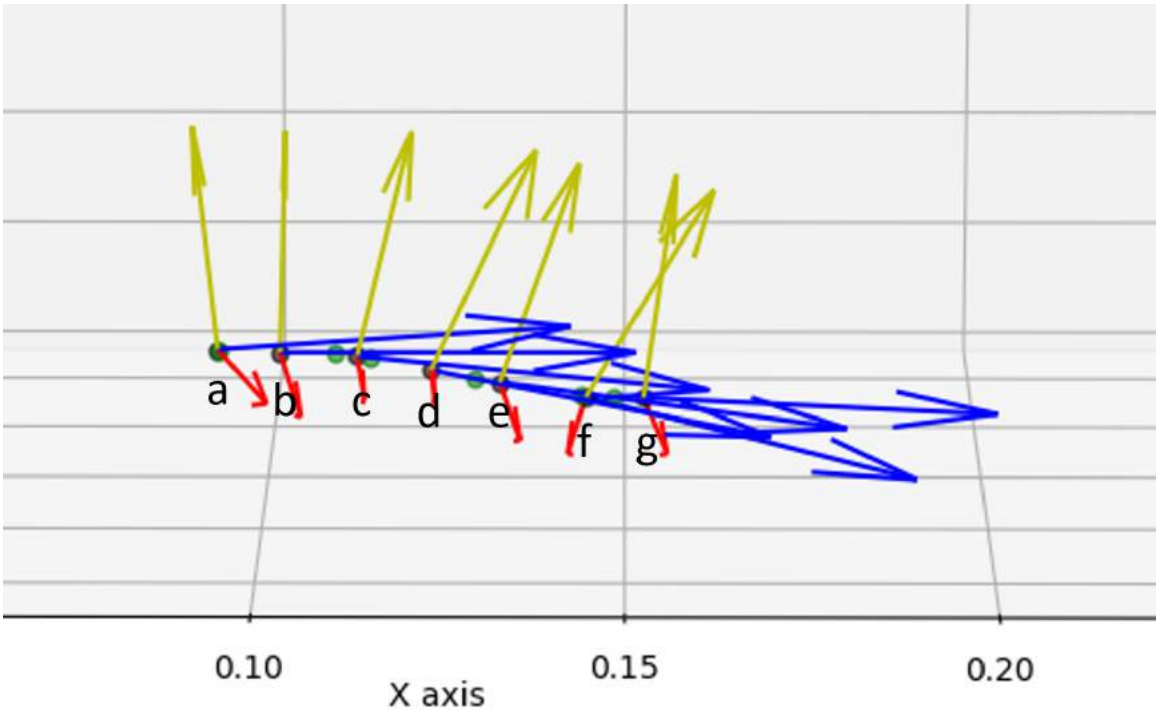
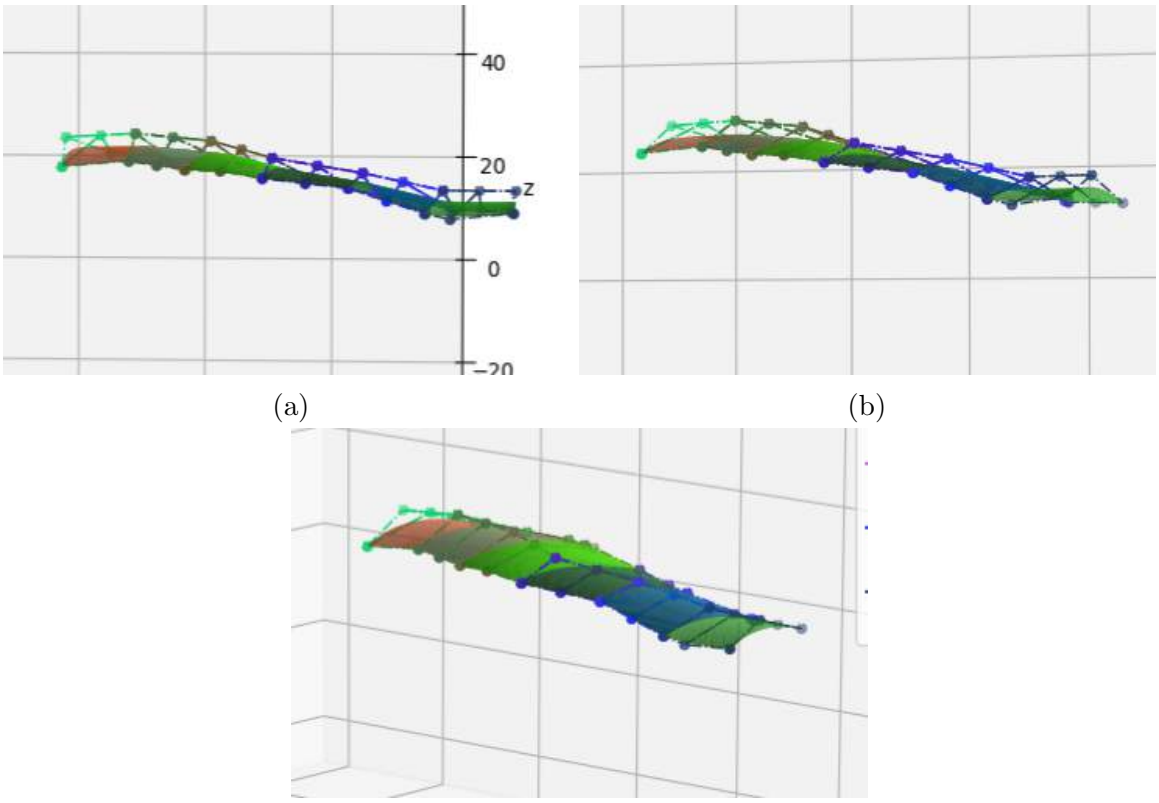


Figure 5.98: Screw Markers



(a) (b)

(c)

Figure 5.99: Screw Estimation

2 lines of points were collected with 7 points for each line during the exploration of the screw. Figure 5.92 shows one such line. The corresponding graphs with the annotated contact points are shown in Figure 5.93, Figure 5.94, Figure 5.95, Figure 5.96 and Figure 5.97. Figure 5.98 shows the collected pose of the contact points and the calculated control points while Figure 5.99 shows the estimated surface in 3 views. Although only 2 lines of points were collected on either side of the screw, the roundness of the screw has been clearly estimated. This is due to the control points calculated between the contact points which shows the importance of our control point calculation method.

5.2.7 Earphone Case

(a) (b)

Figure 5.100: Earphone Case

Figure 5.101: Earphone Case Contact Points

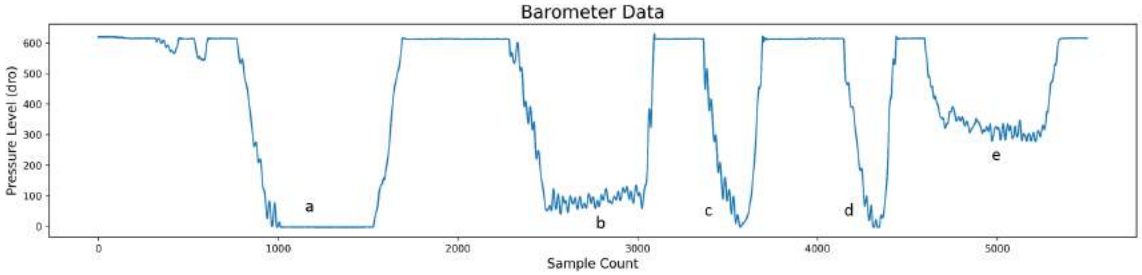


Figure 5.102: Earphone Case Barometer Graph

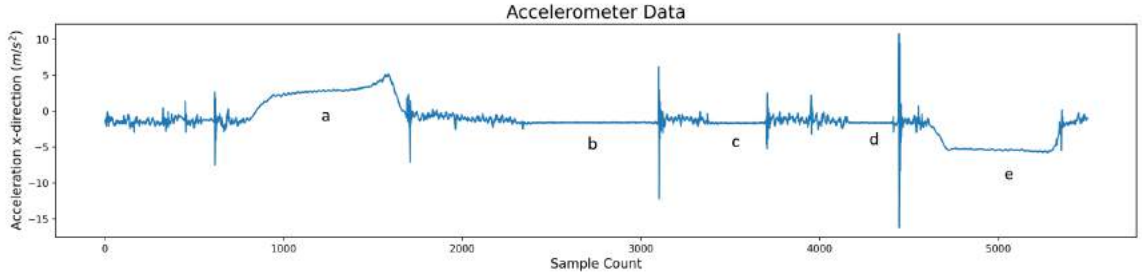


Figure 5.103: Earphone Case Accelerometer Graph

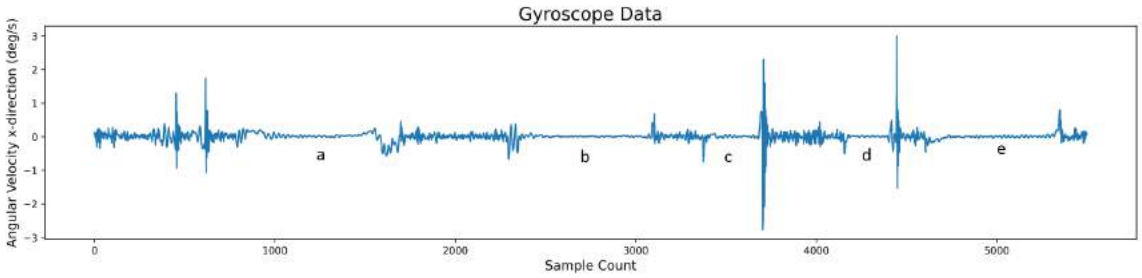


Figure 5.104: Earphone Case Gyroscope Graph

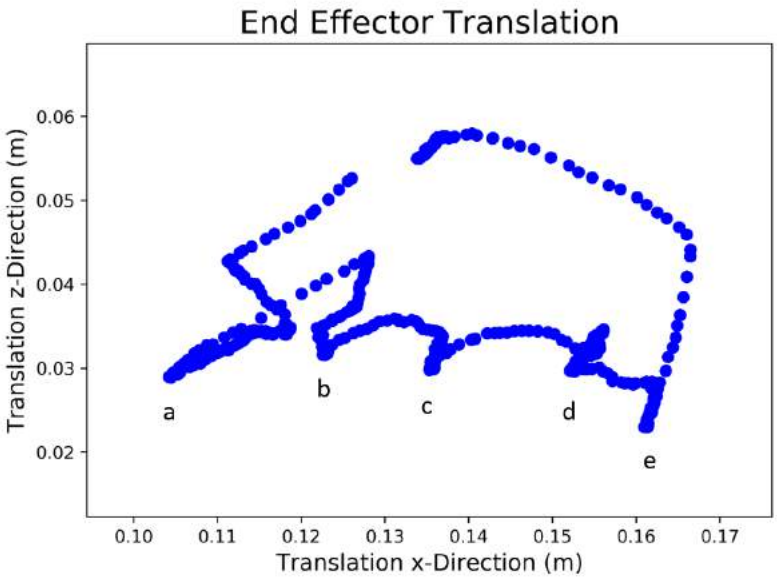


Figure 5.105: Earphone Case Translation Graph

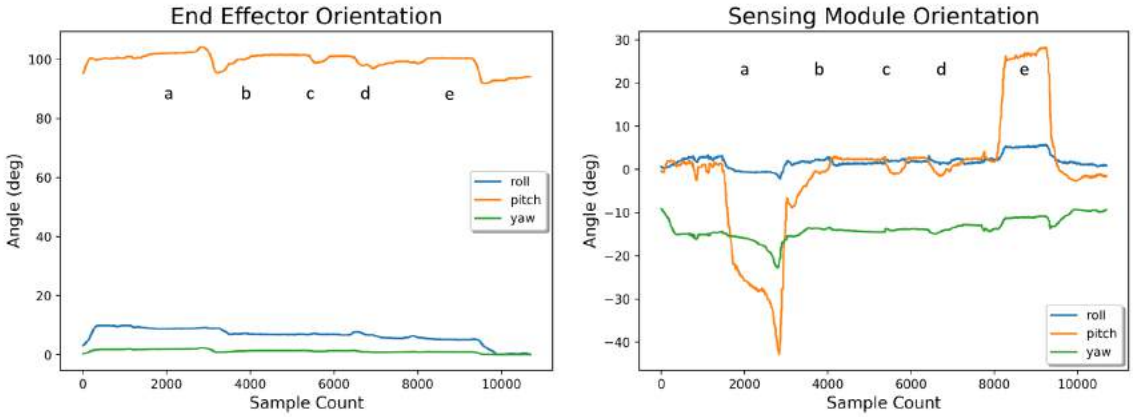


Figure 5.106: Earphone Case Orientation Graphs

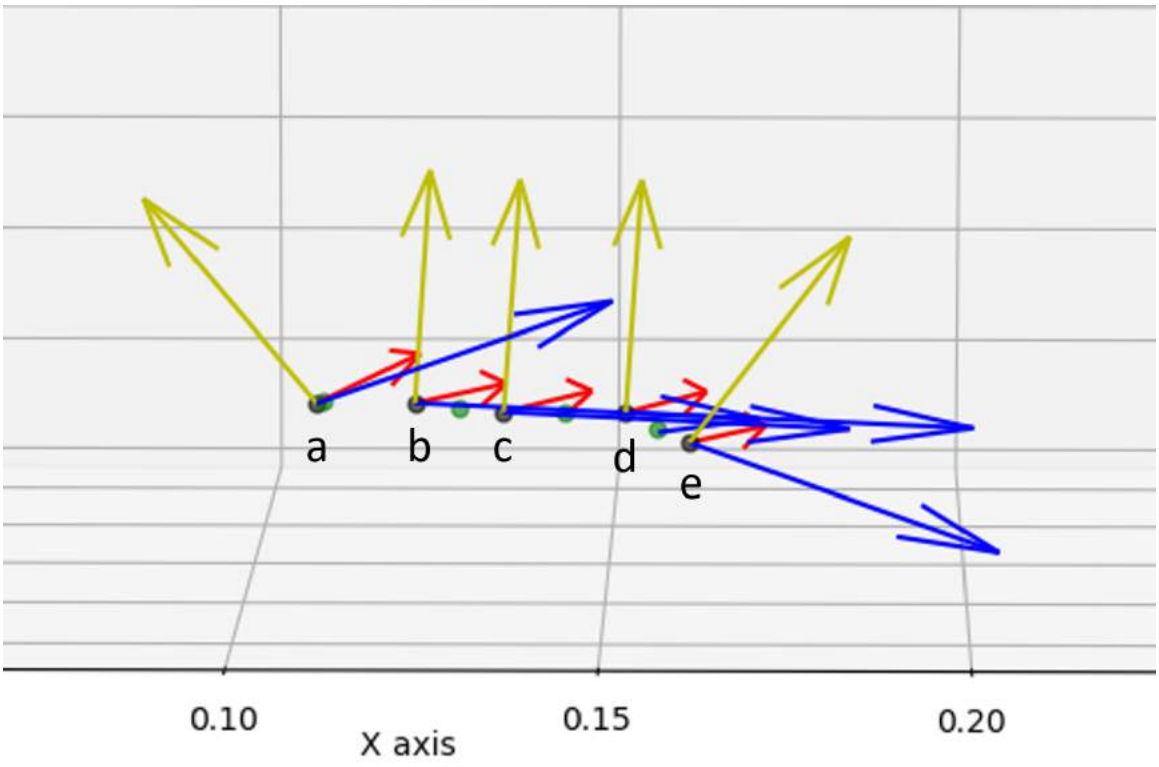
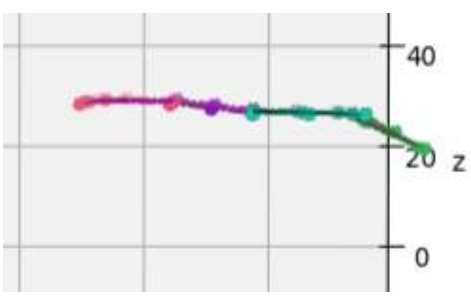
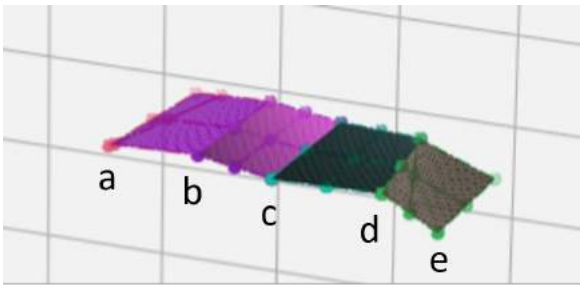


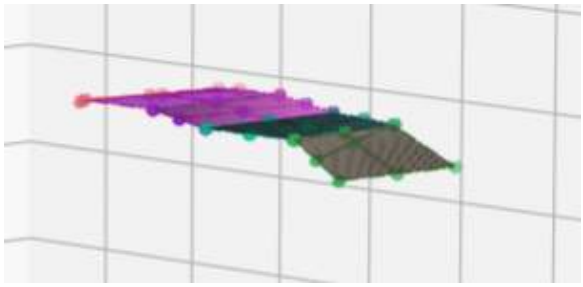
Figure 5.107: Earphone Case Markers



(a)



(b)



(c)

Figure 5.108: Earphone Case Estimation

For the earphone case, 2 lines of points were collected with 5 points on each line during the exploration. One such line is shown in Figure 5.101. The corresponding graphs with the annotated contact points are shown in Figure 5.102, Figure 5.103, Figure 5.104, Figure 5.105 and Figure 5.106. In Figure 5.106, it can be seen that the end-effector pitch stays fairly constant while the pitch of the sensing module dips down at the start when the deformation is upwards and stays at 0 in the middle and peaks at the end where the deformation is downwards. Figure 5.107 shows the collected pose of the contact points and the calculated control points while Figure 5.108 shows the estimated surface in 3 views. The earphone case has a straight downward slope with two curved edges at either end. The two curves are very small compared to the sensor and as such, the estimation is not as detailed at the edges as the actual object which is a limitation of the sensor used. With a smaller sensor, these details could also be captured.

5.2.8 Inhaler

(a) (b)

Figure 5.109: Inhaler

Figure 5.110: Inhaler Contact Points

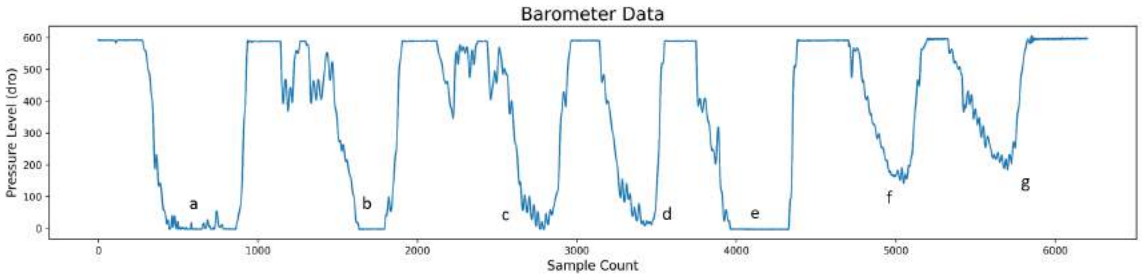


Figure 5.111: Inhaler Barometer Graph

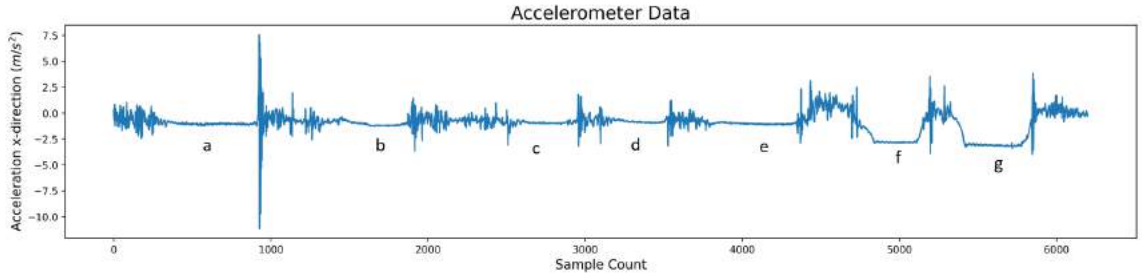


Figure 5.112: Inhaler Accelerometer Graph

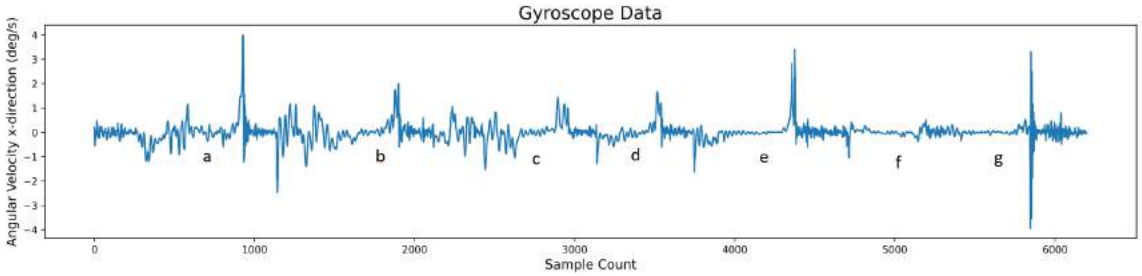


Figure 5.113: Inhaler Gyroscope Graph

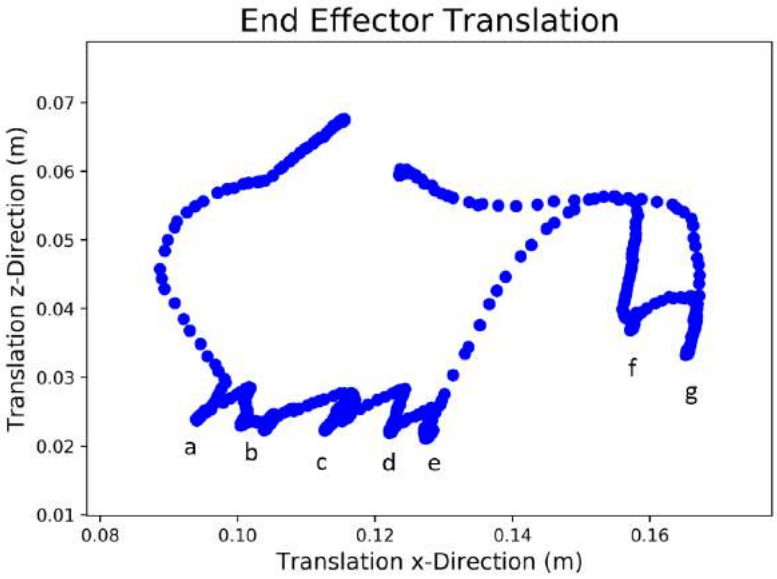


Figure 5.114: Inhaler Translation Graph

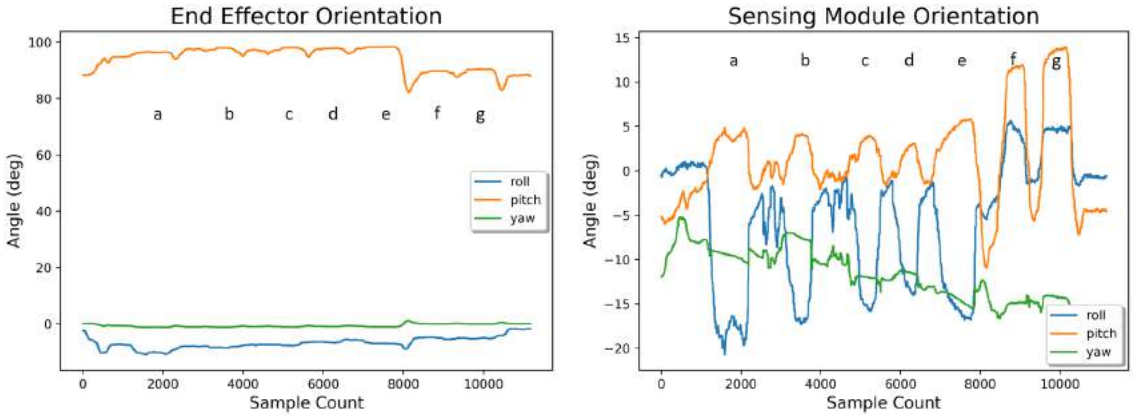


Figure 5.115: Inhaler Orientation Graphs

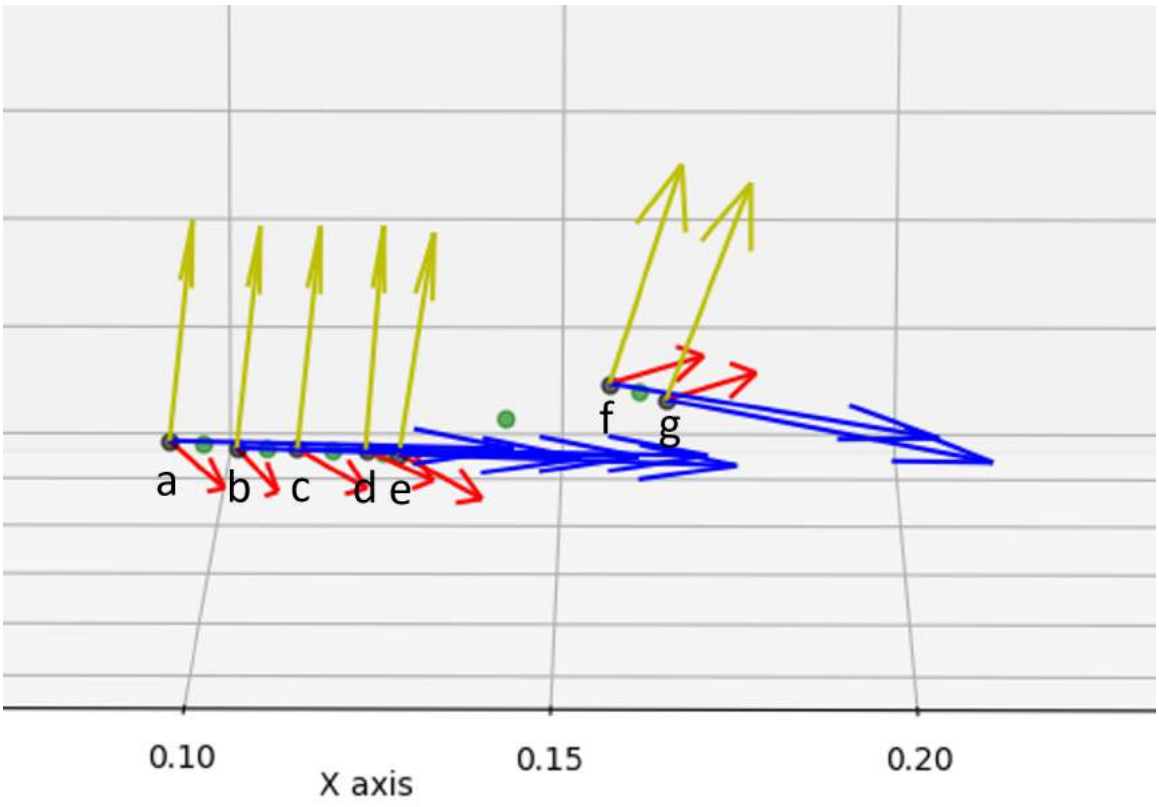
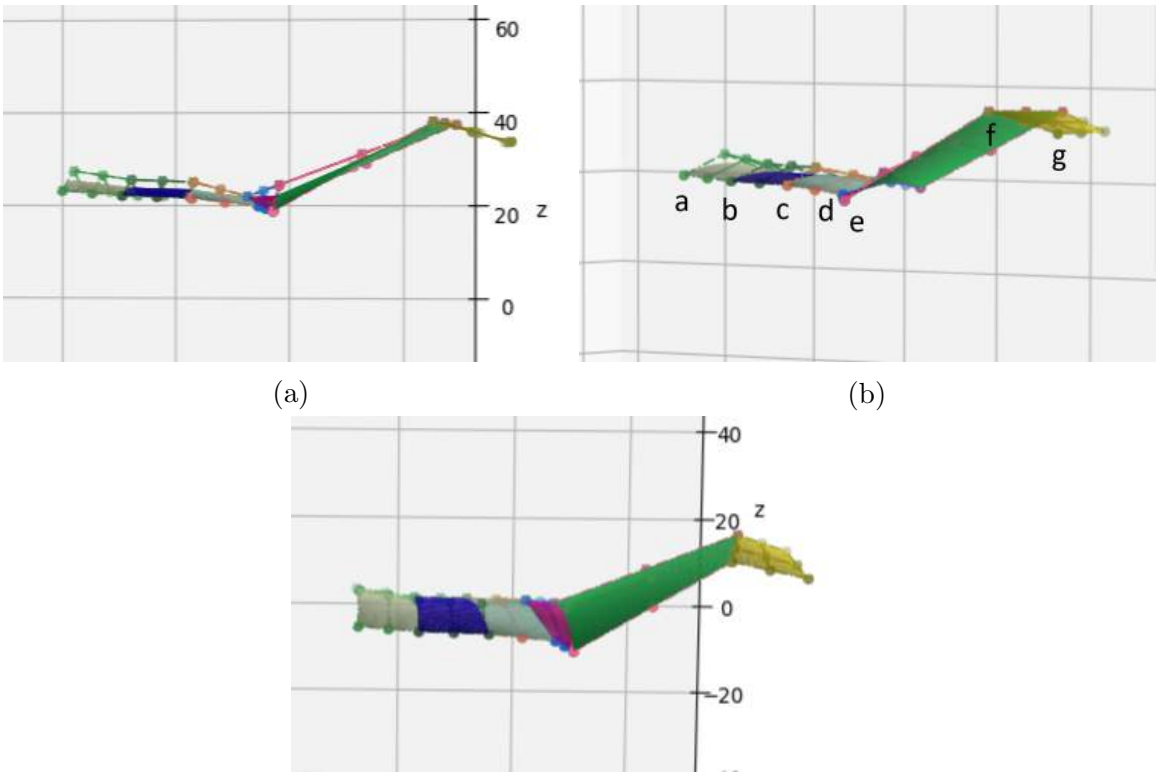


Figure 5.116: Inhaler Markers



(a) (b)

(c)

Figure 5.117: Inhaler Estimation

2 lines of points were collected with 5 points on each line during the exploration of the inhaler. One such line is shown in Figure 5.110. The corresponding graphs with the annotated contact points are shown in Figure 5.111, Figure 5.112, Figure 5.113, Figure 5.114 and Figure 5.115. Figure 5.116 shows the collected pose of the contact points and the calculated control points while Figure 5.117 shows the estimated surface in 3 views. A difference between the estimated surface and the actual surface can be seen in this case. This is due to the presence of a steep step in the inhaler. Due to the end-effector having a constant angle of approach, if there is a step along the angle of the end-effector, it cannot be estimated which could be considered as a limitation of our approach. Although, if the sensor was smaller and was able to collect a point as close to the step as possible, the estimation would be much better than the current estimation.

It is worth mentioning that the tactile data collection process used to estimate these surfaces does not require the robot to have prior information about the surface topology, i.e., no assumptions were made about the end-effector's approach angle and it

was kept constant as the robot probes the surface.

5.3 Texture Exploration

Figure 5.118: Surface 5

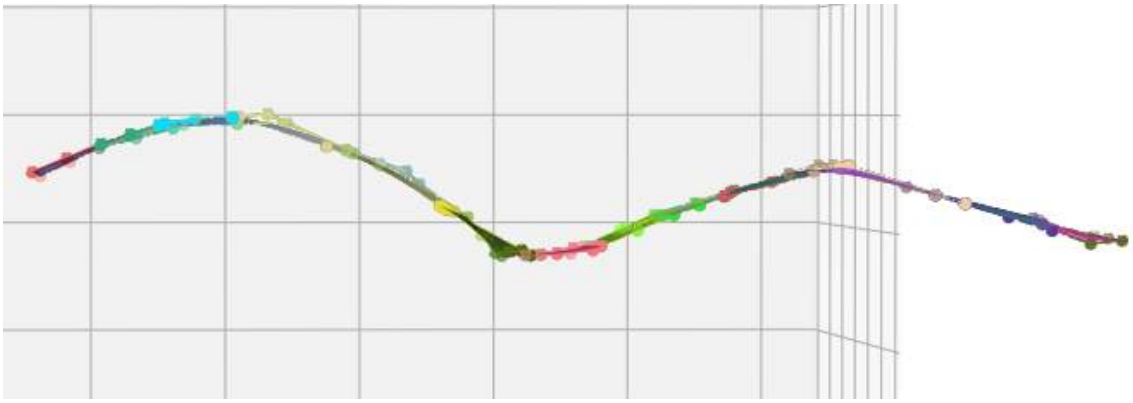


Figure 5.119: Surface 5 Estimation

Surface 5 was used to 3D print 4 textures with the same shape. The shape and it's estimation is shown above in Figure 5.118 Figure 5.119 respectively.

Figure 5.120: Textures

The textures are named and shown in Figure 5.120. Texture A is a 3D texture while Textures B, C, and D are 2D and have varying line spaces between their gratings with texture B having the highest line spaces and texture D having the lowest line spaces.

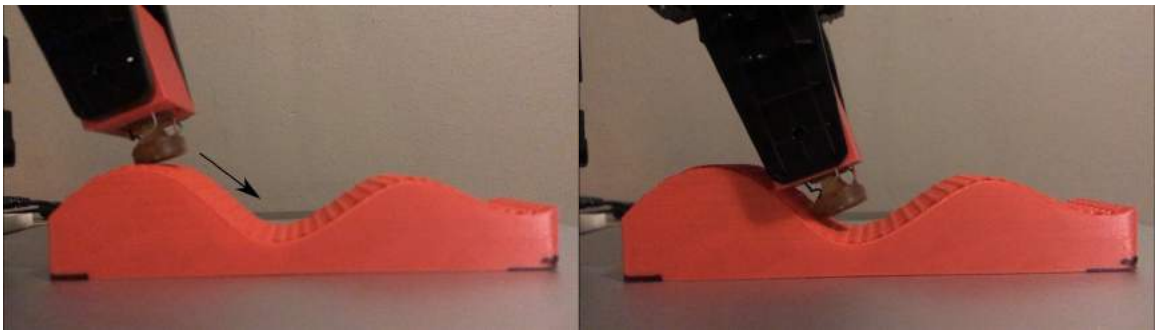


Figure 5.121: Dynamic Tactile Sensing Path

To demonstrate that textures can be explored once a clear estimation of a surface is obtained is possible, we used the robotic manipulator to glide the sensor down the surface as seen in Figure 5.121. The edge of the sensor needs to be in contact with the

surface so that the vibrations that occur from the texture can be obtained. Gliding the sensor over the full surface is a task that cannot be done using our 4DOF robotic manipulator as the end effector cannot perform the necessary rotation to glide over the entire surface with only the edge of the sensor touching the surface. However, it is possible with a robot with a higher DOF. The results obtained from the barometer, accelerometer, gyroscope, and magnetometer are shown below.

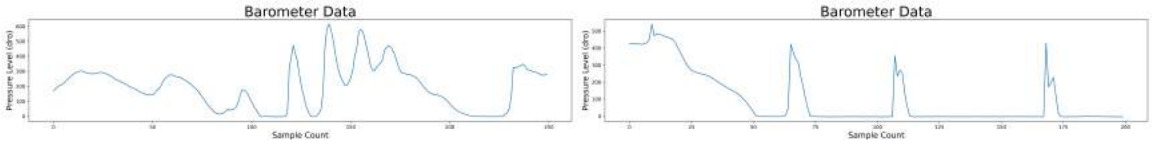


Figure 5.122: Barometer Comparison: Texture A and Texture B

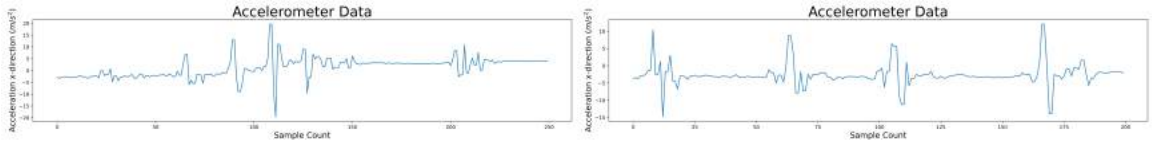


Figure 5.123: Accelerometer Comparison: Texture A and Texture B

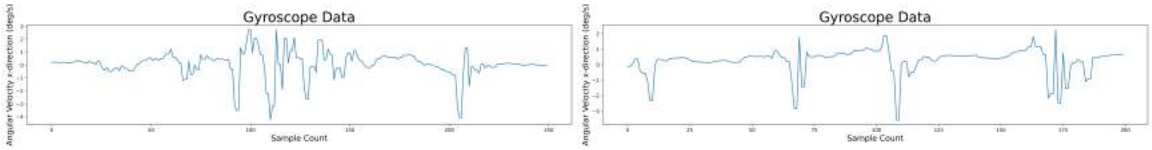


Figure 5.124: Gyroscope Comparison: Texture A and Texture B

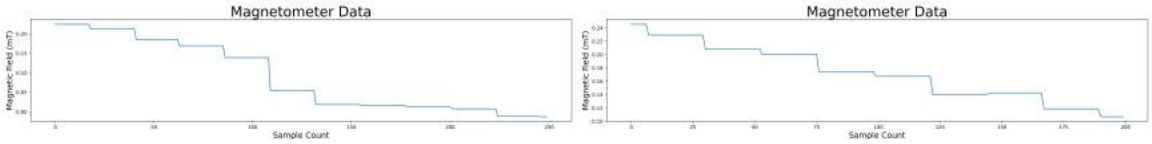


Figure 5.125: Magnetometer Comparison: Texture A and Texture B

The vibrations that occur during the dynamic tactile exploration can clearly be seen in the barometer, accelerometer, and gyroscope graphs in Figure 5.122, Figure 5.123 and Figure 5.124 respectively. However, the data obtained from the magnetometer shown

in Figure 5.125 is a bit ambiguous. For Texture A, even though clear vibrations can be seen, it is a bit difficult to recognize the object from the vibrations obtained from this method as the texture is a 3D texture. However, in Texture B, the pressure changes from the barometer and the vibrations from the accelerometer and the gyroscope are similar to the gratings on Texture B.

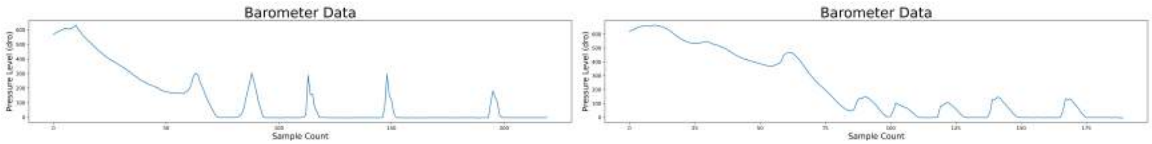


Figure 5.126: Barometer Comparison: Texture C and Texture D

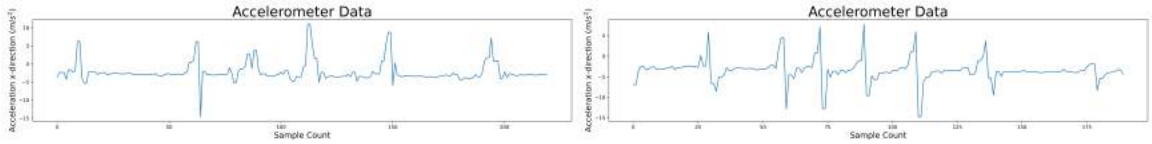


Figure 5.127: Accelerometer Comparison: Texture C and Texture D

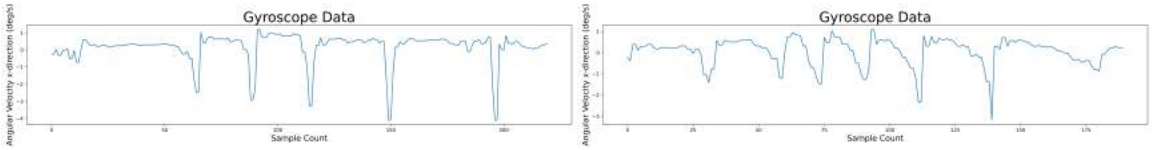


Figure 5.128: Gyroscope Comparison: Texture C and Texture D

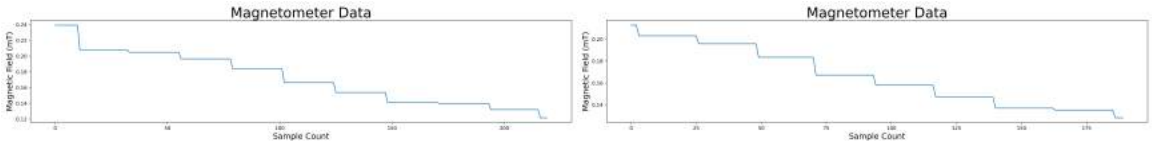


Figure 5.129: Magnetometer Comparison: Texture C and Texture D

The data obtained through dynamic tactile sensing for Texture C and Texture D from the barometer, accelerometer, gyroscope and magnetometer is shown in Figure 5.126, Figure 5.127, Figure 5.128 and Figure 5.129 respectively. The results look fairly similar but the frequency of pressure changes in the barometer in Figure 5.126 and

the frequency of the vibrations in Figure 5.127 and Figure 5.128 can be used to identify these textures. Texture C has a lower frequency while Texture D has a higher frequency which matches their gratings. Considering Textures B, C, and D it can be seen that they can be ordered from increasing frequency which can be used to identify their textures.

Chapter 6

Conclusion

This thesis explores the estimation of surfaces in unstructured environments using a robotic manipulator and a soft tactile sensing module. Historically, the exploration of environments by robots has been achieved by the use of computer vision. However, there are some issues that are difficult to solve through computer vision alone such as the presence of challenging light conditions and occlusion. In addition, it is impossible to measure certain features of objects such as hardness and tactile texture using computer vision. To this end, we proposed an approach using a bio-inspired tactile sensing module and a robotic manipulator to estimate the shape of objects in unstructured environments.

The proposed approach is able to obtain accurate poses of contact points by taking the data from the deformation of the sensing module. The contact orientation is obtained by applying the Madgwick filter on the data obtained from the MARG sensor and the contact position is obtained by utilizing the kinematic chain of the robotic manipulator. The approach taken to collect points allows the end-effector to remain constant; in our case perpendicular to the ground. This allows us to maintain the pitch of the end-effector at a neutral angle without having to follow the normal of the surface while points are taken on the surface. Instead, the deformation of the sensor is used to collect information about the normals of the surface. This approach can be exported to real-life scenarios where robots are expected to explore surfaces

in unstructured environments. This setup allows the robot to react to environmental uncertainties and can be used in situations where the robot cannot control the angle of approach. It also allows the exploration of unknown or fragile objects. Another point to note is that we were able to collect all the necessary data from one contact and multiple probes to the same contact point is unnecessary.

Using the collected data points and their surface normals obtained through static tactile exploration, we were able to calculate control points to define the curvature of the surface. These control points reduce the burden of collecting a large number of contact points as it leverages the geometry of the collected orientations and provide a good estimation of the curvature between contact points.

Using the collected contact points and the calculated control points, we use Bezier surface patches to estimate the surface in 3D. We were able to obtain similar estimates for 5 synthetic surfaces as well as 9 everyday objects including soft surfaces and transparent surfaces proving that this work could be applied in real-world scenarios. The obtained estimates were very similar to the actual surfaces and we were able to obtain estimates but there were a few limitations that were observed.

Very small curves in objects were not reflected in the estimations because of the size of the sensing module used. If the size of the sensing module was bigger than the surface curves, it was not reflected in the estimation. The solution to this would be to use a smaller sensor which would then be able to better reflect smaller changes in the surfaces. Objects that contained steep steps were not reflected well in their estimation due to the degree of freedom of the robot used. A robot with more degrees of freedom would also be able to estimate the sides of the surfaces using our approach.

After obtaining the estimate of a surface, we used dynamic tactile sensing to slide over the surface to obtain data about the texture of the surface. We used 4 different textures on the same surface curvature and were able to obtain good results which show the difference in the data collected for each texture.

The work done in this thesis is a proof of concept which proposes a method to perform surface estimation and texture exploration on unknown surfaces. However, further validation must be performed on a larger set of synthetic and everyday objects which is a task for future work.

Future work may look into embedding multiple sensing modules in robots. The

modules could be used in a hand-like structure with the sensing modules acting as the fingertips. Using this method, more data could be collected in a single probe and once an estimation is obtained, tasks such as grasping and manipulation will also become a possibility.

Another avenue to consider is to embed vision to this approach. Using vision, an initial estimate of the environment model could be obtained and probes using the tactile sensing module could be made more accurately only onto the needed objects which will save time and resources. Another approach would be to estimate objects using vision and then refine the estimate using our approach.

Bibliography

[1] V. Prado da Fonseca, T. E. Alves de Oliveira, and E. M. Petriu, "Estimating the Orientation of Objects from Tactile Sensing Data Using Machine Learning Methods and Visual Frames of Reference," *Sensors (Basel, Switzerland)*, vol. 19, no. 10, pp. 1–20, 2019.

[2] R. S. Fearing, "Tactile Mechanisms," *Tactile Sensing Mechanisms. The International Journal of Robotics Research*, vol. 9(3), no. 1985, pp. 3–23, 1990.

[3] R. D. Howe and M. R. Cutkosky, "Integrating tactile sensing with control for dextrous manipulation," *Proceedings of the IEEE International Workshop on Intelligent Motion Control, IMC 1990*, vol. 1, no. 90, pp. 369–374, 1990.

[4] R. S. Dahiya, P. Mittendorfer, M. Valle, G. Cheng, and V. J. Lumelsky, "Directions toward effective utilization of tactile skin: A review," *IEEE Sensors Journal*, vol. 13, no. 11, pp. 4121–4138, 2013.

[5] T. E. A. De Oliveira, A. M. Cretu, V. P. Da Fonseca, and E. M. Petriu, "Touch sensing for humanoid robots," *IEEE Instrumentation and Measurement Magazine*, vol. 18, no. 5, pp. 13–19, 2015.

[6] H. Dang, J. Weisz, and P. K. Allen, "Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5917–5922, 2011.

[7] A. Bicchi, J. K. Salisbury, and P. Dario, "Augmentation of grasp robustness using intrinsic tactile sensing," pp. 302–307, 1989.

[8] T. Shimizu, M. Shikida, K. Sato, K. Itoigawa, and Y. Hasegawa, "Micromachined active tactile sensor for detecting contact force and hardness of an object," *MHS*

2002 - Proceedings of 2002 International Symposium on Micromechatronics and Human Science, vol. 103, pp. 67–71, 2002.

[9] R. Crowder, "Toward robots that can sense texture by touch," *Science*, vol. 312, no. 5779, pp. 1478–1479, 2006.

[10] V. Maheshwari and R. F. Saraf, "High-Resolution Thin-Film Device to," vol. 312, no. June, pp. 1501–1504, 2006.

[11] D. Zou, Q. Cao, Z. Zhuang, H. Huang, R. Gao, and W. Qin, "An improved method for model-based training, detection and pose estimation of texture-less 3D objects in occlusion scenes," *Procedia CIRP*, vol. 83, pp. 541–546, 2019. [Online]. Available: https://doi.org/10.1016/j.procir.2019.04.115

[12] T. E. Alves de Oliveira, A. M. Cretu, and E. M. Petriu, "Multimodal Bio-Inspired Tactile Sensing Module for Surface Characterization," *Sensors (Basel, Switzerland)*, vol. 17, no. 6, pp. 1–19, 2017.

[13] T. E. A. De Oliveira, V. Prado Da Fonseca, E. Huluta, P. F. Rosa, and E. M. Petriu, "Data-driven analysis of kinaesthetic and tactile information for shape classification," *2015 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, CIVEMSA 2015*, pp. 2–6, 2015.

[14] L. C. Ramos, O. C. Silveira, T. E. De Oliveira, and P. F. Rosa, "Haptic Bio-Inspired Sensor Calibration for Surface Reconstruction," *IECON Proceedings (Industrial Electronics Conference)*, vol. 2019-October, pp. 663–668, 2019.

[15] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," *Mechatronics*, vol. 48, pp. 54–67, 2017.

[16] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 243–248, 2009.

[17] Z. Pezzementi, E. Plaku, C. Reyda, and G. D. Hager, "Tactile-object recognition from appearance information," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 473–487, 2011.

[18] H. Liu, J. Greco, X. Song, J. Bimbo, L. Seneviratne, and K. Althoefer, "Tactile image based contact shape recognition using neural network," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 138–143, 2012.

[19] U. Martinez-Hernandez, G. Metta, T. J. Dodd, T. J. Prescott, L. Natale, and N. F. Lepora, "Active contour following to explore object shape with robot touch," *2013 World Haptics Conference, WHC 2013*, pp. 341–346, 2013.

[20] M. Schöpfer, M. Pardowitz, and H. Ritter, "Using entropy for dimension reduction of tactile data," *2009 International Conference on Advanced Robotics, ICAR 2009*, 2009.

[21] S. Chitta, J. Sturm, M. Piccoli, and W. Burgard, "Tactile sensing for mobile manipulation," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 558–568, 2011.

[22] T. Corradi, P. Hall, and P. Iravani, "Bayesian tactile object recognition: Learning and recognising objects using a new inexpensive tactile sensor," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 3909–3914, 2015.

[23] R. Russell, "Object recognition by a 'smart' tactile sensor," *Proceedings of the Australian Conference on Robotics*, pp. 93–98, 2000. [Online]. Available: http://www.araa.asn.au/acra/acra2000/papers/paper17.pdf

[24] Y. Bekiroglu, J. Laaksonen, J. A. Jørgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 616–629, 2011.

[25] A. Drimus, G. Kootstra, A. Bilberg, and D. Kragic, "Design of a flexible tactile sensor for classification of rigid and deformable objects," *Robotics and Autonomous Systems*, vol. 62, no. 1, pp. 3–15, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2012.07.021

[26] S. Luo, X. Liu, K. Althoefer, and H. Liu, "Tactile Object Recognition with Semi-Supervised Learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9245, no. August, pp. 15–26. [Online]. Available: http://link.springer.com/10.1007/978-3-319-22876-1{_}2

[27] S. Luo, W. Mou, K. Althoefer, and H. Liu, "Novel Tactile-SIFT Descriptor for Object Shape Recognition," *IEEE Sensors Journal*, vol. 15, no. 9, pp. 5001–5009, 2015.

[28] S. Luo, W. Mou, M. Li, K. Althoefer, and H. Liu, "Rotation and translation invariant object recognition with a tactile sensor," *Proceedings of IEEE Sensors*, vol. 2014-Decem, no. December, pp. 1030–1033, 2014.

[29] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[30] S. Salti, F. Tombari, and L. Di Stefano, "SHOT: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2014.04.011

[31] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, nov 2004. [Online]. Available: http://link.springer.com/10.1023/B:VISI.0000029664.99615.94

[32] A. Khasnobish, G. Singh, A. Jati, A. Konar, and D. N. Tibarewala, "Object-shape recognition and 3D reconstruction from tactile sensor images," *Medical and Biological Engineering and Computing*, vol. 52, no. 4, pp. 353–362, 2014.

[33] G. Heidemann and M. Schöpfer, "Dynamic tactile sensing for object identification," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, no. 1, pp. 813–818, 2004.

[34] H. Liu, X. Song, T. Nanayakkara, L. D. Seneviratne, and K. Althoefer, "A computationally fast algorithm for local contact shape and pose classification using a tactile array sensor," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, may 2012, pp. 1410–1415. [Online]. Available: http://ieeexplore.ieee.org/document/6224872/

[35] H. Soh, Y. Su, and Y. Demiris, "Online spatio-temporal Gaussian process experts with application to tactile classification," *IEEE International Conference on Intelligent Robots and Systems*, no. December 2014, pp. 4489–4496, 2012.

[36] H. Soh and Y. Demiris, "Incrementally learning objects by touch: Online discriminative and generative models for tactile-based recognition," *IEEE Transactions on Haptics*, vol. 7, no. 4, pp. 512–525, 2014.

[37] M. Madry, L. Bo, D. Kragic, and D. Fox, "ST-HMP: Unsupervised Spatio-Temporal feature learning for tactile data," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2262–2269, 2014.

[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016.

[39] A. Schmitz, Y. Bansho, K. Noda, H. Iwata, T. Ogata, and S. Sugano, "Tactile object recognition using deep learning and dropout," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Febru, pp. 1044–1050, 2015.

[40] L. Cao, R. Kotagiri, F. Sun, H. Li, W. Huang, and Z. M. M. Aye, "Efficient spatio-temporal tactile object recognition with randomized tiling convolutional networks in a hierarchical fusion strategy," *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 3337–3345, 2016.

[41] P. K. Allen and K. S. Roberts, "Haptic object recognition using a multi-fingered dextrous hand," pp. 342–347, 1989.

[42] S. Caselli, C. Magnanini, and F. Zanichelli, "On the robustness of haptic object recognition based on polyhedral shape representations," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2, pp. 200–206, 1995.

[43] Y. B. Jia, L. Mi, and J. Tian, "Surface patch reconstruction via curve sampling," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, no. January 2006, pp. 1371–1377, 2006.

[44] R. Ibrayev and Y. B. Jia, "Semidifferential invariants for tactile recognition of algebraic curves," *International Journal of Robotics Research*, vol. 24, no. 11, pp. 951–969, 2005.

[45] M. Meier, M. Schöpfer, R. Haschke, and H. Ritter, "A probabilistic approach to tactile shape reconstruction," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 630–635, 2011.

[46] S. Luo, W. Mou, K. Althoefer, and H. Liu, "Iterative Closest Labeled Point for tactile object shape recognition," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 3137–3142, 2016.

[47] J. M. Gandarias, J. M. Gómez-de Gabriel, and A. J. García-Cerezo, "Enhancing perception with tactile object recognition in adaptive grippers for human-robot interaction," *Sensors (Switzerland)*, vol. 18, no. 3, 2018.

[48] J. M. Gandarias, A. J. Garcia-Cerezo, and J. M. Gomez-De-Gabriel, "CNN-Based Methods for Object Recognition with High-Resolution Tactile Sensors," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6872–6882, 2019.

[49] F. Pastor, J. M. Gandarias, A. J. García-Cerezo, and J. M. Gómez-De-gabriel, "Using 3d convolutional neural networks for tactile object recognition with robotic palpation," *Sensors (Switzerland)*, vol. 19, no. 24, 2019.

[50] S. Wang, J. Wu, X. Sun, W. Yuan, W. T. Freeman, J. B. Tenenbaum, and E. H. Adelson, "3D Shape Perception from Monocular Vision, Touch, and Shape Priors," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1606–1613, 2018.

[51] M. Bjorkman, Y. Bekiroglu, V. Hogman, and D. Kragic, "Enhancing visual perception of shape through tactile glances," *IEEE International Conference on Intelligent Robots and Systems*, no. May 2016, pp. 3180–3186, 2013.

[52] P. Falco, S. Lu, A. Cirillo, C. Natale, S. Pirozzi, and D. Lee, "Cross-modal visuo-tactile object recognition using robotic active exploration," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5273–5280, 2017.

[53] G. Rouhafzay and A. M. Cretu, "An application of deep learning to tactile data for object recognition under visual guidance," *Sensors (Switzerland)*, vol. 19, no. 7, 2019.

[54] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, pp. 707–714, 2006.

[55] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 569–585, 2011.

[56] C. Corcoran and R. Platt, "A measurement model for tracking hand-object state during dexterous manipulation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4302–4308, 2010.

[57] R. P. Jr, F. Permenter, and J. Pfeiffer, "Using Bayesian filtering to interpret tactile data during flexible materials manipulation," *IEEE Transactions on Robotics*, vol. 27, pp. 586–598, 2011.

[58] M. C. Koval, M. R. Dogar, N. S. Pollard, and S. S. Srinivasa, "Pose estimation for contact manipulation with manifold particle filters," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 34, no. 7. IEEE, nov 2013, pp. 4541–4548. [Online]. Available: http://ieeexplore.ieee.org/document/6697009/

[59] J. Bimbo, P. Kormushev, K. Althoefer, and H. Liu, "Global estimation of an objects pose using tactile sensing," *Advanced Robotics*, vol. 29, no. 5, pp. 363–374, 2015.

[60] G. Vezzani, U. Pattacini, G. Battistelli, L. Chisci, and L. Natale, "Memory Unscented Particle Filter for 6-DOF Tactile Localization," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1139–1155, 2017.

[61] J. Bimbo, S. Luo, K. Althoefer, and H. Liu, "In-Hand Object Pose Estimation Using Covariance-Based Tactile To Geometry Matching," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 570–577, 2016.

[62] R. Li, R. Platt, W. Yuan, A. Ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson, "Localization and manipulation of small parts using GelSight tactile sensing," *IEEE International Conference on Intelligent Robots and Systems*, vol. 1, no. Iros, pp. 3988–3993, 2014.

[63] J. Bimbo, S. Rodriguez-Jimenez, H. Liu, X. Song, N. Burrus, L. D. Senerivatne, M. Abderrahim, and K. Althoefer, "Object pose estimation and tracking by fusing visual and tactile information," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 65–70, 2012.

[64] J. Bimbo, L. D. Seneviratne, K. Althoefer, and H. Liu, "Combining touch and vision for the estimation of an object's pose during manipulation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4021–4026, 2013.

[65] T. Bhattacharjee, A. A. Shenoi, D. Park, J. M. Rehg, and C. C. Kemp, "Combining tactile sensing and vision for rapid haptic mapping," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 1200–1207, 2015.

[66] D. Álvarez, M. A. Roa, and L. Moreno, "Tactile-Based In-Hand Object Pose Estimation," *Advances in Intelligent Systems and Computing*, vol. 694, no. November 2019, pp. 716–728, 2018.

[67] T. Anzai and K. Takahashi, "Deep Gated Multi-modal Learning: In-hand Object Pose Estimation with Tactile and Image," 2019. [Online]. Available: http://arxiv.org/abs/1909.12494

[68] B. M. Rocha Lima, V. P. da Fonseca, T. E. A. de Oliveira, Q. Zhu, and E. M. Petriu, "Dynamic tactile exploration for texture classification using a miniaturized multi-modal tactile sensor and machine learning," *SYSCON 2020 - 14th Annual IEEE International Systems Conference, Proceedings*, 2020.

[69] S. H. Kim, J. Engel, C. Liu, and D. L. Jones, "Texture classification using a polymer-based MEMS tactile sensor," *Journal of Micromechanics and Microengineering*, vol. 15, no. 5, pp. 912–920, 2005.

[70] J. Edwards, J. Lawry, J. Rossiter, and C. Melhuish, "Extracting textural features from tactile sensors," *Bioinspiration and Biomimetics*, vol. 3, no. 3, 2008.

[71] E. Kerr, T. M. McGinnity, and S. Coleman, "Material recognition using tactile sensing," *Expert Systems with Applications*, vol. 94, pp. 94–111, 2018. [Online]. Available: https://doi.org/10.1016/j.eswa.2017.10.045

[72] M. Johnsson and C. Balkenius, "Sense of touch in robots with self-organizing maps," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 498–507, 2011.

[73] N. Jamali, P. Byrnes-Prestony, R. Salleh, and C. Sammut, "Texture recognition by tactile sensing," *Proceedings of the 2009 Australasian Conference on Robotics and Automation, ACRA 2009*, 2009.

[74] N. Jamali and C. Sammut, "Material classification by tactile sensing using surface textures," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2336–2341, 2010.

[75] V. A. Ho, T. Araki, M. Makikawa, and S. Hirai, "Experimental investigation of surface identification ability of a low-profile fabric tactile sensor," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4497–4504, 2012.

[76] H. Liu, X. Song, J. Bimbo, L. Seneviratne, and K. Althoefer, "Surface material recognition through haptic exploration using an intelligent contact sensing finger," *IEEE International Conference on Intelligent Robots and Systems*, pp. 52–57, 2012.

[77] J. M. Romano and K. J. Kuchenbecker, "Methods for robotic tool-mediated haptic surface recognition," *IEEE Haptics Symposium, HAPTICS*, pp. 49–56, 2014.

[78] M. Kaboli, P. Mittendorfer, V. Hugel, and G. Cheng, "Humanoids learn object properties from robust tactile feature descriptors via multi-modal artificial skin," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-February, pp. 187–192, 2015.

[79] R. Li and E. H. Adelson, "Sensing and recognizing surface textures using a gelsight sensor," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1241–1247, 2013.

[80] P. Dallaire, P. Giguère, D. Émond, and B. Chaib-Draa, "Autonomous tactile perception: A combined improved sensing and Bayesian nonparametric approach," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 422–435, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2013.11.011

[81] P. Dallaire, D. Émond, P. Giguere, and B. Chaib-Draa, "Artificial tactile perception for surface identification using a triple axis accelerometer probe," *ROSE 2011 - IEEE International Symposium on Robotic and Sensors Environments, Proceedings*, pp. 101–106, 2011.

[82] H. Orii, S. Tsuji, T. Kouda, and T. Kohama, "Tactile texture recognition using convolutional neural networks for time-series data of pressure and 6-axis acceleration sensor," *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 1076–1080, 2017.

[83] ——, "Recurrent neural network for tactile texture recognition using pressure and 6-axis acceleration sensor data," *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2017-December, pp. 2012–2016, 2017.

[84] B. Ward-Cherrier, N. Pestell, and N. F. Lepora, "NeuroTac: A Neuromorphic Optical Tactile Sensor applied to Texture Recognition," 2020. [Online]. Available: http://arxiv.org/abs/2003.00467

[85] W. Yuan, S. Wang, S. Dong, and E. Adelson, "Connecting Look and Feel: Associating the visual and tactile properties of physical materials," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, no. April 2017, pp. 4494–4502, 2017.

[86] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, "ViTac: Feature Sharing between Vision and Tactile Sensing for Cloth Texture Recognition," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2722–2727, 2018.

[87] ——, "Cloth Texture Recognition using Vision and Tactile Sensing," no. August, pp. 2–4, 2018.

[88] W. Yuan, Y. Mo, S. Wang, and E. H. Adelson, "Active clothing material perception using tactile sensing and deep learning," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, pp. 4842–4849, 2018.

[89] T. E. Alves De Oliveira, A. M. Cretu, and E. M. Petriu, "Multimodal Bio-Inspired Tactile Sensing Module," *IEEE Sensors Journal*, vol. 17, no. 11, pp. 3231–3243, 2017.

[90] T. E. A. De Oliveira, V. P. Da Fonseca, B. M. R. Lima, A. M. Cretu, and M. Petriu, "End-effector approach flexibilization in a surface approximation task using a bioinspired tactile sensing module," *IEEE International Symposium on Robotic and Sensors Environments, ROSE 2019 - Proceedings*, pp. 1–6, 2019.

[91] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *IEEE International Conference on Rehabilitation Robotics*, no. June, 2011.

[92] 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. [Online]. Available: https://doi.org/10.1016/j.softx.2018.12.005