

HIGH-DENSITY AND HOMOGENEOUS MAPPING USING A MOTION
CONTROLLED LIDAR

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İSMAİL MELİH CANBOLAT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

JANUARY 2023

Approval of the thesis:

**HIGH-DENSITY AND HOMOGENEOUS MAPPING USING A MOTION
CONTROLLED LIDAR**

submitted by **İSMAİL MELİH CANBOLAT** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. M. A. Sahir Arıkan
Head of the Department, **Mechanical Engineering** _____

Prof. Dr. Erhan İlhan Konukseven
Supervisor, **Mechanical Engineering, METU** _____

Examining Committee Members:

Assoc. Prof. Ahmet Buğra Koku
Mechanical Engineering, METU _____

Prof. Dr. Erhan İlhan Konukseven
Mechanical Engineering, METU _____

Assoc. Prof. Ali Emre Turgut
Mechanical Engineering, METU _____

Assoc. Prof. Kıvanç Azgın
Mechanical Engineering, METU _____

Assist. Prof. Andaç Töre Şamiloğlu
Mechanical Engineering, Başkent University _____

Date: 25.01.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Canbolat, İsmail Melih

Signature:

ABSTRACT

HIGH-DENSITY AND HOMOGENEOUS MAPPING USING A MOTION CONTROLLED LIDAR

Canbolat, İsmail Melih
Master of Science, Mechanical Engineering
Supervisor: Prof. Dr. Erhan İlhan Konukseven

January 2023, 99 pages

LiDAR is a commonly used method for mapping and obstacle detection in various robotics applications. With the advancement of technology, 3D scanning LiDAR sensors became widely available. Most 3D scanning LiDAR sensors have a 360° horizontal field of view, but they have a limited vertical field of view which is determined by the desired vertical scanning resolution and the number of laser scan layers. Limited number of horizontal layers in a LiDAR sensor introduces large gaps in the obtained point cloud map and the limited vertical field of view prevents stationary LiDAR sensors from being able to scan the environment completely. In this thesis work, a 3D scanning Velodyne Puck is used in a designed, motion controlled, two axis gimbal system. This motion control capability makes it possible to move the sensor in such a way that a more homogeneous, high-density point cloud map of the environment with increased and controlled field of view can be obtained. This improved point cloud map provides a good source of information to be used for object identification and SLAM algorithms. Additionally, with the help of the developed motion control algorithms for the designed system, the density

distribution of the point cloud map obtained by the designed system was successfully focused on certain regions by increasing the density of the point cloud data at these regions.

Keywords: Lidar, Velodyne Puck, Ros, Webots

ÖZ

HAREKET KONTROLLÜ LİDAR SENSÖRÜ KULLANARAK YÜKSEK YOĞUNLUKLU VE HOMOJEN HARİTALAMA

Canbolat, İsmail Melih
Yüksek Lisans, Makina Mühendisliği
Tez Yöneticisi: Prof. Dr. Erhan İlhan Konukseven

Ocak 2023, 99 sayfa

LiDAR, çeşitli robotik uygulamalarda haritalama ve engel tespiti için yaygın olarak kullanılan bir yöntemdir. Teknolojinin gelişmesiyle birlikte, 3B tarama yapabilen LiDAR sensörleri yaygın olarak kullanılmaya başlamıştır. 3D tarama yapabilen LiDAR sensörlerinin çoğu 360° yatay görüş alanına sahiptir, ancak istenen dikey tarama çözünürlüğü ve lazer tarama katmanlarının sayısı tarafından belirlenen sınırlı bir dikey görüş alanına sahiptirler. Bir LiDAR sensöründeki sınırlı sayıdaki yatay katman, elde edilen nokta bulutu haritasında büyük boşluklar yaratır ve sınırlı dikey görüş alanı, sabit LiDAR sensörlerinin ortamı tamamen taramasını engeller. Bu tez çalışmasında, Velodyne Puck LiDAR sensörü, tasarlanan, hareket kontrollü, iki eksenli bir gimbal sisteminde kullanılmıştır. Bu hareket kontrol yeteneği, sensörün yüksek yoğunluklu, daha homojen bir nokta bulutu haritası elde edilebilecek şekilde hareket ettirilmesini, dikey eksenindeki tarama çözünürlüğünün artırılmasını ve bu çözünürlüğün kontrol edilebilmesini mümkün kılmaktadır. Bu geliştirilmiş nokta bulutu haritası, nesne tanımlama ve eş zamanlı konumlandırma ve haritalama algoritmaları için kullanılacak iyi bir bilgi kaynağı sağlamaktadır. Ayrıca, tasarlanan sistem için geliştirilen hareket kontrol algoritmaları sayesinde, tasarlanan sistem

tarafından elde edilen nokta bulutu haritasının yoğunluk dağılımı, bazı bölgelerdeki nokta bulutu verilerinin yoğunluğu artırılarak, bu bölgelere başarıyla odaklanılmıştır.

Anahtar Kelimeler: Lidar, Velodyne Puck, Ros, Webots

To You

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Erhan İlhan KONUKSEVEN for his most valuable guidance, advice, criticism, encouragements, and insight throughout this thesis.

I am grateful to my family for their endless support throughout my education.

Finally, I would like to acknowledge opportunities provided by Hidroan Ankara Hidrolik Mak. San. Tic. Ltd. Şti. which allowed me to carry out this thesis study.

TABLE OF CONTENTS

ABSTRACT.....	v
ÖZ.....	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS.....	xi
LIST OF TABLES	xiv
LIST OF FIGURES	xv
LIST OF ABBREVIATIONS	xviii
LIST OF SYMBOLS	xx
CHAPTERS	
1 INTRODUCTION	1
1.1 LiDAR - Light Detection and Ranging.....	1
1.2 Mobile Robotics.....	3
1.3 Autonomous Vehicles	4
1.4 Motorized Gimbal	4
1.5 Robot Operating System – ROS	5
1.6 Webots Robotic Simulation	6
1.7 Aim of the Thesis.....	7
1.8 Outline of the Thesis	8
2 LITERATURE REVIEW	11
2.1 Use of 1D LiDAR for 2D and 3D Scanning.....	11
2.2 Use of 2D LiDAR for 3D Scanning.....	15
2.3 Moving 3D LiDAR.....	18

3	COMPONENTS	23
3.1	Velodyne Puck.....	23
3.2	BaseCamBGC Pro	25
3.3	BLDC Motors.....	27
3.4	AMS AS5048A Magnetic Encoder	28
3.5	CAN IMU	30
4	METHODOLOGY	31
4.1	Design of 2-Axis Gimbal System for a LiDAR	31
4.1.1	Problem Definition	31
4.1.2	Design Requirements	31
4.1.3	Concept Generation	33
4.1.4	Evaluation of Concepts	35
4.1.5	Designed System.....	37
4.1.6	Manufacturing and Assembly	40
4.2	Mathematical Modelling	42
4.2.1	Kinematic Modelling	42
4.3	BaseCamBGC Pro Controller Setup	52
4.3.1	General Setup.....	52
4.3.2	Serial Communication	54
4.3.3	PID Tuning	55
4.4	ROS Workspace	61
4.4.1	Developed Packages	61
4.4.2	Available Packages	63
4.4.3	ROS Node Graph	65

4.5	Robotic Simulation	65
5	EXPERIMENTATION AND RESULTS	71
5.1	Mapping an Indoor Environment.....	72
5.2	High Density Mapping of an Obstacle.....	77
5.3	Controlled Density Variation	79
5.4	Robotic Simulation Results.....	81
5.5	Focused Region Scans	84
6	CONCLUSION	91
6.1	Conclusion	91
6.2	Future Work	93
	REFERENCES	95

LIST OF TABLES

TABLES

Table 3.1 GB-PM120 Motor Specifications.....	28
Table 4.1 Evaluation of solutions for rotational motion generation.....	35
Table 4.2 Evaluation of solutions for rotational position measurement.....	36
Table 4.3 Evaluation of solutions for angular rate and orientation measurement...	36
Table 4.4 Evaluation of solutions for controllers for actuation and sensor reading	37
Table 4.5 PID Gains	56

LIST OF FIGURES

FIGURES

Figure 2.1 3D LiDAR Scanner Using Galvanometer Mirrors [11]	12
Figure 2.2 RE05 Scanner [12]	13
Figure 2.3 3D Scanner Using a Mirror and a Prism [13].....	14
Figure 2.4 Risley scanner for LiDAR Applications [14].....	15
Figure 2.5 3D Scanning Platform using Hokuyo URG-04LX [16].....	16
Figure 2.6 Swinging 2D LiDAR and Obtained Point Cloud [20].....	17
Figure 2.7 Nodding LiDAR Setup [23]	18
Figure 2.8 Laser Scanning Platform [24].....	19
Figure 2.9 Velodyne HDL-64E Tilt System [25]	20
Figure 2.10 Velomotion-16 [26]	20
Figure 2.11 Point Cloud Obtained by Tilting Velodyne Puck.....	21
Figure 2.12 Designed Rangefinder in [27] and in [28]	21
Figure 3.1 Velodyne Puck.....	23
Figure 3.2 Velodyne Puck Web User Interface	24
Figure 3.3 BaseCamBGC Pro [31]	26
Figure 3.4 GB-PM120 BLDC Motor.....	27
Figure 3.5 AMS AS5048A Magnetic Encoder [32]	29
Figure 3.6 Encoder Setup on the Motor.....	29
Figure 3.7 CAN IMU	30
Figure 3.8 Mounted IMU	30
Figure 4.1 Centered Design Concept	32
Figure 4.2 Configuration-1 (left) and Configuration-2 (right).....	38
Figure 4.3 Final Design Configuration-1	38
Figure 4.4 Final Design Configuration-2.....	39
Figure 4.5 Exploded View	40
Figure 4.6 Partially Assembled System	41
Figure 4.7 Assembled System.....	41

Figure 4.8 Manipulator Links.....	42
Figure 4.9 Kinematic Diagram.....	43
Figure 4.10 VLP-16 Sensor Coordinate System [30].....	44
Figure 4.11 System Architecture.....	53
Figure 4.12 Plotting Functionality of the Developed Python Package.....	55
Figure 4.13 PID auto-tuning.....	56
Figure 4.14 System Analysis Tool.....	57
Figure 4.15 System Analysis – Pitch.....	58
Figure 4.16 System Analysis – Yaw.....	58
Figure 4.17 Rest to Center.....	59
Figure 4.18 Slow Sinusoidal Motion.....	60
Figure 4.19 Fast Sinusoidal Motion.....	60
Figure 4.20 Sinusoidal Motion with Disturbance.....	61
Figure 4.21 ROS Node Graph.....	65
Figure 4.22 2-Axis Gimbal Webots Model.....	66
Figure 4.23 Webots Exploded View.....	66
Figure 4.24 Webots Point Cloud.....	67
Figure 4.25 Gimbal System with Pioneer 3-AT.....	67
Figure 4.26 Webots Encoder Plots.....	70
Figure 5.1 Test Setup.....	71
Figure 5.2 Stationary Instant Point Cloud Map.....	72
Figure 5.3 Stationary Accumulated Point Cloud Map.....	73
Figure 5.4 Accumulated Point Cloud Map Using the Designed System.....	74
Figure 5.5 Lidar Pitching Only.....	75
Figure 5.6 Point Cloud Map of the Building Using a Stationary LiDAR.....	76
Figure 5.7 Point Cloud Map of the Building Obtained by the Designed System ...	76
Figure 5.8 Placement of the Obstacle and the Sensor.....	77
Figure 5.9 Obstacle Scan with Stationary LiDAR.....	78
Figure 5.10 Obstacle Scan with Moving LiDAR.....	79
Figure 5.11 Constant Density Scan.....	80

Figure 5.12 Increased Point Cloud Density at the Center.....	80
Figure 5.13 Increased Point Cloud Density at the Bottom	81
Figure 5.14 Simulated Scan Coverage.....	82
Figure 5.15 Simulated Increased Density at the Center.....	83
Figure 5.16 Vertically Focused Scan	83
Figure 5.17 Constant Motion High Density Scan Region	85
Figure 5.18 Two-Band Focused Region	86
Figure 5.19 Three-Band Focused Region	86
Figure 5.20 Vertical Focus Region	87
Figure 5.21 Rectangular Scan Region.....	88
Figure 5.22 Rectangular Scan Region - 30°, 60° and 90° Wide.....	89
Figure 5.23 Rectangular Scan Region – Top View	89

LIST OF ABBREVIATIONS

ABBREVIATIONS

LiDAR: Light Detection and Ranging

AMCW: Amplitude Modulated Continuous Wave

FMCW: Frequency Modulated Continuous Wave

MEMS: Microelectromechanical Systems

RADAR: Radio Detection and Ranging

ROS: Robot Operating System

ODE: Open Dynamics Engine

Mbit/s: Megabit per second

UDP: User Datagram Protocol

PWM: Pulse Width Modulation

IMU: Inertial Measurement Unit

SLAM: Simultaneous Localization and Mapping

FOV: Field of View

UART: Universal Asynchronous Receiver-Transmitter

USB: Universal Serial Bus

API: Application Programming Interface

BLDC: Brushless Direct Current

CAN: Controller Area Network

IC: Integrated Circuit

GUI: Graphical User Interface

RPM: Revolution Per Minute

I/O: Input/Output

SLAM: Simultaneous Localization and Mapping

LIST OF SYMBOLS

SYMBOLS

α : Some angle

β : Some other angle

d : Distance

c : Speed of light

t : Time

θ : Joint angle

β : Twist angle

s_k : Offset between the common normals

b_k : Effective link length

d_{layer} : Distance between laser scan layers

CHAPTER 1

INTRODUCTION

1.1 LiDAR - Light Detection and Ranging

Autonomous vehicles and mobile robots need to obtain accurate range information of their environment. Correct distance measurement to surrounding objects is very crucial for safe operation of these platforms. One of the most common methods used for this purpose is the LiDAR technology. LiDAR stands for Light Detection and Ranging and uses the information of time-of-flight of light emitted by the sensor. In a LiDAR sensor, a laser is emitted from the sensor which is targeted at a certain known direction. When this laser hits a surface, it backscatters, and some of this reflection is then detected by the sensor. The amount of time for a laser to travel from sensor to the surface and then back to the sensor is used with the knowledge of speed of light to determine the distance. Simple formula to calculate the distance d can be given as,

$$d = \frac{c \times t}{2} \quad (1.1)$$

where c is the speed of light in the air and t is the travel time of the emitted laser.

In addition to distance measurement, LiDAR sensors can detect the intensity of the received laser light. This intensity is determined by the amount of scattering when the laser light is reflected from the surface and the reflectivity of that surface. Diffuse reflectors scatter the light more and thus, the received laser has lower intensity. On the other hand, retroreflectors have minimal scattering when reflecting the light which results in higher intensity of received laser light. This intensity knowledge can help identify the surface material and the object. For example, license plates and road

signs have retroreflective surfaces which can be identified if the intensity information is available.

Measurement of the time passed between emission and detection of the laser light can be achieved by three main approaches. First approach is to simply use a short laser pulse. The time of detection of the reflected pulse can be used to determine the total travel time with the knowledge of the time of emission of the that laser pulse. This way of measurement can provide resolutions at centimeter level. Second approach is to use a continuous laser beam with amplitude modulation which is called as amplitude modulation of a continuous wave (AMCW). In this approach, amplitude of the emitted laser beam and the amplitude of the detected backscattered laser beam is compared, and the time-of-flight is deduced. A resolution similar to that of pulse technique can be achieved in AMCW method but, the range of measurement is less. Final approach also uses a continuous laser beam but this time the frequency of the emitted laser beam is modulated. This technique is called the frequency modulated continuous wave (FMCW). Like AMCW, in FMCW, the frequencies of the emitted and detected laser beams are compared to deduce the time-of-flight. FMCW technique can achieve superior resolution with high precision compared to other two techniques [1].

LiDAR sensors use different mechanisms to perform the laser scanning operation. Most used mechanisms are electromechanical, micro-electromechanical (MEMS), opto-mechanical and solid-state scanning. Currently, the most dominantly used mechanism is the electro-mechanical scanning. In opto-mechanical scanning, optical components are used to deflect the laser beam in the desired direction and by controlling the motion of these components, 2D laser scanning can be obtained with a single laser rangefinder. In electro-mechanical systems, usually electric motors are used to increase the dimensionality of 2D LiDAR sensors by providing and controlling the motion of the whole sensor. MEMS scanning technology in LiDAR uses MEMS mirrors to control the laser beam. Finally, in solid-state scanning technology, various methods are utilized to steer the laser beam without using a

moving component. Some of these methods are optical phased array beam scanning and photonic crystal waveguide beam scanning [2].

Main alternatives to LiDAR method used to determine ranges in robotics and autonomous vehicle industry are RADAR and cameras. Some of the main advantages of LiDAR over its alternatives can be listed as the provided highly accurate long range distance measurements, light reflectivity measurement of the surfaces, and the ability to work at day and night without the need for illumination of the environment. Because of these advantages, LiDAR sensors are frequently used in robotics applications.

1.2 Mobile Robotics

Mobile robots are frequently used in various fields to complete various task with better speed and accuracy compared to humans. They can be used in harsh, unsafe environments to perform difficult tasks without risking any human life. They can help humans in their simple daily life and provide a more comfortable life. There are many more advantages that a mobile robot can provide to us.

To be able to navigate and position these mobile robots safely and reliably, it is imperative that the surrounding environment is accurately scanned. For this purpose, LiDAR sensors are frequently utilized in mobile robots. The data obtained from a LiDAR sensor can be used to detect obstacles and identify them in addition to mapping the environment and the localization of the robot in that map of the environment.

LiDAR sensors used in mobile robots have limited scanning capabilities and using more capable LiDAR sensors or using multiple LiDAR sensors can increase the cost too much that exceeds reasonable limits for a mobile robot. Thus, using a cheaper LiDAR and trying to increase its scanning capability with the help of a designed mechanism can be a good choice in mobile robotics.

1.3 Autonomous Vehicles

Nowadays there is a great focus on autonomous driving technology. Biggest car companies spend a considerable effort to improve autonomy of their products. Autonomous vehicles have the potential to decrease fatal car accidents most of which are caused by human error. It is also expected to reduce traffic jam by efficient management of the traffic by autonomous vehicles. Another advantage is that these vehicles can provide disabled people the ability to travel on their own. However, one of the main concerns related to autonomous vehicles is their safety. To provide the safe operation of autonomous vehicles, major factor is the ability to accurately sense the environment of the vehicle. For this purpose, various sensors are utilized, including LiDAR, RADAR, and cameras. LiDAR can give reliable, precise and fast range measurements. Therefore, almost all autonomous vehicles utilize some sort of LiDAR sensor to accurately scan their environment.

One of the biggest drawbacks of usage of LiDAR sensors in autonomous vehicles is their pricing. These sensors scan in a certain direction with limited resolution. To increase the coverage of these sensors and scan the environment of the vehicle with minimal blind spots, multiple LiDAR sensors are used in a vehicle, which increases the cost considerably. Ability to control the motion of a LiDAR sensor to change its pose as desired would be very beneficial to increase the field of view of and improve the scan density of a LiDAR sensor.

1.4 Motorized Gimbal

A gimbal can be defined as a device that allows an object to rotate about an axis. A three-axis gimbal system can be used to keep an object's orientation stationary irrespective of the motion of its support. When electric motors are used in each of the axes of the gimbal set, the motion of the object can be controlled in yaw, pitch, and roll axes. This type of gimbals can be frequently encountered for video cameras to stabilize the camera and also to move the camera in the desired direction in a stable

manner. In this study, a motorized 2-axis gimbal system is designed for a LiDAR sensor and this system is used to control the motion of the lidar to give a desired motion to the sensor. This motion capability improves the field of view of the sensor and allows us to obtain a more homogeneous, higher density point cloud data.

1.5 Robot Operating System – ROS

Robot Operating System (ROS), unlike the name implies, is not an operating system, but it can be defined as a robot framework or robot software development kit that requires a host operating system to work [3]. ROS is open-source and free and it provides a wide variety of software libraries and tools that are needed in robot applications. ROS provides the required drivers, algorithms, and communications layer for any robotic application. In addition to already available software and tools, users can develop their own software in ROS and share their work with others with ease.

ROS has two versions called ROS 1 and ROS 2. Unlike a distribution release, ROS 2 brought major changes to improve ROS by removing some mediocre features of ROS and introduced new features. Since its first release in 2007, ROS 1 had many distribution releases until 2020. In May 2020, the last ROS 1 distribution called “Noetic Ninjemys” was released which has an end-of-life date of May 2025. On the other hand, ROS 2 is under active development. First non-beta ROS 2 distribution was released in 2017 with the name of “Ardent Apalone”. The last ROS 2 distribution Humble Hawksbill was released in May 2022 and has an end-of-life date of May 2027 [4].

Communication pattern of ROS 2 application programming interface involves topics, services, and actions. Topics are message flow frameworks that allow publish-subscribe functionality using asynchronous messages. Another concept in ROS 2, nodes, publish and subscribe to these topics. The communication framework in ROS 2 allows many to many communications and thus makes the introspection

and development easier. On the other hand, services provide a request-response style communication unlike asynchronous messages on a topic. In addition to these two communication styles, a goal-oriented communication is also required in robotics applications. Actions provide a communication structure involving request, response, and feedback which is usually preferred for long running tasks [5].

1.6 Webots Robotic Simulation

Robotic simulators allow developers to quickly prototype and test their robotic systems in a simulated version of a physical environment. In addition to development of new robotic systems, robotic simulators can be used to develop and test new algorithms using available robotic platforms without requiring expensive equipment. Even though speed is preferred over accuracy in some simulation software, physical solutions given by these simulations are quite accurate to real world scenarios if the simulation environment is carefully constructed [6].

Webots [7] is a 3D robotics simulation software that can be used to rapidly prototype simulation environments. Robotic systems can easily be modelled, programmed, and simulated using Webots. After the simulation process, developed control programs can be transferred to real robots. In addition to its own programming interface to control the designed robots and the simulation environment, Webots provides ROS support. This support allows users to test their ROS software on the simulated environment and makes it easier to transfer this tested software to the real robot.

In December 2018, Webots is made a fully open source and free software. It is licensed under the Apache 2.0 License [8]. Being a free and open-source software makes Webots a popular choice for robotics developers.

Webots uses Open Dynamics Engine (ODE) as the physics engine to simulate the physical phenomena in the simulation environment. This physics engine allows Webots to simulate complex robotic systems including articulated robots. Webots has a large library of various sensors and actuators that can easily be integrated in

the simulated robot by carefully configuring their specifications. It also has some of the commercially available sensor models that can be easily imported and used in the designed robotic system [9]. These sensors and actuators are then can be controlled by user written controller programs. Webots supports various programming languages including C, C++, and Python. Users can also write supervisor programs that has limitless access to simulation environment unlike a controller program. Additionally, Webots provides a ROS interface which allows easy communication between the simulation software and the ROS workspace.

In a 2022 study [10], in which CoppeliaSim, Gazebo, MORSE, and Webots were compared quantitatively, it was concluded that for a high accuracy and high simulation efficiency, Webots was the best choice for a robotics simulation software among four alternatives. In this study, Webots is utilized to model and simulate the designed motion control system with a Velodyne Puck LiDAR sensor.

1.7 Aim of the Thesis

The aim of this thesis study is to develop a 2-axis motion-controlled gimbal system to be used with a Velodyne Puck LiDAR sensor to obtain high-density, homogeneous point cloud map of the environment and to increase the vertical field of view obtained from the sensor. It was aimed to obtain a better point cloud map of the environment which can be used for algorithms like SLAM and object identification to improve their performance. Additionally, the density of the obtained point cloud can be increased in certain regions to allow for a high-density scan of objects of interest and to get a high-resolution scan in certain directions.

Kinematic description of the designed system was also carried out in order to interpret sensor data accurately. Point cloud data obtained from the sensor in different orientations need to be transformed to obtain the point cloud map of the environment.

It was also aimed to use a robotic simulator in the process of developing the robotic system. In this study, Webots was used for this purpose. Using a simulator software for a robotic system improves the success of the development process and decreases the costs that occur during this process by allowing users to test the designed robotic system even before manufacturing the necessary parts and assembling the system. Furthermore, the simulation environment helps to visualize and analyze the point cloud that can be obtained by the system.

1.8 Outline of the Thesis

Chapter 1 briefly explains subjects related to this thesis work and points out their relationship to this study.

Chapter 2 provides the literature survey performed on similar studies. This chapter is divided into three parts. In the first part 2D and 3D scanning solutions using 1D LiDAR sensors are given. In the second part 3D scanning solutions using 2D LiDAR sensors are given. And in the last part, literature survey of use of 3D LiDAR sensors to obtain better scanning performance is provided.

In Chapter 3, components used in the designed system are explained.

In Chapter 4, the work done for this thesis is included in detail. Firstly, the design procedure for the system is thoroughly explained including manufacturing and assembly procedures. Then, mathematical modeling of the system is given in which forward kinematics and inverse kinematics of the system is realized. Then, the setup of the BaseCamBGC Pro controller board and the system architecture is presented. Developed python package for serial communication and PID tuning of the system are also given in this section. Then the main software developed for the system using ROS framework is introduced. Developed ROS packages and used ROS packages are explained in this part. Finally, the Webots robotic simulation model of the system is explained in the chapter.

Chapter 5 includes the mapping applications using the developed system and evaluates the performance of the system qualitatively. Additionally, the simulated system in Webots is also used to visualize and analyze the point cloud obtained by the designed system. Simulated point cloud data makes it easier to understand the obtained field of view from the sensor, point cloud distribution pattern and the density distribution in different motion control scenarios. Also, the application of the point cloud data focusing on certain regions in the point cloud map is shown in this chapter.

Finally, in Chapter 6, conclusions regarding this thesis study and possible future work are provided.

CHAPTER 2

LITERATURE REVIEW

2.1 Use of 1D LiDAR for 2D and 3D Scanning

A 1D LiDAR sensor, sometimes called laser rangefinder, uses a single non-moving laser to obtain depth measurements. This type of sensors does not involve any moving parts to deflect the laser beam to increase scanning dimensionality. They are fixed in a location to measure the distance in a certain direction.

For an additional dimensionality, usually, high-quality mirrors are utilized. Laser from a 1D LiDAR sensor hits the mirror and by rotating this mirror, the direction of the laser could be changed to provide a second scanning dimension. Additionally, by providing a second rotation axis to the rotating mirror, or by introducing another rotating mirror, 3D scans can be obtained by utilizing a single 1D LiDAR sensor. However, it should be noted that if a fast-scanning capability is needed, the resolution of the scans decreases. To obtain high resolution scans in these types of configurations, the laser beam needs to move slow enough to reach desired resolution. But this slow scanning speed is undesirable especially in dynamic environments.

This method of using optical components such as mirrors in LiDAR scanning is called opto-mechanical scanning. Main reason for choosing this approach is the size and weight of old LiDAR sensors. It is easier and less troublesome to control the motion of a smaller and lighter mirror than to move the whole sensor. Many commercially available LiDAR sensors use this approach.

In [11], a system is proposed that is capable of directing the laser beam from a 1D LiDAR sensor to an object of interest to obtain the 3D scan of that object. They have used two rotating galvanometer mirrors to direct the laser towards a certain, limited region. In their work, they have assumed that a limited information about the location and the geometry of the object of interest is known beforehand. Their selection of galvanometer mirrors that are 10mm in diameter resulted in decreased performance in detecting the reflected laser beam and decreased accuracy in measuring the distance, but they have been able to satisfy their requirements with the proposed system with the usage of very cheap components. Principle of operation of their system can be seen in Figure 2.1.

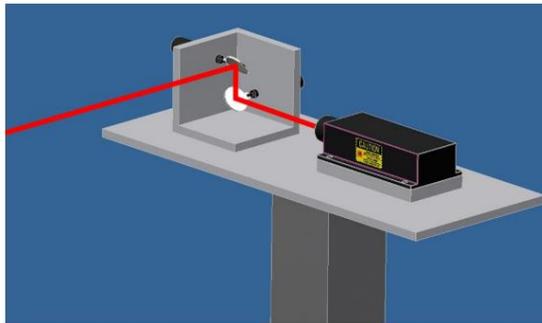


Figure 2.1 3D LiDAR Scanner Using Galvanometer Mirrors [11]

A company called Ocular Robotics, developed a 3D laser scanner using a 1D LiDAR and a unique mechanical design that allows a full 360° scan. The Figure 2.2 shows the system. According to their paper [12], their design owes its performance to its lightweight and well-balanced moving mass and stationary motors and the LiDAR module. One great feature of the designed system is the programmability of the scan region and scan resolution that it provides.

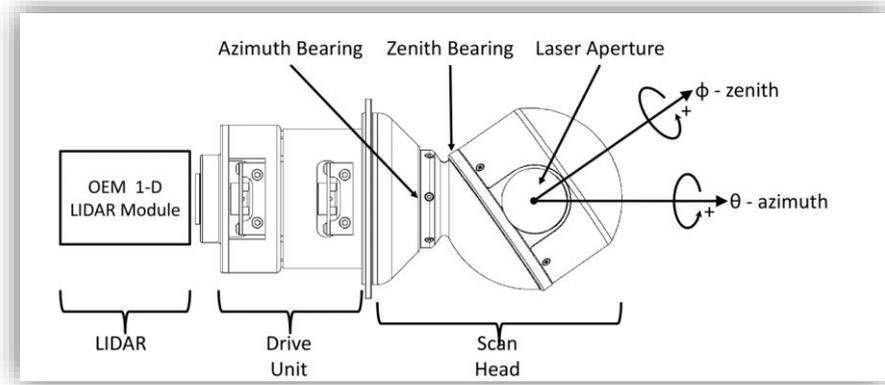


Figure 2.2 RE05 Scanner [12]

In a more recent study, Niewola and Podseǳkowski [13] used one rotating mirror and one rotating prism with a single laser rangefinder in their design. Field of view that they were able to obtain can be seen in Figure 2.3. They were able to obtain various scan patterns by separately controlling the rotational speed of the mirror and the prism. It was claimed in their paper that, they were able to get a better point cloud distribution than Velodyne Puck LiDAR. Velodyne Puck has 60 times greater scanning frequency than their solution. However, there are empty spaces between each of the 16 layers that Velodyne Puck scans. This drawback was eliminated in their solution but the resolution of the scans that they were able to get were much lower than Velodyne Puck. In this thesis work a Velodyne Puck LiDAR sensor will be used in a designed 2 axis motion-controlled gimbal system to overcome the drawback of having an unscanned region in between the scan layers of Velodyne Puck and similar LiDAR sensors while keeping the scan resolution the same. The system will also increase the vertical FOV of 30° of Velodyne Puck by controlling the orientation of the sensor as required.

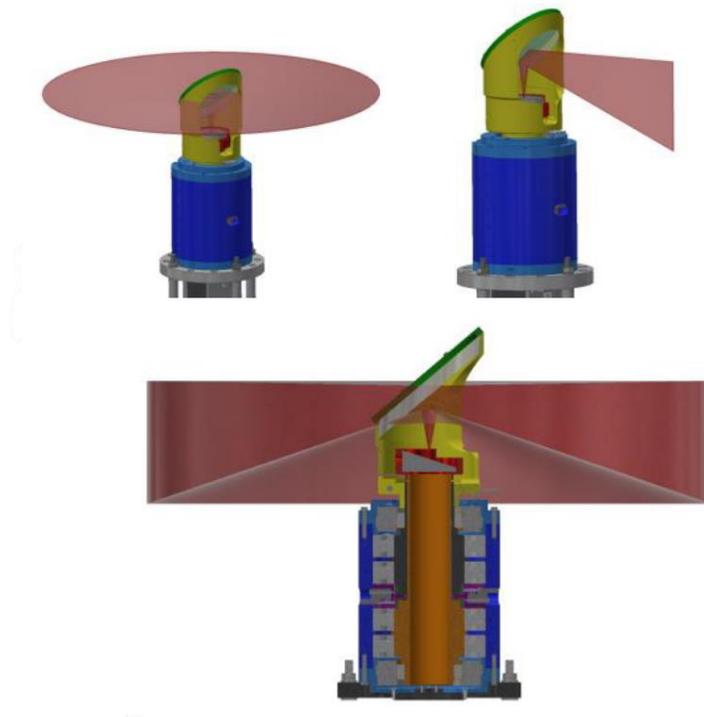


Figure 2.3 3D Scanner Using a Mirror and a Prism [13]

Another approach for controlling the direction of the laser beam is to use two rotating prisms which are called Risley prisms. Rotation of a single wedge prism directs the laser such that a circular scan pattern is obtained. Addition of another wedge prism and controlling the relative angular velocity between the prisms, a variety of scan patterns with the shape of rose curves can be obtained. In [14], a Risley scanner is designed to be used in LiDAR applications as shown in Figure 2.4.

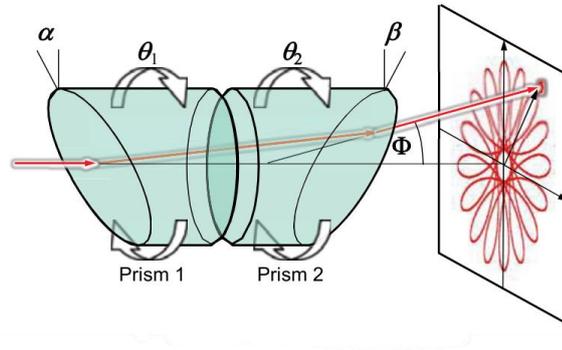


Figure 2.4 Risley scanner for LiDAR Applications [14]

The proposed system is a low-cost solution and can be very robust against external vibrations induced on the system. However, the region scanned is limited to the front of the sensor in a circular region and there is an inevitable blind spot at the center of the scan patterns generated by the system. Variations in point cloud density is also another problem with this method as can be seen in the figure above.

2.2 Use of 2D LiDAR for 3D Scanning

Another approach to increase the dimensionality of LiDAR sensors is to use a 2D planer LiDAR sensor and control its orientation with mechanical means, usually electric motors, to obtain a 3D scan of the environment. This configuration of laser scanning method is called electro-mechanical scanning.

Main motivation in increasing the dimensionality of a 2D LiDAR sensor to obtain 3D scans instead of using an already available 3D LiDAR is to decrease the cost of the application. According to [15], a moving 2D LiDAR while being very cost-effective in 3D scanning, cannot completely replace 3D LiDAR sensors in applications such as autonomous driving where a high real-time performance is required. It should also be noted, that a moving 2D LiDAR has the advantage of

having a controlled vertical field of view and resolution while stationary 3D LiDAR sensors do not.

A common way to obtain 3D scanning ability with a 2D LiDAR sensor is to rotate the sensor around pitch or roll axes. In a thesis work from our department, Aslan [16] designed and implemented a 3D scanning platform by the rotating a sideways placed 2D LiDAR sensor around the vertical axis. The sensor used was a Hokuyo URG-04LX which has a horizontal field-of-view of 240° and a maximum range of 4 meters. Its low power consumption, small form factor, and low weight makes it a good choice compact 3D scanning system. The designed system, as can be seen in Figure 2.5, is able to get an almost complete spherical scan of the environment except the bottom part which is caused by the preferred placement of the sensor and the structure of the designed system. The sensor was placed sideways looking upwards in this study, therefore the horizontal FOV of the sensor itself becomes the vertical FOV of the system. Manipulation of the rotation speed of the sensor about the z-axis allows the control of horizontal point cloud density. However, in the designed system, the density distribution of the point cloud obtained is not uniform.

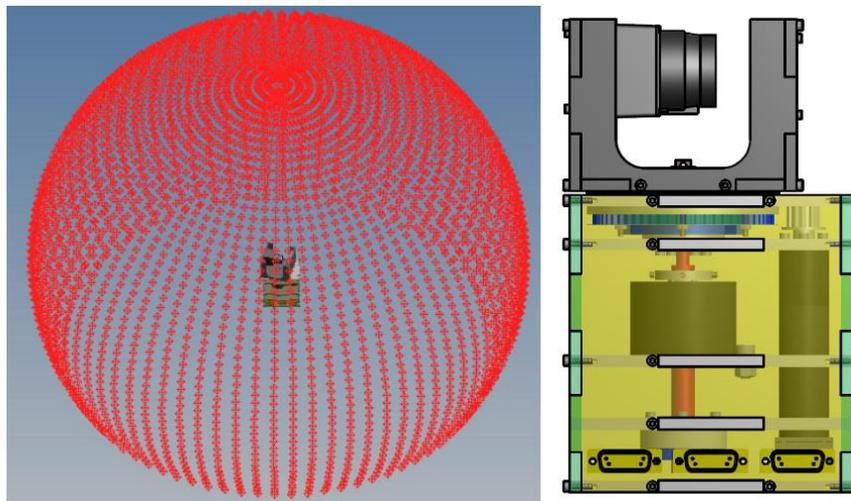


Figure 2.5 3D Scanning Platform using Hokuyo URG-04LX [16]

Another approach to obtain scans is to move the sensor obliquely. This way, instead of obtaining a pattern with parallel lines, an oblique grid pattern can be obtained. There are various studies [17][18][19] that utilize this motion which is usually called as roundly-swinging motion of the sensor. Yoshida et al. [20] developed a 3D laser scanner using a Hokuyo UTM-30LX 2D LiDAR in a gimbal mechanism that allows roll and pitch motions of the sensor. They have used a single motor to give a roundly swinging motion to the sensor as show in Figure 2.6 to obtain 3D scans of the environment. Main contribution of this work is the proposed method that provides the system with ability to gaze at a specific region of interest. Reducing the speed of the motor increases the resolution of scans in that region and thanks to the swinging motion of the LiDAR, they were able to focus on a specific region to get high density measurements from that region. Figure 2.6 clearly show the success of this method.

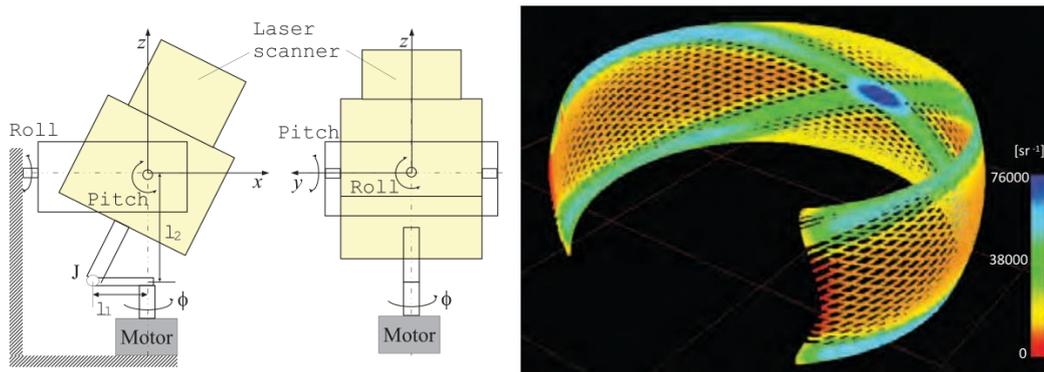


Figure 2.6 Swinging 2D LiDAR and Obtained Point Cloud [20]

It is also preferred in some cases to use two servo-controlled motors to precisely adjust the pitch and roll angles of the sensor to control the scanned region as needed. In [21], two servo motors are used to control the pitch and roll angles of 2D LiDAR. Ability to control the motion of the sensor as needed allowed the authors to develop an autonomous 2D SLAM and 3D mapping algorithm using the same sensor.

2.3 Moving 3D LiDAR

A 3D LiDAR can already be used to obtain 3D point cloud of the environment. However, most commonly used compact and relatively more affordable 3D LiDAR sensors like Velodyne Puck [22] has a certain limited vertical field of view and they scan at a certain number of horizontal layers which leads to unscanned regions between these layers. In applications where high-resolution point cloud of the environment is required it becomes a poor choice to use these sensors. And, in applications that require a full view of the environment, these sensors cannot be utilized due to their limited vertical field of view. And the sensors that provide higher resolution and higher scan range becomes bulkier and much more expensive. In order to overcome these limitations, various approaches were used in the literature.

Pfrunder et al. [23] used a Velodyne Puck LiDAR in their work in which they focused on the localization of a fully autonomous ground vehicle. They have used the setup shown in Figure 2.7 to overcome the limited vertical field of view of the sensor by providing a rotation to the sensor about a single axis. The sensor was nodded 30° up and down at a rotational speed of $70^\circ/\text{s}$. With this approach, they were able to increase the vertical field of view of the sensor.

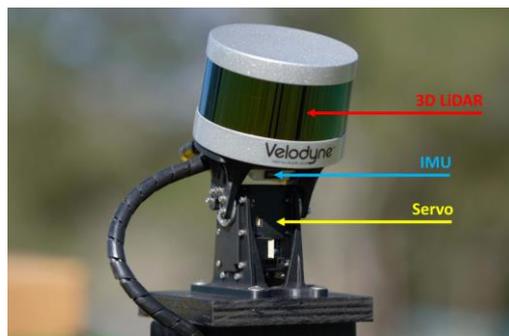


Figure 2.7 Nodding LiDAR Setup [23]

In another study [24], a novel rotating platform that can be used with 2D and 3D LiDAR sensors was presented. A Velodyne Puck, in addition to a 2D Hokuyo

LiDAR sensor for close range measurements, was used in their system. The design allows an increased vertical field of view and more uniform 3D point cloud distribution. The downside of the proposed system is the bulky design which is a problem for mobile robotics applications.

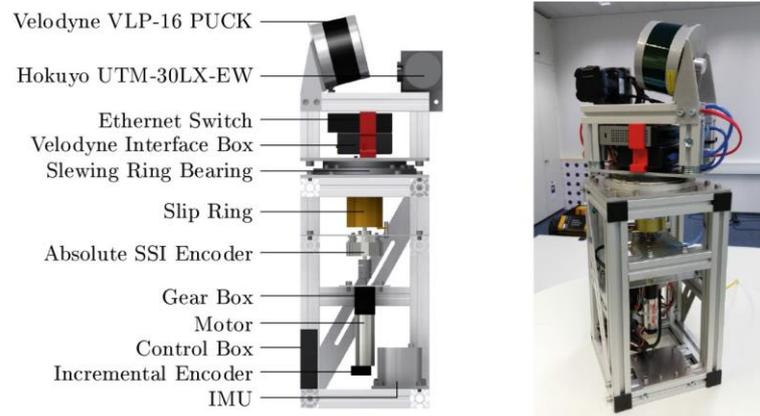


Figure 2.8 Laser Scanning Platform [24]

In [25], Velodyne HDL-64E that can scan in 64 horizontal planes was used in a mobile robot with a designed system that can tilt the sensor to increase its vertical resolution. The purpose of the study was to design a platform that can be used in high quality mapping applications of underground environments. The sensor used in the system is quite heavy with a weight of nearly 15 kilograms. For this reason, the authors preferred using a worm gear. However, this method increases inaccuracies in positioning of the sensor due to the backlash introduced by the worm gear. Their design in Gazebo robotic simulator and the robot itself can be seen in Figure 2.9.

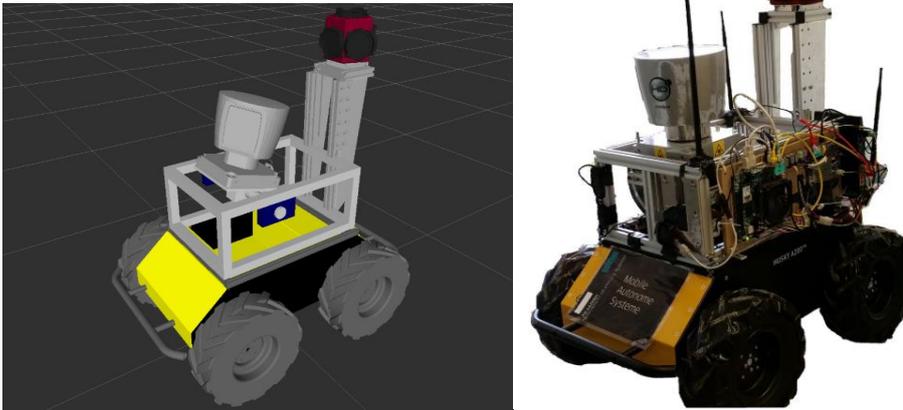


Figure 2.9 Velodyne HDL-64E Tilt System [25]

Morales et al. [26] designed and used a portable LiDAR tilting platform in their paper for the purpose of analyzing the distribution of the laser scan measurements obtained with that platform. They have used a Velodyne Puck in the mechanism with which they were able to tilt the sensor 45° with an angular velocity of up to $56.25^\circ/\text{s}$. This design allowed an increase in the vertical field of view of scanning range from 30° to 75° . They have successfully showed the importance of rotating a multi-beam LiDAR to describe the environment with much more detail. However, the single degree of freedom added by the mechanism prevent obtaining a more homogeneous point cloud distribution as can be seen in Figure 2.11.



Figure 2.10 Velomotion-16 [26]

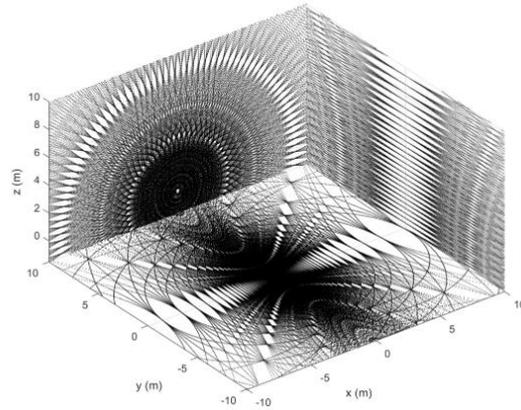


Figure 2.11 Point Cloud Obtained by Tilting Velodyne Puck [26]

A similar approach of providing tilt motion or nodding motion to Velodyne Puck LiDAR sensor has been also studied in [27]. In this mechanism, a continuously rotating motor and a crank linkage is used to provide the nodding motion to the sensor as shown in Figure 2.12. And in [28], a belt and pulley system is used to tilt the same sensor.

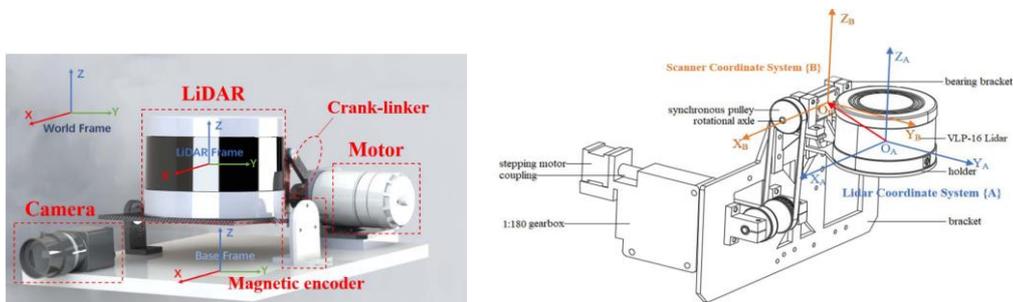


Figure 2.12 Designed Rangefinder in [27] and in [28]

In order to obtain dense point cloud data using a Velodyne Puck LiDAR sensor in terrestrial scanning applications, Bula et al. [29] placed the sensor on a ready to use motorized system which allows the rotation about a single axis. The sensor was placed sideways and rotated around the vertical axis which allowed the authors to manipulate the density of the point cloud data by controlling the rotation speed.

However, like previous solutions, uniformity of the obtained point cloud data is not satisfactory due to the provided rotational motion about a single axis perpendicular to the sensor coordinate axes.

CHAPTER 3

COMPONENTS

3.1 Velodyne Puck

In this thesis, a Velodyne Puck LiDAR sensor is used. Velodyne Puck (formerly known as the VLP-16) is one of the most commonly used 3D LiDAR sensors in both literature and commercial applications. Its compact design, low weight and low power consumption makes it a good choice for many applications. It uses the time-of-flight technique to get range measurements. In addition to providing a rich set of 3D point cloud data in real-time, the sensor also provides the reflectivity data for each point which can help identify the surface. The sensor uses 100 Mbit/s Ethernet connection to transfer its data in the form of UDP packets.



Figure 3.1 Velodyne Puck

Main specifications of the sensor are as follows [22]:

- 16 Channels
- 100 m Measurement Range
- ± 3 cm Range Accuracy
- 360° Horizontal FOV – Up to 0.1° Horizontal Angular Resolution

- 30° Vertical FOV - 2° Vertical Angular Resolution
- 5 to 20 Hz Rotational Rate
- 8 W Power Consumption
- ~830 g Weight
- ~300000 Scan Points Per Second

Web interface of the sensor can be used to configure the sensor according to the needs of the application. The connection of the sensor with the computer is made using an ethernet cable and after configuring the network settings of the computer as explained in the user manual, sensor's web interface can be accessed as shown in Figure 3.2. Using this interface, laser status, return type, motor RPM, vertical FOV, network settings and many more can be configured.

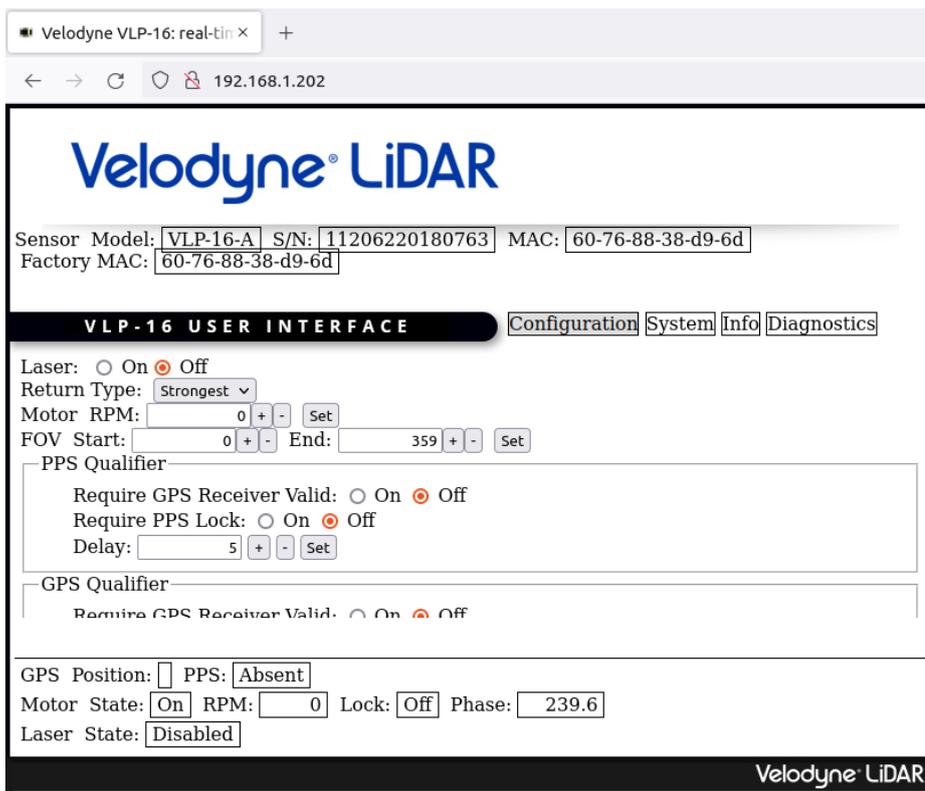


Figure 3.2 Velodyne Puck Web User Interface

In a laser range measurement, usually multiple returns of the emitted laser beam can be detected because of beam divergence. As the distance increases, the diameter of the laser beam increases enough to hit multiple objects resulting in multiple returns and multiple distance measurements. Velodyne Puck supports three return modes, strongest, last, and dual. It analyzes multiple returns and depending on the configuration, returns distance data using either one of the three modes. In dual return mode, the sensor reports both strongest and last returns resulting in doubled data rate.

Motor RPM setting can be used to adjust the rotational speed of the motor inside the sensor. This speed can be selected between 300 RPM and 1200 RPM with increments of 60 RPM. Variations in rotational speed directly affects the horizontal resolution of the sensor which can be calculated in degrees using the following formula from the user manual [30].

$$Res_{Horizontal} = RPM \times \frac{1 \text{ min}}{60 \text{ s}} \times \frac{360^\circ}{\text{rev}} \times \frac{55.296 \times 10^{-6} \text{ s}}{\text{firing cycle}} \quad (3.1)$$

3.2 BaseCamBGC Pro

BaseCamBGC Pro is a controller board designed to be used for up-to 3-axis gimbal systems that can be used to keep a camera stable in the desired orientation. The controller has a sufficient processing power with a 32-bit MCU ARM Cortex M4 processor that runs at 72MHz frequency. It has high-efficiency field-effect transistors that are used to drive up-to three brushless DC motors with 13 A current consumption per motor and allows users to connect up absolute rotary encoders for each axis of rotation and an external IMU sensor through CAN bus connector on the board. The controller board have many other connection capabilities like analog input pins, PWM input and output pins, Bluetooth, etc., which will not be needed in this work.



Figure 3.3 BaseCamBGC Pro [31]

The controller can be used with a designed 1, 2 or 3 axis gimbal system with brushless DC motors and it tries to keep the device connected to this gimbal system in the desired position by controlling its orientation in roll, pitch, and yaw axes. It uses PID controller to achieve this and the parameters of this controller can be adjusted by the user. For the feedback needed in the PID controller, angle measurements of the IMU sensor are the main source of information. Additionally, enabling encoders connected to the system improves the performance of the controller by using the feedback from these encoders. The board also features a PID auto-tune functionality to make it easier to determine PID parameters for the system at hand optimal performance.

Another very important feature of the controller is the adaptive PID control functionality that it provides. The controller may be very well tuned with high PID gains for the expected operating range, but it may become unstable when this range is exceeded due to some unexpected disturbances. In this scenario, in addition to losing position control, self-excitations may cause strong vibrations that can damage the system and the sensor connected to it. At this point adaptive PID control comes to the rescue which adaptively decreases the PID gains until the stability of the system is achieved again and then returns the PID gains back to the normal values.

For the adaptive PID control, the board allows the adjustment of RMS error threshold, attenuation rate and recovery factor.

BaseCamBGC Pro can be used offline after making necessary adjustment through its graphical user interface if the only requirement is to keep the sensor stabilized. However, when it is needed to control the motion of the system and to read sensor data like, position and acceleration, a communication method with the board is required. The board can be connected to a computer using the USB port on the controller and the serial communication can be established with UART interface. There is a serial API protocol specification documentation which can be used to create serial messages necessary to communicate with the controller, get real-time data and adjust the variables like PID parameters, target positions, acceleration limits, motor power limits etc.

BaseCamBGC Pro has been chosen to be used in the system designed for this thesis for being a compact and quick solution that answers all the requirements expected from the designed mechanism.

3.3 BLDC Motors

The designed two axis gimbal mechanism uses two inrunner GB-PM120 BLDC motors shown in Figure 3.4 to provide rotation in two axes. Main features of these motors are given in the Table 3.1.



Figure 3.4 GB-PM120 BLDC Motor

Table 3.1 GB-PM120 Motor Specifications

Specification	<i>Value</i>	<i>Unit</i>
Nominal Voltage	24	V
Nominal Current	4	A
Nominal Speed	270	rpm
Nominal Torque	6	N·m
Stall Current	11.2	A
Stall Torque	6.4	N·m
Winding Connection	Y	
Weight	1.3	kg

Main advantages of using these motors are that they are backlash free as they are direct drive motors, and they are quite powerful which allows us to obtain a better control of unbalanced masses.

3.4 AMS AS5048A Magnetic Encoder

In order to precisely measure the absolute angular position and velocity of each motor in the system, two AMS AS5048A magnetic encoders are used. The AMS AS5048A is a contactless rotary position sensor with 14-bit high resolution. It supports SPI, I2C and PWM and works at both 3.3 V and 5 V. This sensor uses the orientation of the magnetic field to measure the absolute rotary position as illustrated in Figure 3.5. IMU can also provide the angular orientation of the LiDAR connected to the gimbal system, however, encoders are necessary when the relative position of the LiDAR with respect to the base frame is needed.

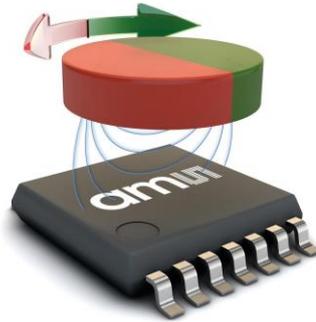


Figure 3.5 AMS AS5048A Magnetic Encoder [32]

A cross-section of one of the motors is given in Figure 3.6. which shows how the sensor-magnet-motor configuration in the designed system is made. The sensor is placed at the back of the motor facing the hollow center of the motor. An aluminum cylinder that holds the magnet with the correct polarity is connected to the rotor of the motor. With this setup, the task of accurate absolute measurement of angular position of the motor shaft is accomplished.

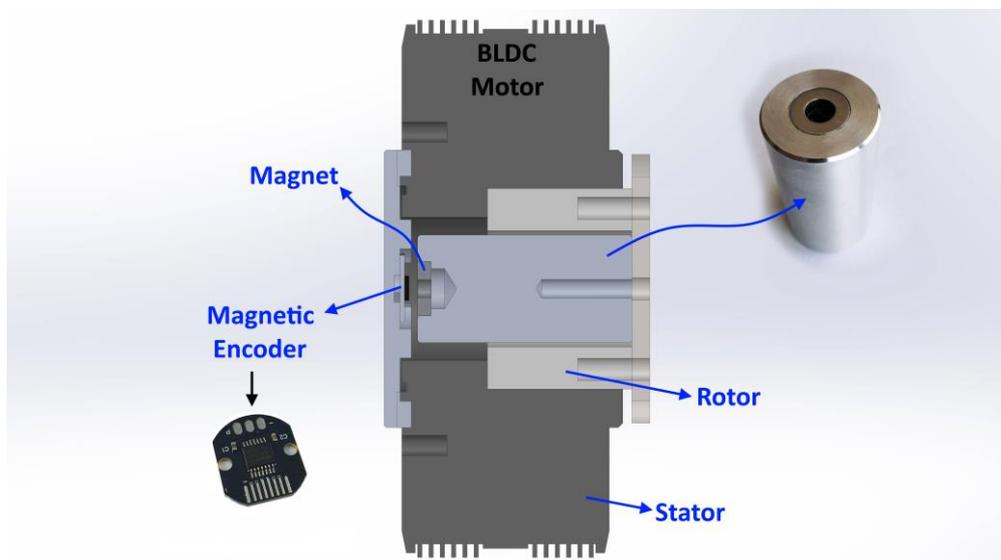


Figure 3.6 Encoder Setup on the Motor

3.5 CAN IMU

The designed gimbal system has a CAN IMU sensor board shown in Figure 3.7 is designed to be used with the BaseCamBGC Pro. This board has a high precision 6-axis MEMS IMU which provides 3-axis gyroscope data with 19-bit resolution and a 3-axis accelerometer data with 18-bit resolution. The sensor board also has a CAN Transceiver IC that allows CAN bus connectivity of the sensor board with the BaseCamBGC Pro.



Figure 3.7 CAN IMU

This sensor is the main source of feedback used for controlling the position of the LiDAR sensor in the designed gimbal system, while encoders are used to improve the performance of the PID controller. This IMU is placed on the arm that holds the LiDAR as can be seen in Figure 3.8.

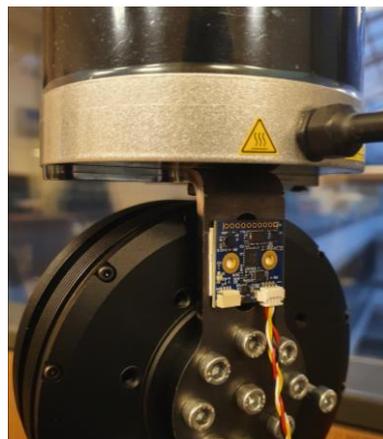


Figure 3.8 Mounted IMU

CHAPTER 4

METHODOLOGY

4.1 Design of 2-Axis Gimbal System for a LiDAR

This part of the chapter contains the mechanical design procedure including the manufacturing and the assembly of the designed system.

4.1.1 Problem Definition

Main purpose of the design procedure in this study is to design a mechanical system that can control the orientation and motion of a LiDAR with sufficient performance. The sensor focused on is a 3D scanning Velodyne Puck LiDAR sensor which was readily available to use for this study.

4.1.2 Design Requirements

Velodyne Puck has a horizontal field of view of 360° . The designed system should not block the field of view of the sensor. Because of this requirement, a regular gimbal in which the center of mass of the object intersects the axes of the gimbal system cannot be used. A sample configuration illustrating this situation is given in Figure 4.1 in which the horizontal field of view of the sensor is greatly blocked. Moving the center of mass of the rotating object away from the field of view of the sensor by utilizing a counter mass connected opposite to the sensor is not feasible as that would increase the weight of system too much and limits the motion of the system. As a result, it is determined that the sensor needs to be placed away from the arms and the motors of the designed system, which makes the system behave like an

inverted pendulum. This design decision while making the control of the system harder, allows us to use the full horizontal scanning capability of the sensor.

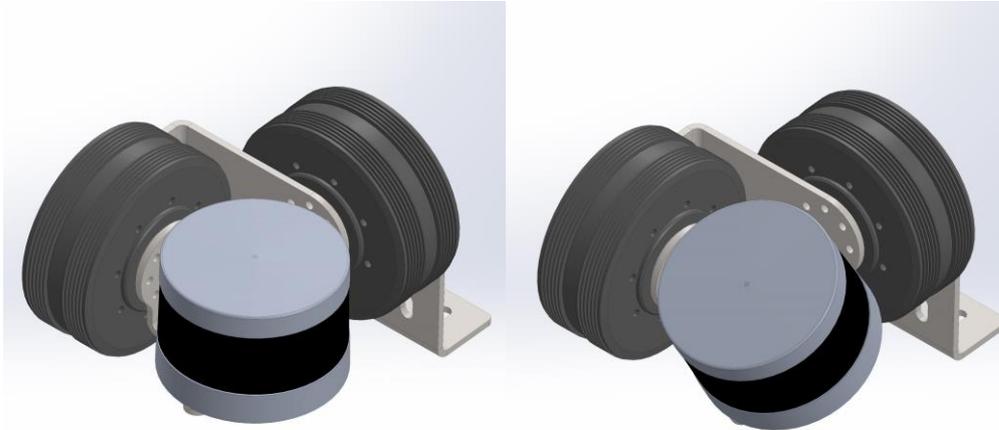


Figure 4.1 Centered Design Concept

A gimbal system that provides motion in three axes can be used to control motion in yaw, pitch, and roll axes. For a sensor that has a limited horizontal field of view, this motion capability would be advantageous. However, the horizontal field of view of 360° in Velodyne Puck makes the third axis of motion redundant. To decrease the system complexity and the total cost of the system it is concluded that a 2-axis gimbal system was needed. These two axes should allow the sensor to perform tilt and roll motions or pan and tilt motions. Considering the horizontal FOV of Velodyne Puck, panning may seem unnecessary at first, however, in some scenarios pan and tilt motion could be useful if the pan motion is used in combination with the tilt motion as can be seen in Figure 4.3.

The system needs to provide sufficient rotational freedom. It is expected from the system to be able to rotate the sensor at least $\pm 45^\circ$ in both axes.

Since it is planned to perform oscillatory motion when rotating the sensor, it is more reasonable to define the frequency of oscillations rather than the angular speed. The system is expected to be able reach 1 Hz when scanning by rotating the sensor. It

means that after 1 second, the sensor should be able to come back to the starting point. If both axes are rotated at this rate, a full scan by the system would be obtained in 1 second. However, the system can be used more slowly to increase the density of point cloud measurements taken from the environment.

It is also necessary to know the location of the sensor with respect to the base frame in order to accurately interpret sensor readings. For this purpose, encoders are needed for both axes.

In addition to encoders, it is necessary to know the orientation of the sensor with respect to the earth fixed reference frame. Thus, an IMU will be utilized in the system.

It was also desired to use the available equipment as much as possible for the system to be designed. This can decrease the cost of the project and also eliminates the delays caused by shipment processes of the products.

Abovementioned design requirements can be summarized as below.

Product design specifications:

- Oscillation Frequency: 1 Hz
- 360° Unblocked Horizontal FOV
- Rotational motion in two axes
- Weight < 5 kg
- Position Measurement with respect to the base frame
- Position Measurement with respect to the earth fixed reference frame

4.1.3 Concept Generation

The desired system functionality can be decomposed into following sub-functions:

- Laser range finding

- Rotational motion generation
- Rotational position measurement
- Angular rate and orientation measurement
- Controller for actuation and sensor reading

For each sub-function several possible solutions can be generated. Some alternative solutions are given as:

- Laser range finding
 - 1D LiDAR Sensor
 - 2D LiDAR Sensor
 - 3D LiDAR Sensor
 - Solid State LiDAR Sensor
- Rotational motion generation
 - Stepper Motor
 - Brushed DC Motor
 - Brushless DC Motor
 - Servo Motor
- Rotational position measurement
 - Optical Rotary Encoders
 - Magnetic Encoders
 - Potentiometric Position Sensors
- Angular rate and orientation measurement
 - MEMS IMU
 - FOG - fiber optic gyros
 - RLG - ring laser gyros

- Controllers for actuation and sensor reading
 - Arduino Mega
 - Raspberry Pi 4
 - BaseCamBGC Pro
 - NodeMCU

4.1.4 Evaluation of Concepts

Generated solutions are evaluated using different evaluation criteria with different weight factors. However, since Velodyne Puck is aimed to be used in this design, evaluation regarding different solutions for laser range finding will not be considered. Some of the evaluation criteria that need to be explained are explained below the respective table and the others are already self-explanatory. Weight factor for each criterion is written next to it in each table. In these tables *S* represents the score between 1 and 5 that the solution gets for each criterion and *R* represents the rating that the solution gets for that criterion. It is calculated as the multiplication of weight factor and the score.

Table 4.1 Evaluation of solutions for rotational motion generation

Evaluation Criteria	Weight Factor	<i>Stepper Motor</i>		<i>Brushed DC Motor</i>		<i>Brushless DC Motor</i>		<i>Servo Motor</i>	
		<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>
Cost	0.25	4	1.00	5	1.25	4	1.00	3	0.75
Size	0.10	4	0.40	3	0.30	5	0.50	2	0.20
Weight	0.15	3	0.45	3	0.45	3	0.45	3	0.45
Durability	0.30	5	1.50	2	0.60	5	1.50	2	0.60
Accuracy	0.40	4	1.60	3	1.20	4	1.60	4	1.60
Total	1		4.95		3.80		5.05		3.60

Table 4.2 Evaluation of solutions for rotational position measurement

Evaluation Criteria	Weight Factor	<i>Optical Rotary Encoders</i>		<i>Magnetic Encoders</i>		<i>Potentiometric Position Sensors</i>	
		<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>
Cost	0.15	5	0.75	4	0.60	5	0.75
Size	0.10	3	0.30	5	0.50	4	0.40
Weight	0.05	3	0.15	5	0.25	3	0.15
Durability	0.20	5	1.00	5	1.00	2	0.40
Accuracy	0.30	4	1.20	4	1.20	2	0.60
Friction	0.20	4	0.80	5	1.00	2	0.40
Total	1		4.20		4.55		2.70

- Friction criteria consider the friction introduced by the product on the system that resist rotation on each axis.

Table 4.3 Evaluation of solutions for angular rate and orientation measurement

Evaluation Criteria	Weight Factor	<i>MEMS</i>		<i>FOG</i>		<i>RLG</i>	
		<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>
Cost	0.20	5	1.00	3	0.60	1	0.20
Size	0.20	5	1.00	4	0.80	3	0.60
Weight	0.15	5	0.75	3	0.45	2	0.30
Availability	0.30	5	1.50	4	1.20	1	0.30
Accuracy	0.15	3	0.45	5	0.75	4	0.60
Total	1		4.70		3.80		2.00

Table 4.4 Evaluation of solutions for controllers for actuation and sensor reading

Evaluation Criteria	Weight Factor	<i>Arduino Mega</i>		<i>Raspberry Pi 4</i>		<i>BaseCamBGC Pro</i>		<i>NodeMCU</i>	
		<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>	<i>S</i>	<i>R</i>
Cost	0.10	5	0.50	4	0.40	3	0.30	5	0.50
Availability	0.25	5	1.25	2	0.50	5	1.00	5	1.25
Connectivity	0.25	4	1.00	3	0.75	5	1.25	4	1.00
Functionality	0.25	2	0.50	3	0.75	5	1.25	4	1.00
Reliability	0.15	4	0.60	4	0.60	5	0.75	4	0.60
Total	1		3.85		3.00		4.55		4.35

- Availability is determined by the market availability and if the product is already available to be used.
- Connectivity covers number and variety of I/O connectors of the product.
- Functionality criteria considers additional functionality of the board, which include motor drivers, voltage regulators, wireless communication components, etc.

According to the performed evaluation of solutions for each sub-function, best concept is found to be the combination of the following.

- *Velodyne Puck*
- *Brushless DC Motor*
- *Magnetic Encoders*
- *MEMS IMU*
- *BaseCamBGC Pro*

4.1.5 Designed System

According to the design requirements, a 2-axis gimbal system to control the motion of a Velodyne Puck LiDAR is designed, simulated in Webots, and manufactured.

The designed system has a great advantage that it can be utilized in two possible configurations. In the Figure 4.2, these two configurations and their corresponding rotation axes can be seen.

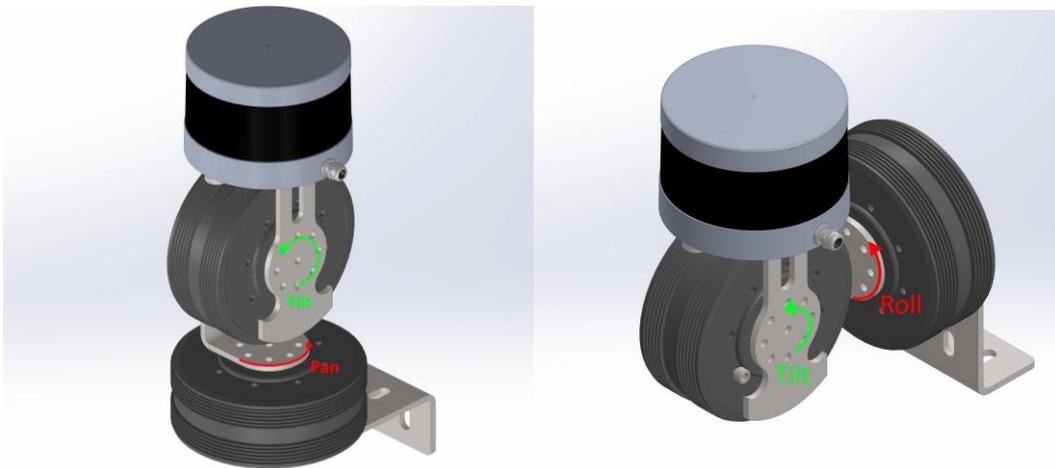


Figure 4.2 Configuration-1 (left) and Configuration-2 (right)

In the first configuration, the system can provide motion to the sensor in pan (yaw) and tilt (pitch) axes. This configuration can also be seen in Figure 4.3 in various orientations. When the sensor is straight, the motion in the pan axis has no advantage as the sensor can already scan 360° horizontally. However, when the sensor has a slight tilt, panning motion becomes important and helps the sensor scan in the desired direction which is needed to manipulate the scan pattern obtained by this system. With this design configuration, in the pan axis, a full rotation capability can be obtained while in the tilt axis approximately 225° of rotation can be possible.

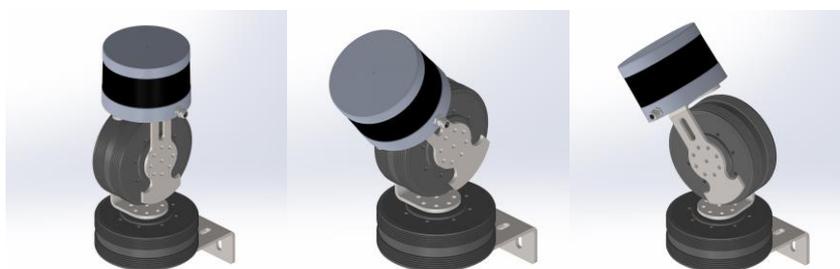


Figure 4.3 Final Design Configuration-1

In the second configuration, the designed system can provide motion in roll and tilt axes. This configuration of the designed system can be seen in Figure 4.4 in various orientations. In this configuration, tilt motion affects the scans in the front and the back of the sensor while the tilt motion affects the scans in the right and the left of the sensor. In this configuration the sensor can still be rotated within limits similar to the previous configuration. However, when the placement of the system on a vehicle is considered, some limits would be introduced. In the roll axis, a rotation more than 180° would be redundant since the sensor can scan 360° . For the tilt axis, behind the sensor, the first motor is present which limits the motion. In the other direction, 90° from the straight position would be more than enough as more rotation would lead to scanning the vehicle to which the sensor is mounted but not the ground. In this configuration, controlling the system becomes harder than the previous configuration. In the current configuration, both motors are exposed to varying loads depending on the rotation of the system, while in the first configuration, only a single motor needs to deal with a varying load. Considering the holding torque requirements, this also has an impact on the power consumption of the system.

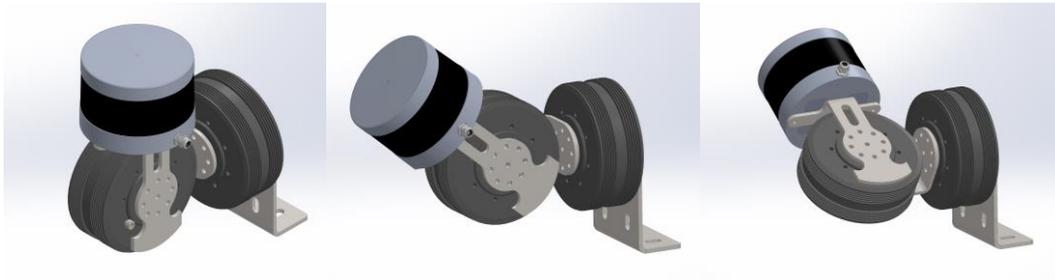


Figure 4.4 Final Design Configuration-2

4.1.6 Manufacturing and Assembly

Three connection parts in the system are manufactured from 4 mm thick S235JR steel sheet metal. This sheet metal was laser cut and then bent to the desired shape. Plate and the rod that holds the encoder and the magnet for the encoder respectively are manufactured from Aluminum using CNC milling and turning operations. Magnet required by the magnetic encoder is glued to its position using epoxy glue. Magnetic encoder PCB is also glued to its place. Remaining parts of the system are connected to each other using socket countersunk head cap screws and socket head cap screws. Exploded view of the system is given in Figure 4.5. Assembly procedure can be realized from this figure.

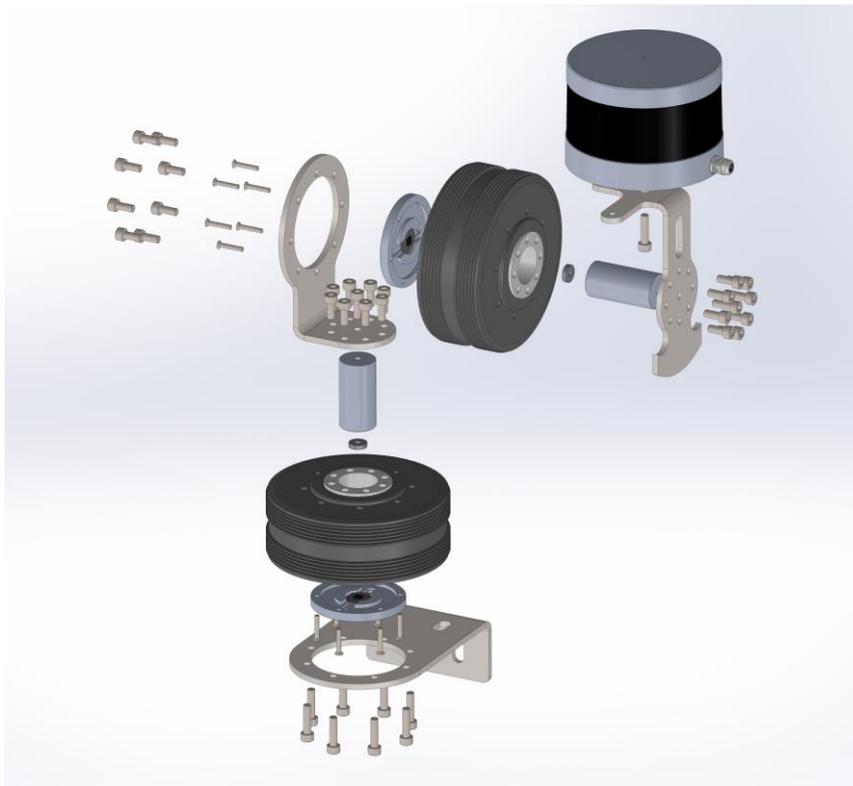


Figure 4.5 Exploded View

Figure 4.6 and Figure 4.7 show the pictures from the assembly procedure and the final assembled system respectively.



Figure 4.6 Partially Assembled System



Figure 4.7 Assembled System

4.2 Mathematical Modelling

This part contains the kinematic modeling of the system. In order to determine the position and orientation of the sensor coordinate system with respect to the base frame of reference, kinematic description of the system is needed. With the knowledge of pitch and yaw angles, this kinematic description allows us to accurately transform LiDAR point cloud data.

4.2.1 Kinematic Modelling

Denavit-Hartenberg convention is used to describe the kinematics of the designed gimbal system. This gimbal system is essentially a 2-DOF serial manipulator with 2 revolute joints. Base link (link 0) is where the system can be mounted on to the surface on which the system will operate. The first motor is rigidly connected to the base link and thus, it is a part of the base link. The link 1 is connected to the base link with a revolute joint (joint 1) which comes from the rotation of the first motor. Motor 2 is rigidly connected to the link 1. Link 2 is connected to the link 1 with a revolute joint (joint 2) which comes from the rotation of the second motor. The sensor is rigidly connected to the end of the link 2. Links on the manipulator is shown in Figure 4.8.

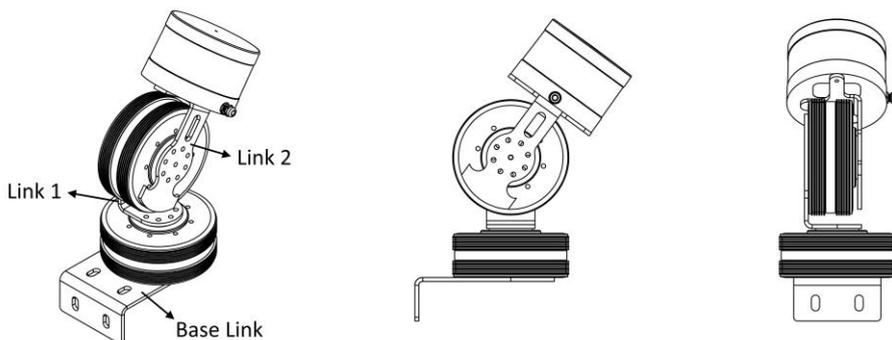


Figure 4.8 Manipulator Links

4.2.1.1 Kinematic Diagram

Kinematic diagram of the system which is formulated according to the Denavit-Hartenberg convention is given in Figure 4.9.

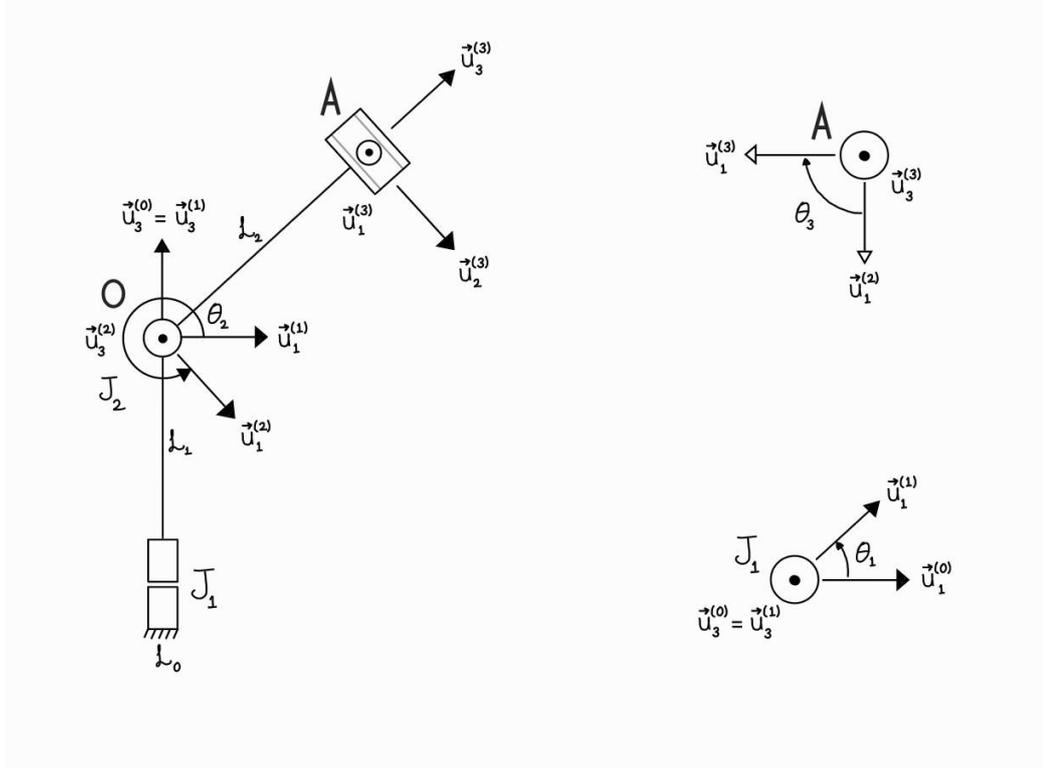
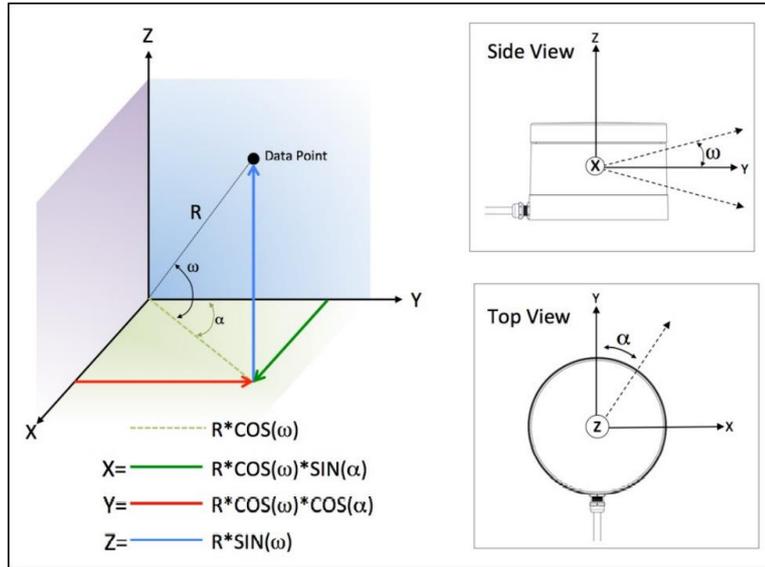


Figure 4.9 Kinematic Diagram

The designed system is a 2-DOF serial manipulator with two revolute joints. Joint axes of these two revolute joints intersect each other perpendicularly. Furthermore, z-axis of the VLP-16 Sensor Coordinate System intersects the joint axis of the second revolute joint perpendicularly.

In this configuration, the LiDAR sensor can be called the end-effector B_e and it is rigidly connected to the second link L_2 . It should be noted that the length of the second link terminates at the origin of the sensor coordinate system which is shown

in Figure 4.10. Velodyne Puck's sensor data origin is located 37.7 mm above the sensor base, on the z-axis as shown in that figure.



In the kinematic diagram, $\vec{u}_3^{(k)}$ are assigned to joint axes of each joint with arbitrarily selected direction. The unit vector $\vec{u}_1^{(k)}$ should be assigned to the common normal between the axes of the joints. However, since the axes of joints intersect, $\vec{u}_1^{(k)}$ are selected arbitrarily as shown in Figure 4.9. Then, $\vec{u}_2^{(k)}$ is selected to make the reference frame right-handed.

$$\vec{u}_2^{(k)} = \vec{u}_3^{(k)} \times \vec{u}_1^{(k)} \quad (4.1)$$

4.2.1.2 Denavit-Hartenberg Parameters

According to [33], Denavit-Hartenberg parameters can be explained as follows.

Joint (Rotation) Angle is between the axes of common normals N_{k-1} and N_k :

$$\theta_k = \sphericalangle \left[\vec{u}_1^{(k-1)} \rightarrow \vec{u}_1^{(k)} \right] \text{ about } \vec{u}_3^{(k)} \quad (4.2)$$

Twist Angle is between the axes of joints J_{k-1} and J_k :

$$\beta_k = \sphericalangle \left[\vec{u}_3^{(k-1)} \rightarrow \vec{u}_3^{(k)} \right] \text{ about } \vec{u}_1^{(k-1)} \quad (4.3)$$

Offset is between the common normals N_{k-1} and N_k :

$$s_k = B_k O_k \text{ along } \vec{u}_3^{(k)} \quad (4.4)$$

Effective Length of the link L_k between the joints J_k and J_{k+1} :

$$b_k = O_k B_{k+1} \text{ along } \vec{u}_1^{(k)} \quad (4.5)$$

4.2.1.3 Kinematic Description According to the D-H Convention

Denavit-Hartenberg parameters for the designed system are listed in this section. It should be noted that, since J_1 and J_2 are revolute joints, s_1 , s_2 , and s_3 are constants. Additionally, θ_3 is also a constant since Velodyne Puck is rigidly connected to the second link.

Joint (Rotation) Angles:

$$\theta_1 = \angle[\vec{u}_1^{(0)} \rightarrow \vec{u}_1^{(1)}] \text{ about } \vec{u}_3^{(1)} \quad (4.6)$$

$$\theta_2 = \angle[\vec{u}_1^{(1)} \rightarrow \vec{u}_1^{(2)}] \text{ about } \vec{u}_3^{(2)} \quad (4.7)$$

$$\theta_3 = \angle[\vec{u}_1^{(2)} \rightarrow \vec{u}_1^{(3)}] \text{ about } \vec{u}_3^{(3)} = -\pi/2 \quad (4.8)$$

Twist Angles:

$$\beta_1 = \angle[\vec{u}_3^{(0)} \rightarrow \vec{u}_3^{(1)}] \text{ about } \vec{u}_1^{(0)} = 0 \quad (4.9)$$

$$\beta_2 = \angle[\vec{u}_3^{(1)} \rightarrow \vec{u}_3^{(2)}] \text{ about } \vec{u}_1^{(1)} = \frac{\pi}{2} \quad (4.10)$$

$$\beta_3 = \angle[\vec{u}_3^{(2)} \rightarrow \vec{u}_3^{(3)}] \text{ about } \vec{u}_1^{(2)} = -\frac{\pi}{2} \quad (4.11)$$

Offsets:

$$s_1 = 0, \quad s_2 = 0, \quad s_3 = OA = 107.7 \text{ mm} \quad (4.12)$$

Effective Link Lengths:

$$b_1 = 0, \quad b_2 = 0, \quad b_3 = 0 \quad (4.13)$$

Link Frame Origins:

$$O_0 = O, \quad O_1 = O, \quad O_2 = O, \quad O_3 = A \quad (4.14)$$

O_3 is the origin of the VLP-16 Lidar Sensor Coordinate System.

4.2.1.4 Forward Kinematics in the Position Domain

Orientation Matrices of the Links with respect to the Base Frame

Link to link orientation matrices can be calculated according to the following formula:

$$\hat{C}^{(k-1,k)} = e^{\tilde{u}_1 \beta_k} e^{\tilde{u}_3 \theta_k} \quad (4.15)$$

Then,

$$\hat{C}^{(0,1)} = e^{\tilde{u}_1 \beta_1} e^{\tilde{u}_3 \theta_1} = e^{\tilde{u}_1 0} e^{\tilde{u}_3 \theta_1} = e^{\tilde{u}_3 \theta_1} \quad (4.16)$$

$$\hat{C}^{(1,2)} = e^{\tilde{u}_1 \beta_2} e^{\tilde{u}_3 \theta_2} = e^{\tilde{u}_1 \left(\frac{\pi}{2}\right)} e^{\tilde{u}_3 \theta_2} \quad (4.17)$$

$$\hat{C}^{(2,3)} = e^{\tilde{u}_1 \beta_3} e^{\tilde{u}_3 \theta_3} = e^{\tilde{u}_1 \left(-\frac{\pi}{2}\right)} e^{\tilde{u}_3 \left(-\frac{\pi}{2}\right)} \quad (4.18)$$

Orientation matrices with respect to the base frame can be calculated according to the following formula:

$$\hat{C}^{(0,k)} = \hat{C}^{(0,k-1)} \hat{C}^{(k-1,k)} \quad (4.19)$$

Then,

$$\hat{C}^{(0,1)} = e^{\tilde{u}_3 \theta_1} \quad (4.20)$$

$$\hat{C}^{(0,2)} = \hat{C}^{(0,1)} \hat{C}^{(1,2)} = e^{\tilde{u}_3 \theta_1} e^{\tilde{u}_1 \left(\frac{\pi}{2}\right)} e^{\tilde{u}_3 \theta_2} = e^{\tilde{u}_3 \theta_1} e^{-\tilde{u}_2 \theta_2} e^{\tilde{u}_1 \left(\frac{\pi}{2}\right)} \quad (4.21)$$

$$\begin{aligned}\hat{C}^{(0,3)} &= \hat{C}^{(0,1)}\hat{C}^{(1,2)}\hat{C}^{(2,3)} = e^{\tilde{u}_3\theta_1}e^{-\tilde{u}_2\theta_2}e^{\tilde{u}_1\left(\frac{\pi}{2}\right)}e^{\tilde{u}_1\left(-\frac{\pi}{2}\right)}e^{\tilde{u}_3\left(-\frac{\pi}{2}\right)} \\ &= e^{\tilde{u}_3\theta_1}e^{-\tilde{u}_2\theta_2}e^{\tilde{u}_3\left(-\frac{\pi}{2}\right)}\end{aligned}\quad (4.22)$$

Thus, the orientation of the sensor origin with respect to the base frame is

$$\hat{C}^{(0,3)} = e^{\tilde{u}_3\theta_1}e^{-\tilde{u}_2\theta_2}e^{\tilde{u}_3\left(-\frac{\pi}{2}\right)}\quad (4.23)$$

Relative Location of the Sensor Origin with respect to the Base Frame

Relative location of $F_k(O_k)$ with respect to $F_{k-1}(O_{k-1})$ can be found as:

$$\vec{r}_{k-1,k} = b_{k-1}\vec{u}_1^{(k-1)} + s_k\vec{u}_3^{(k)}\quad (4.24)$$

Then the location of the sensor origin with respect to the base frame can be represented as:

$$\vec{r} = \vec{r}_{OA} = b_0\vec{u}_1^{(0)} + s_1\vec{u}_3^{(1)} + b_1\vec{u}_1^{(1)} + s_2\vec{u}_3^{(2)} + b_2\vec{u}_1^{(2)} + s_3\vec{u}_3^{(3)}\quad (4.25)$$

$$\vec{r} = s_3\vec{u}_3^{(3)} \text{ where } s_3 = OA\quad (4.26)$$

Corresponding matrix equation becomes:

$$\begin{aligned}\bar{r} = \bar{r}^{(0)} &= s_3\bar{u}_3^{(3/0)} = s_3\hat{C}^{(0,3)}\bar{u}_3^{(3/3)} \\ &= s_3e^{\tilde{u}_3\theta_1}e^{-\tilde{u}_2\theta_2}e^{\tilde{u}_3\left(-\frac{\pi}{2}\right)}\bar{u}_3\end{aligned}\quad (4.27)$$

In the above equation \bar{u}_3 is the third basic column matrix where the three basic column matrices are:

$$\bar{u}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \bar{u}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \bar{u}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.28)$$

Equation (4.27) can be simplified as follows:

$$\bar{r} = s_3 e^{\tilde{u}_3 \theta_1} e^{-\tilde{u}_2 \theta_2} \left(e^{\tilde{u}_3 \left(-\frac{\pi}{2} \right)} \bar{u}_3 \right) = s_3 e^{\tilde{u}_3 \theta_1} e^{-\tilde{u}_2 \theta_2} \bar{u}_3 \quad (4.29)$$

$$\bar{r} = s_3 e^{\tilde{u}_3 \theta_1} \left(e^{-\tilde{u}_2 \theta_2} \bar{u}_3 \right) = s_3 e^{\tilde{u}_3 \theta_1} (\bar{u}_3 \cos(\theta_2) - \bar{u}_1 \sin(\theta_2)) \quad (4.30)$$

$$\bar{r} = s_3 e^{\tilde{u}_3 \theta_1} \bar{u}_3 \cos(\theta_2) - s_3 \left(e^{\tilde{u}_3 \theta_1} \bar{u}_1 \right) \sin(\theta_2) \quad (4.31)$$

$$\bar{r} = s_3 (\bar{u}_3) \cos(\theta_2) - s_3 (\bar{u}_1 \cos(\theta_1) + \bar{u}_2 \sin(\theta_1)) \sin(\theta_2) \quad (4.32)$$

$$\begin{aligned} \bar{r} &= \bar{u}_3 s_3 \cos(\theta_2) - \bar{u}_1 s_3 \cos(\theta_1) \sin(\theta_2) \\ &\quad - \bar{u}_2 s_3 \sin(\theta_1) \sin(\theta_2) \end{aligned} \quad (4.33)$$

Equation (4.33) indicate that

$$r_1 = -s_3 \sin(\theta_2) \cos(\theta_1) \quad (4.34)$$

$$r_2 = -s_3 \sin(\theta_2) \sin(\theta_1) \quad (4.35)$$

$$r_3 = s_3 \cos(\theta_2) \quad (4.36)$$

Where \bar{r} is,

$$\bar{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \quad (4.37)$$

4.2.1.5 Inverse Kinematics in the Position Domain

Relative location of the sensor origin with respect to the base frame was found as

$$\begin{aligned}\bar{r} = & \bar{u}_3 s_3 \cos(\theta_2) - \bar{u}_1 s_3 \cos(\theta_1) \sin(\theta_2) \\ & - \bar{u}_2 s_3 \sin(\theta_1) \sin(\theta_2)\end{aligned}\quad (4.38)$$

Upon pre-multiplication of the equation (4.38) with \bar{u}_3^t , we get the following

$$\begin{aligned}\bar{u}_3^t \bar{r} = & \bar{u}_3^t \bar{u}_3 s_3 \cos(\theta_2) - \bar{u}_3^t \bar{u}_1 s_3 \cos(\theta_1) \sin(\theta_2) \\ & - \bar{u}_3^t \bar{u}_2 s_3 \sin(\theta_1) \sin(\theta_2)\end{aligned}\quad (4.39)$$

$$\bar{u}_3^t \bar{r} = s_3 \cos(\theta_2)\quad (4.40)$$

$$r_3 = s_3 \cos(\theta_2) \rightarrow \cos(\theta_2) = \frac{r_3}{s_3}\quad (4.41)$$

From the equation (4.41) we get the solution

$$\theta_2 = 2n\pi + \cos^{-1}\left(\frac{r_3}{s_3}\right) \quad n \in \mathbb{Z}\quad (4.42)$$

Or,

$$\theta_2 = 2n\pi - \cos^{-1}\left(\frac{r_3}{s_3}\right) \quad n \in \mathbb{Z}\quad (4.43)$$

However, due to the physical limitations of the system, θ_2 is limited such that $\frac{3\pi}{2} \leq \theta_2 \leq \frac{5\pi}{2}$. In other words, in this situation $n = 1$. Therefore, the multiplicity in the equation can be eliminated and the equation can be written as

$$\theta_2 = 2\pi - \cos^{-1}\left(\frac{r_3}{s_3}\right) \quad (4.44)$$

Upon pre-multiplication of the equation (4.38) with \bar{u}_2^t , we get the following scalar equation

$$\begin{aligned} \bar{u}_2^t \bar{r} &= \bar{u}_2^t \bar{u}_3 s_3 \cos(\theta_2) - \bar{u}_2^t \bar{u}_1 s_3 \cos(\theta_1) \sin(\theta_2) \\ &\quad - \bar{u}_2^t \bar{u}_2 s_3 \sin(\theta_1) \sin(\theta_2) \end{aligned} \quad (4.45)$$

$$r_2 = -s_3 \sin(\theta_1) \sin(\theta_2) \quad (4.46)$$

Noting the fact that the θ_2 is found before, θ_1 can be found as

$$r_2 = -s_3 \sin(\theta_1) \sin(\theta_2) \rightarrow \sin(\theta_1) = \frac{-r_2}{s_3 \sin(\theta_2)} \quad (4.47)$$

$$\theta_1 = \sin^{-1}\left(\frac{-r_2}{s_3 \sin(\theta_2)}\right) + 2n\pi \quad n \in \mathbb{Z} \quad (4.48)$$

In the designed system, θ_1 is also physically limited such that $0 \leq \theta_1 \leq 2\pi$. Therefore, the multiplicity in the solution is eliminated and the angle θ_1 can be determined as

$$\theta_1 = \sin^{-1}\left(\frac{-r_2}{s_3 \sin(\theta_2)}\right) \quad (4.49)$$

Furthermore, upon pre-multiplication of the equation (4.38) with \bar{u}_1^t , we get the following.

$$\begin{aligned} \bar{u}_1^t \bar{r} &= \bar{u}_1^t \bar{u}_3 s_3 \cos(\theta_2) - \bar{u}_1^t \bar{u}_1 s_3 \cos(\theta_1) \sin(\theta_2) \\ &\quad - \bar{u}_1^t \bar{u}_2 s_3 \sin(\theta_1) \sin(\theta_2) \end{aligned} \quad (4.50)$$

$$r_1 = -s_3 \cos(\theta_1) \sin(\theta_2) \quad (4.51)$$

However, since the degree of freedom of the system is 2, this equation is redundant, but it still can be used to check the accuracy of the solution since θ_1 and θ_2 must satisfy this equation.

4.3 BaseCamBGC Pro Controller Setup

In this part of the chapter, setup procedures followed to make BaseCamBGC Pro controller work correctly has been presented. First, general setup procedure is summarized, then the solution developed to be able to communicate with the controller is introduced. Finally, PID tuning procedure is explained.

4.3.1 General Setup

Before configuring the BaseCamBGC Pro controller, necessary connections between the components of the system need to be made. The system architecture of the setup is provided in Figure 4.11, in which blue lines represent data flow while red lines represent power distribution. Computer and the DC Power Supply feeds from AC mains electricity which is not shown on the diagram. Other components require 12 VDC to operate. Data transfer method for each connection is given on the diagram. The connection between Velodyne Puck and its interface box is a special connection which is not allowed to be manipulated by the user. This connection carries sensor readings in the form of UDP packets, provides connectivity for the GPS data, and carries the power that the sensor needs.

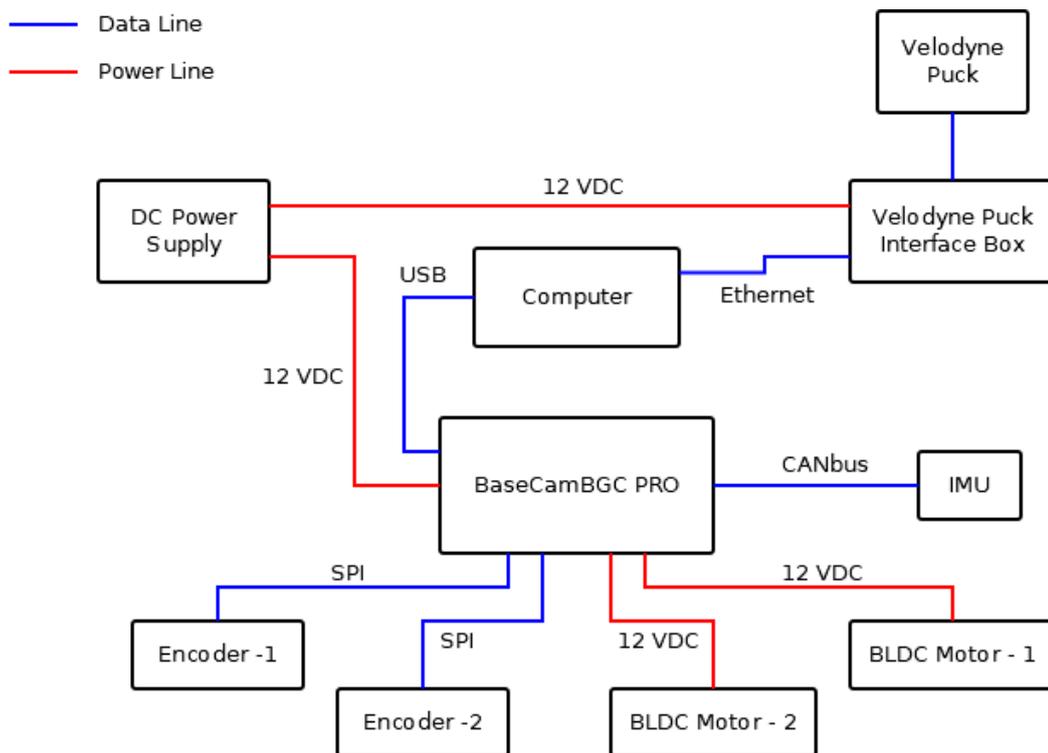


Figure 4.11 System Architecture

After the realization of the necessary connections, BaseCamBGC Pro controller can be configured to work with the designed system. For this process, the graphical user interface of the controller is utilized. Main steps taken to setup the controller are listed below.

- Configure IMU orientation
- Calibrate gyroscope
- Calibrate accelerometer
- Select motor outputs and disable unused axis
- Set motor power
- Set the number of poles of the motors
- Setup encoders

4.3.2 Serial Communication

A python package named “pysbgc” has been developed to perform the required communication between the BaseCamBGC Pro and the computer. This python package uses the message format defined in protocol version 1 of the SimpleBGC Serial Protocol Specification [34] documentation. The package first connects to the BaseCamBGC Pro controller board through serial port. Then, it prepares the necessary hex data for the chosen command and sends this message to the controller. If an answer is expected from the controller, the package listens to the same serial port and unpacks the serial message. This message, depending on the command type, may include acceleration data, gyro data, IMU data, etc. The full list of available commands and their content can be found in the “commands.py” script in the package. The following commands from [34] are available in the developed python package:

- `CMD_READ_PARAMS_3`
- `CMD_REALTIME_DATA_3`
- `CMD_GET_ANGLES`
- `CMD_CONTROL`
- `CMD_MOTORS_ON`
- `CMD_MOTORS_OFF`
- `CMD_REALTIME_DATA_CUSTOM`

This package allows the plotting of the selected incoming data by utilizing the “animate.py” script. This python file of the package includes a function “pysbgc.animate.plot()” accepts three variables; device, command, and the data to plot. The function sends the command to the device and plots the selected data from the response in real-time. A sample plot obtained using this function is given below. It shows the change of pitch angle information from the IMU on the system in time. This plot was obtained when motors were on, and the sensor was stabilized in perpendicular position. Then the sensor was pushed out of position and allowed to move back to the stable equilibrium several times.

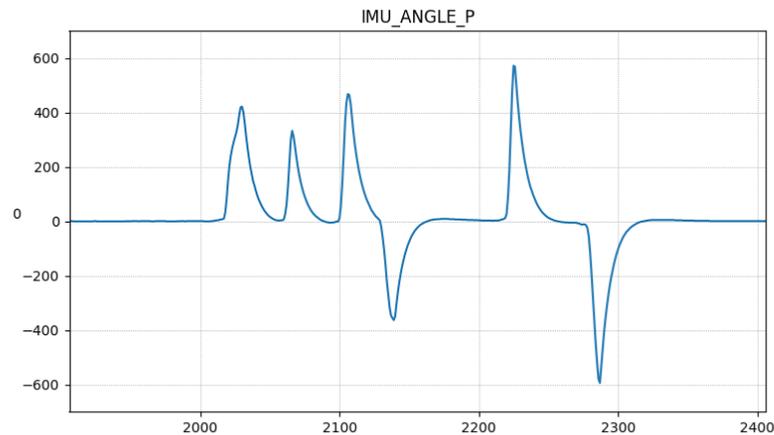


Figure 4.12 Plotting Functionality of the Developed Python Package

4.3.3 PID Tuning

The controller board BaseCamBGC Pro provides separate PID controllers for each of the roll, pitch and yaw axes, gains of which can be manipulated from the GUI and using serial communication. In the current setup roll axis is not used hence disabled from the GUI. Most efficient way to tune PID parameters is to use the auto-tuning functionality of the controller board. According to the documentation of the BaseCamBGC Pro [35], it uses powerful algorithms from control theory to auto-tune PID parameters and it is the recommended way to get the maximum performance from the designed system. PID auto-tuner allows the configuration of the target for tuning with a slider. This slider can be used to adjust if a better stability of better precision is required. If better precision is preferred, auto-tuner will aim for maximum gains while for better stability, moderate gains with emphasis on stability will be targeted.

It is important to have the system correctly configured for the auto-tune feature to work reliably. These configurations include IMU position configuration, accelerometer and gyroscope calibration, motor poles and outputs settings. After making sure that the system is accurately configured, PID auto-tune has been

performed with initial values of $P=1$, $I=0$ and $D=0$ for each axis as shown in Figure 4.13 and acquired PID parameters are given in Table 4.5

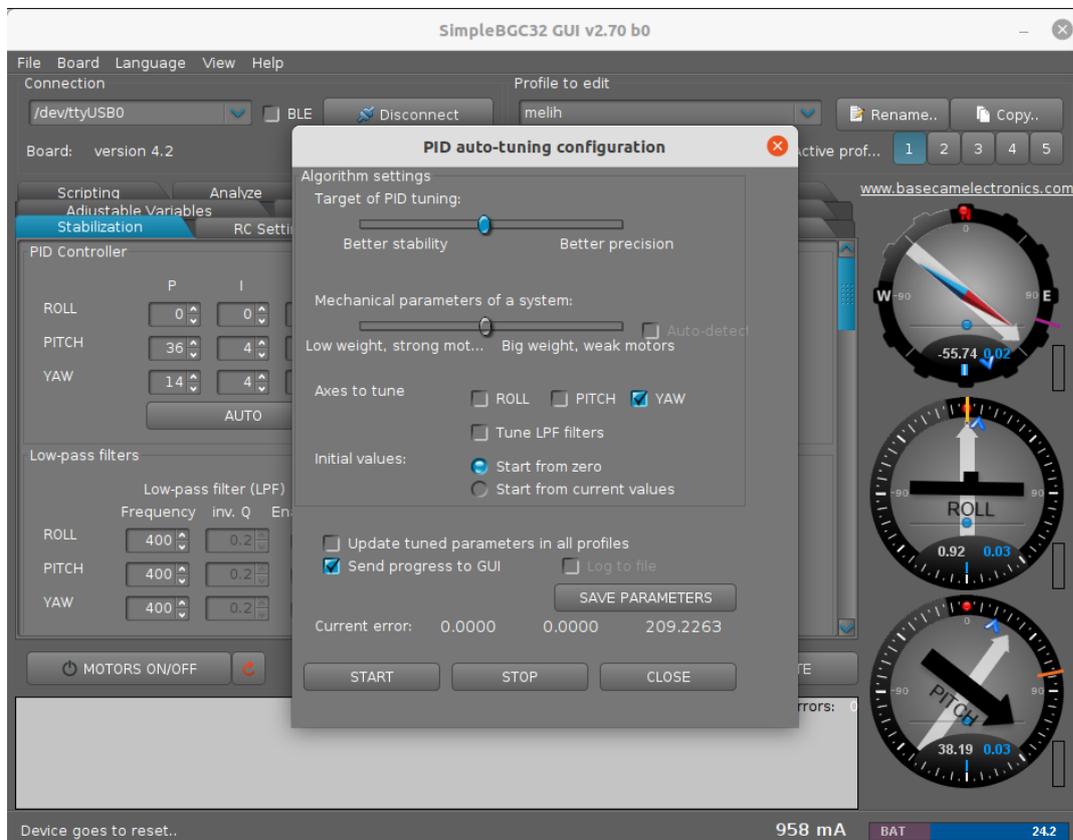


Figure 4.13 PID auto-tuning

Table 4.5 PID Gains

	P	I	D
Pitch	36	4	15
Yaw	14	4	6

Furthermore, the board provides a system analysis tool that can be used obtain information about the system's response to a certain input and displays this data in the form of a Bode Plot which shows amplitude versus frequency and phase response

versus frequency. This frequency response data is also stored in a .csv file which can be used with MATLAB to further investigate the system. The Bode Plot displayed can be used to comment on the stability of the system and determine which areas in the frequency domain are problematic for the system.

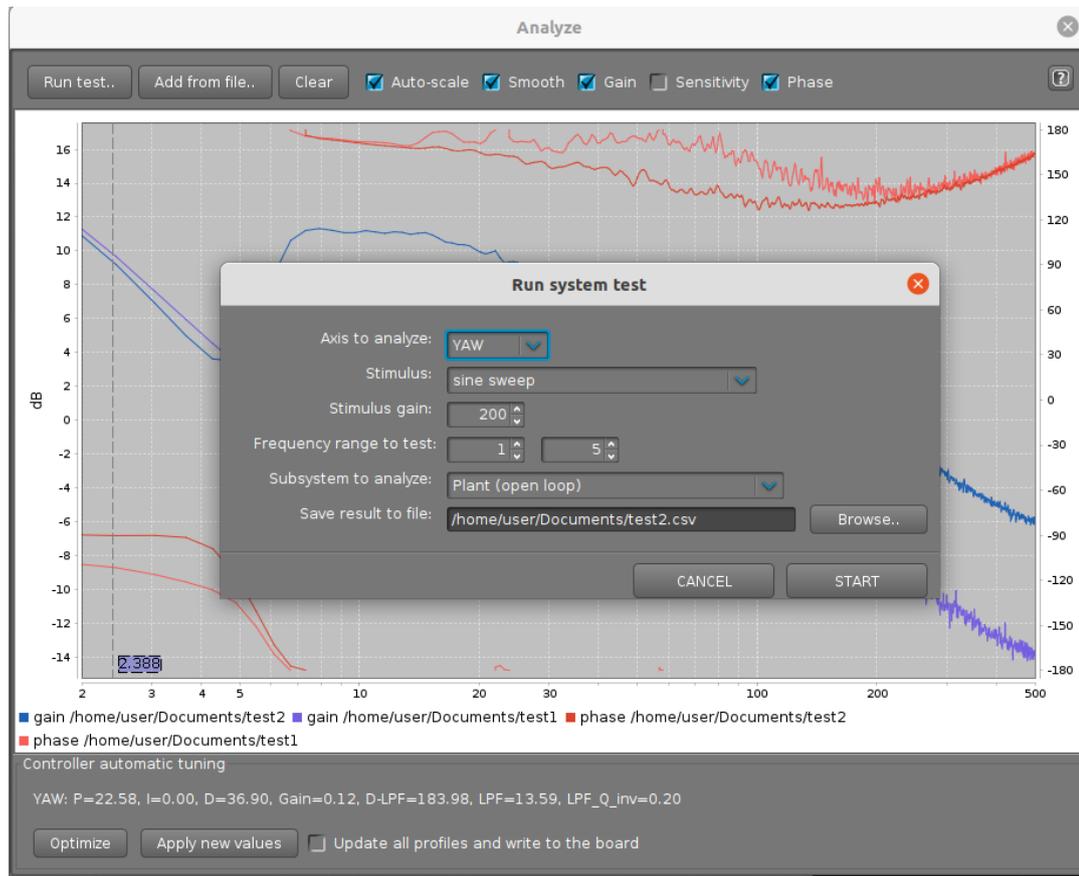


Figure 4.14 System Analysis Tool

System analysis tool has been run with the following configurations.

- Sine sweep stimulus
- 200 stimulus gain
- 1 to 5 frequency range
- Open Loop – Plant

The obtained results are presented in figures Figure 4.15 and Figure 4.16.

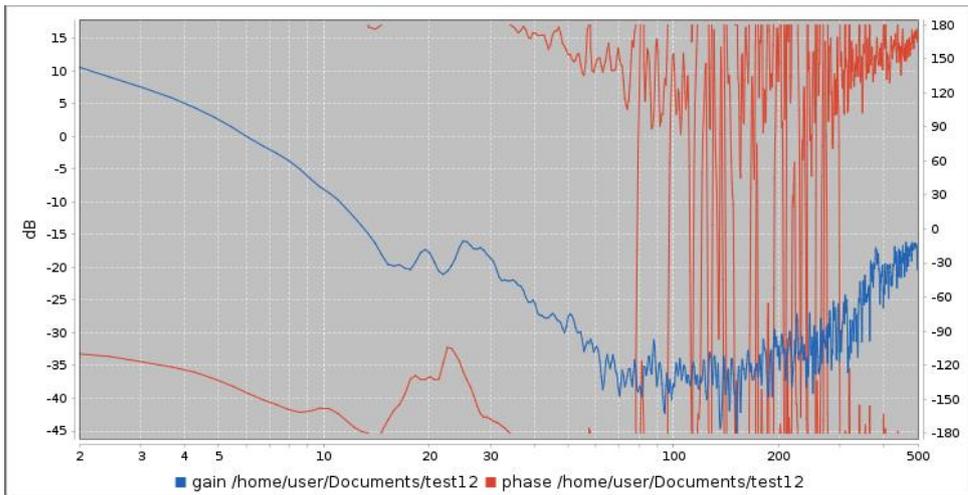


Figure 4.15 System Analysis – Pitch



Figure 4.16 System Analysis – Yaw

The performance of the PID controller with auto-tuned gains can be visualized by comparing the target position and actual position of the system. In the following figures, horizontal axis represents time while the vertical axis represents the angular position measurement. Each column in those figures correspond to 1 second and the angular measurement values are given on the vertical axis as a raw data. In Figure 4.17, the system moves to 0° from the rest position. Green line shows the target pitch angle while purple line shows the actual angular pitch position of the system from

IMU measurements. The error is minimal; thus, these two lines follow each other very closely.

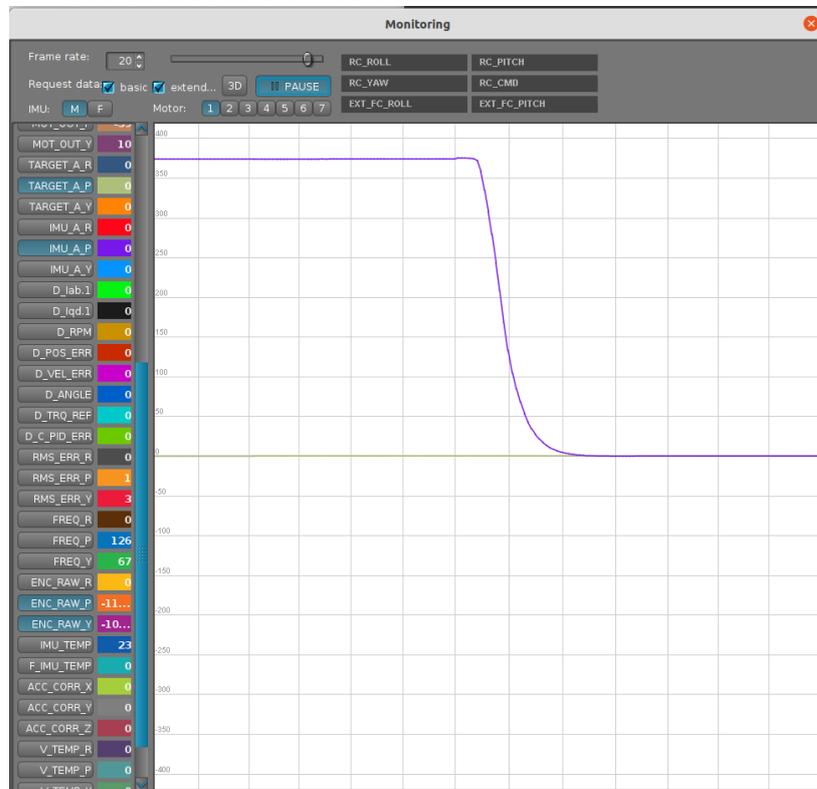


Figure 4.17 Rest to Center

In Figure 4.18 and Figure 4.19 slow and fast sinusoidal motion was given to the system and the system behavior was monitored. Similarly, the system closely follows the target position. In fast sinusoidal motion plot, a close look at the edge of the plot, reveals the slight error.

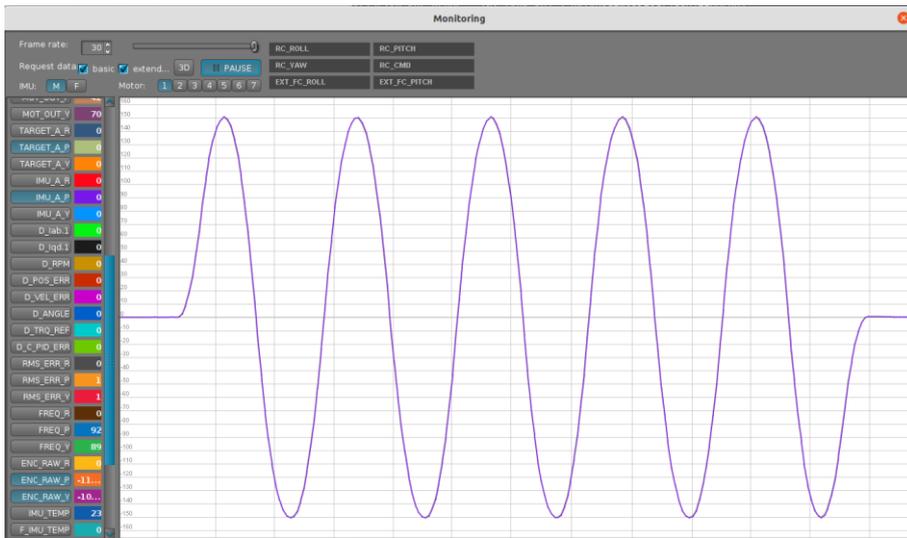


Figure 4.18 Slow Sinusoidal Motion

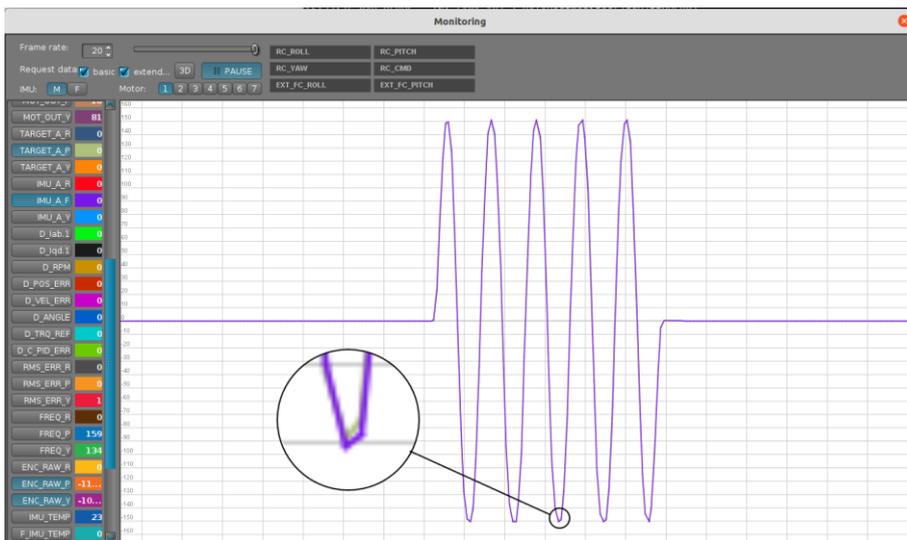


Figure 4.19 Fast Sinusoidal Motion

In another scenario, while the system was performing a sinusoidal motion around the pitch axis, a disturbance was applied by manually trying to prevent the motion of the system by hand. Figure 4.20 shows this scenario. Again, the green line represents the target angular position, and the purple line represents the actual angular position.

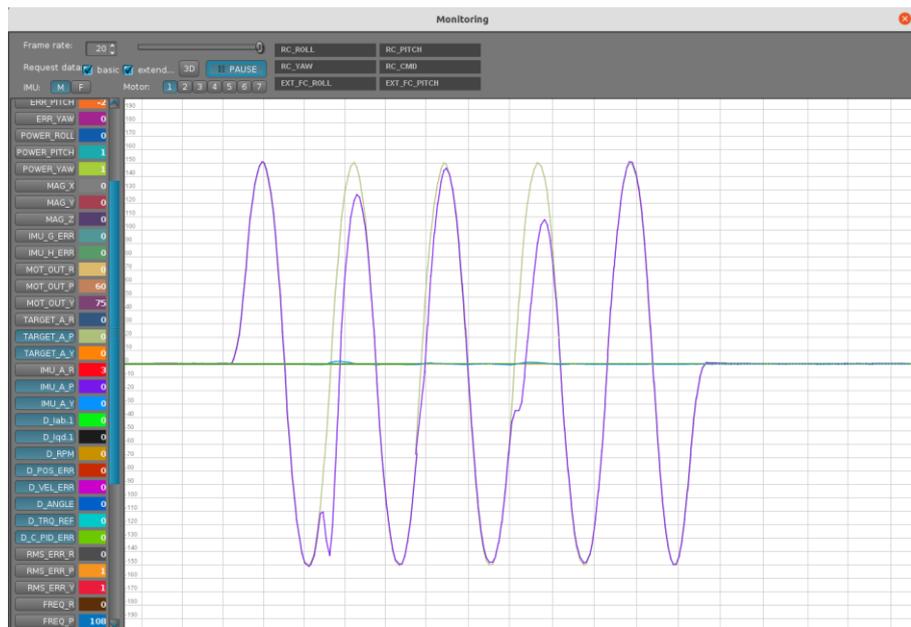


Figure 4.20 Sinusoidal Motion with Disturbance

4.4 ROS Workspace

In this thesis study, ROS has been used as the framework to develop necessary software for the designed robotic system. In this part of the chapter, ROS workspace, developed packages, and other available ROS packages used in the workspace are explained.

4.4.1 Developed Packages

pysbgc

This ROS package has been developed for the purpose of communicating with the BaseCamBGC Pro controller board. It was based on the developed python package with the same name which was explained in Section 4.3.2.

This ROS package has a single ROS node called *basecam_node*. The node first connects to the controller board using serial port with the defined parameters. A publisher with a timer period of 8ms, which is the maximum rate the BaseCamBGC Pro can process incoming messages, reads yaw, pitch, and roll angles, and then publishes this information on a topic called */imu_angles*. This angle information is published as a raw data in 14-bit resolution.

Additionally. Since the connection to the controller board is made in this package, the node also sends the pre-defined motion control commands of the gimbal system to the controller board.

pysbgc_interfaces

This developed ROS package contains message type required by the *basecam_node* to publish messages on the */imu_angles* topic.

gimbal_tf2_broadcaster

This package has been developed to calculate the position and the orientation of the sensor origin with respect to the base frame of reference. This package contains a single ROS node called *lidar_tf2*. This node subscribes to the */imu_angles* topic to listen to yaw, pitch, and roll angles of the system. These angles are then converted to radians from 14-bit resolution angular measurements. Calculated yaw and pitch angles correspond to θ_1 and θ_2 from the kinematic diagram of the designed system. Using the knowledge of θ_1 and θ_2 , equations found in the description of the system's forward kinematics in the position domain are used to locate the sensor's position with respect to the base frame. In other words, transformation between the *base* frame and the *velodyne* frame is now available to us. This information is used to construct the *geometry_msgs/TransformStamped* ROS message to be published on the */tf* topic. In order to broadcast this transform information, *tf2* library [36] of ROS is used. *tf2* is a great transform library from ROS that makes it easier for the users of

ROS to keep track of multiple coordinate frames available in a robotic system whether the transform between them is static or dynamic. This message published on the */tf* topic allows the user to accurately transform point cloud data obtained from the sensor in the *velodyne* frame to the base frame. Thus, the point cloud map of the environment can accurately be built irrespective of the orientation and the location of the sensor with respect to the base frame.

webots_gimbal

In order to simulate the designed robotic system in the Webots robotic simulation environment using the developed simulation model, in addition to using the Webots' own controller interface, a ROS interface can be constructed. This Webots – ROS interface allows us to test the developed ROS software on the simulated robot and gather sensor data from the simulated sensors, published on different ROS topics. For this purpose, a ROS package called *webots_gimbal* has been developed. This package contains a launch file that starts the Webots and the controller node, a simple urdf file that describes the sensors on the robot, the controller written for the simulator and the description of the simulation environment written in Webots' *wbt* file format. This package successfully controls the two motors on the system and provides sensor readings from the simulation similar to the real system.

4.4.2 Available Packages

In addition to using the developed ROS packages, two more readily available collections of ROS packages have been used in this study. These ROS package collections are briefly explained in this section.

velodyne [37]

velodyne is a collection of ROS packages that supports various Velodyne LiDAR sensors including the Velodyne Puck. These packages have been used to get sensor data in the form of ROS topics from the sensor. Used packages from *velodyne* collection are *velodyne*, *velodyne_driver* and *velodyne_pointcloud*. Additionally, *velodyne_driver* depends on the *velodyne_msgs* package. The *velodyne* package contains the main ROS launch file that was modified and used. The *velodyne_driver* package is the driver for the sensor, and it takes the raw data from the sensor and combines it into a ROS message to be published once per revolution of the sensor. For example, when 1200 rpm rotation is selected for the sensor, these ROS messages are published at 20 Hz. The *velodyne_convert_node* from the *velodyne_pointcloud* package subscribes to the raw messages from the *velodyne_driver* package's *velodyne_driver_node* and converts them into point cloud data. This point cloud data is then published by the *velodyne_convert_node* on the topic */velodyne_points*. The message format of for this topic is *sensor_msgs/PointCloud2*, which is a common point cloud data format in ROS.

webots_ros2 [38]

For the developed *webots_gimbal* package to work with Webots, a Webots – ROS interface is needed. *webots_ros2* is a collection of ROS packages developed by the developers of Webots for this purpose. It includes several packages most of which are demos which are designed to make it easier for developers to build their own system. The main package needed for the application in this thesis is the *webots_ros2_driver* package from the collection. This package serves three purposes. These are to automatically create ROS interface for some of the sensors from a Webots robot model, to allow users to configure the ROS interface using the urdf file, and to extend the ROS interface with a plugin mechanism to be able to control the robot in the simulation.

4.4.3 ROS Node Graph

RQt is a GUI framework from ROS that provides many plugins that can be used to visualize and inspect the current ROS system. Node graph, one of these plugins, can be used to visualize ROS computation graph, which can be summarized as the network of ROS processes. Node graph of the system developed in this thesis is given in Figure 4.21. In this figure, boxes represent topics, ellipses represent nodes, and the arrows show the flow of data. Explanations made in the chapters 4.4.1 and 4.4.2 are visualized in this figure. Additionally, `/parameter_events` and `/diagnostics` topics come from the ROS itself. On the other hand, `/tf_static` topic, which has no connection to other nodes, is not used. This could happen when there is a subscriber to a topic but there is no message on that topic.

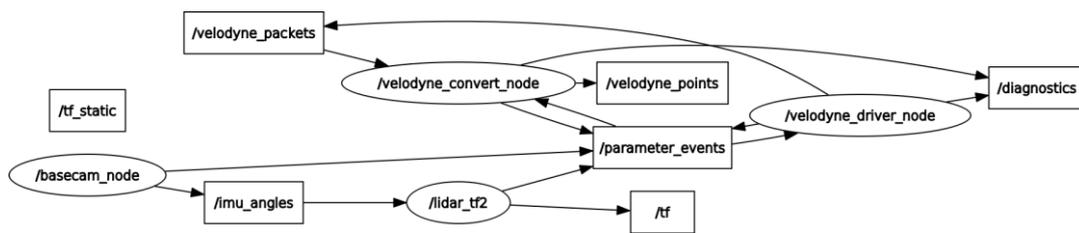


Figure 4.21 ROS Node Graph

4.5 Robotic Simulation

The designed 2-axis gimbal system is also modeled in Webots robotic simulation environment. This simulation model was created using the parameters of the real system. This way an almost identical twin of the real system is obtained in a simulation environment. This simulation model of the robotic manipulator in Webots environment can be seen in figure below.

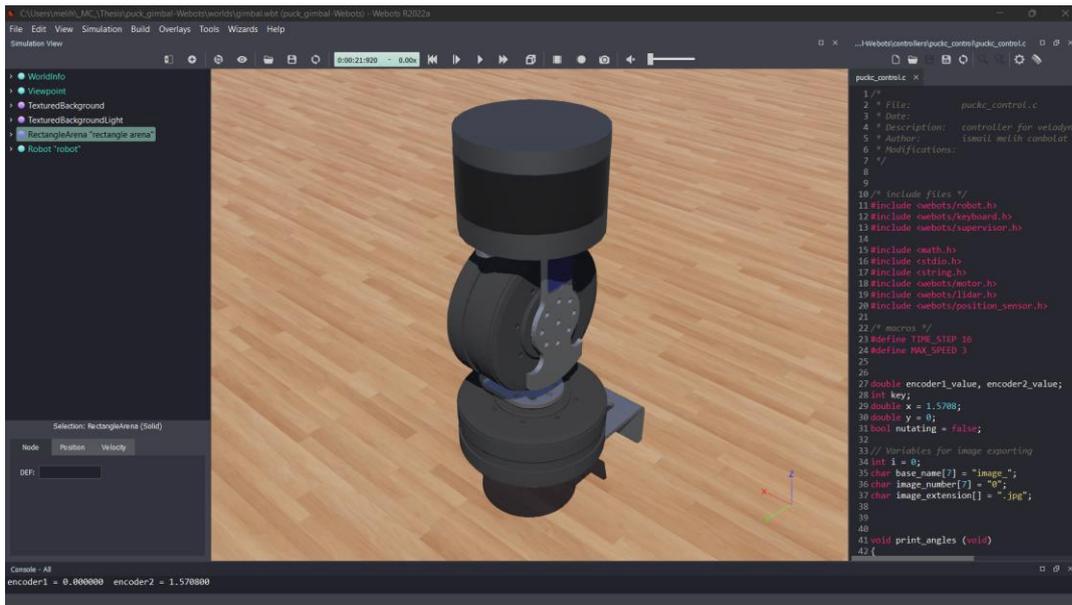


Figure 4.22 2-Axis Gimbal Webots Model

And the following figure shows the exploded view of the robotic simulation model of the system in Webots.

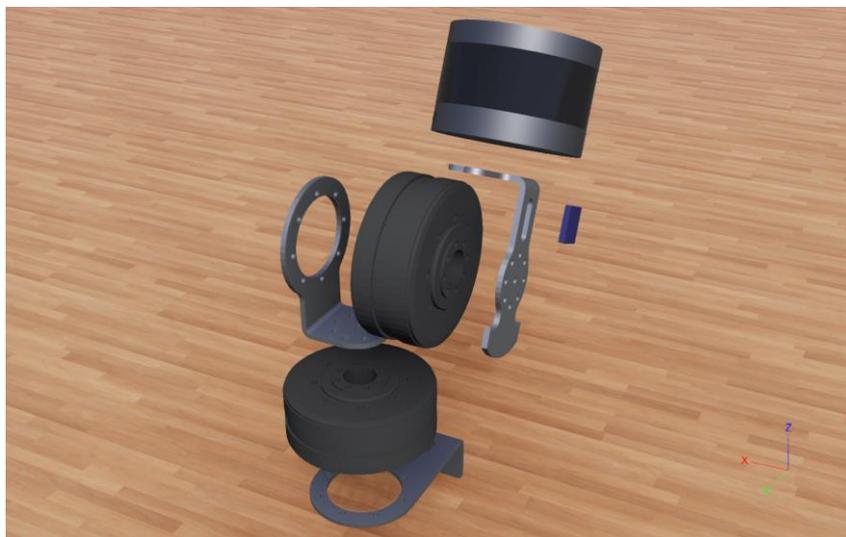


Figure 4.23 Webots Exploded View

This simulation model allows rapid testing of the system. Before starting the manufacturing process, this simulation environment was used to test the motion of

the system and to observe the effects of these motions to LiDAR point cloud data distribution in the simulation environment. Figure 4.24 shows sample point cloud data obtained by the sensor in two different orientations.

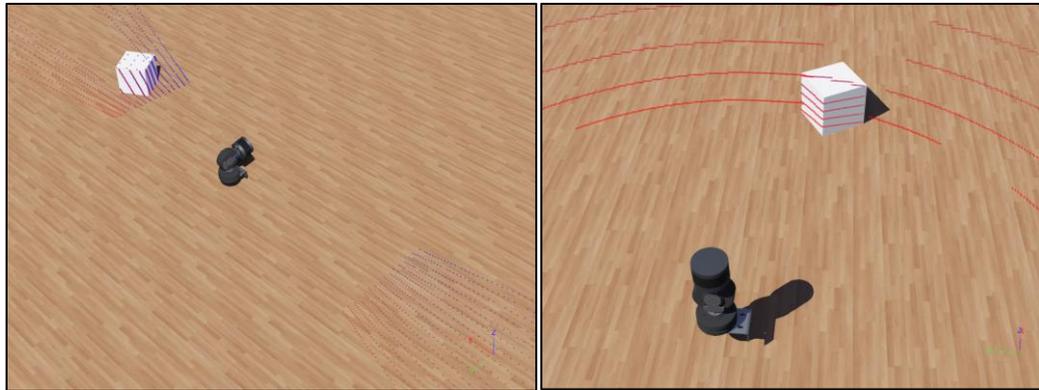


Figure 4.24 Webots Point Cloud

Even after the manufacturing and assembly of the system, use of this simulation environment makes it much easier to develop algorithms and test these algorithms in different scenarios. The simulation environment can be manipulated as desired, and the algorithms can easily be tested in various environments. Additionally, this simulation model of the system can be used in other simulations too. For example, this gimbal system could be mounted on a robotic platform and can be used with it. One such scenario is shown as an example in Figure 4.25 in which the gimbal system is mounted on a Pioneer 3-AT robot model from Webots library.



Figure 4.25 Gimbal System with Pioneer 3-AT

Representation of the designed system's robot model tree in Webots simulation environment can be summarized as follows:

Robot

Base (Solid)

Motor 1 (Solid)

Transform

HingeJoint

Joint Parameters

Device

Rotational Motor

Position Sensor

endpoint

Shape (Link 1)

Transform

Motor2 (Solid)

Transform

HingeJoint

Joint Parameters

Device

Rotational Motor

Position Sensor

endpoint

Shape (Link 2)

Transform

Shape (IMU)

Accelerometer

Gyro

Transform

Velodyne Puck (PROTO)

This robot model of the designed system is also made available as a PROTO definition. PROTO mechanism of Webots allows users to easily reuse complex systems by defining these systems according to PROTO file structure of Webots. PROTO definitions can be imported to any simulation world defined in Webots, and the changes made in the PROTO definition of the system would be reflected to other simulations, which use this PROTO, without any effort.

Controller

A controller software to easily control the simulated system is written in C. This controller runs in Webots, and it was prepared according to the requirements of the simulation, using the node API functions of Webots. Basic functionality of this controller can be listed as,

- Enable and drive motors
- Enable and read encoders
- Enable LiDAR sensor
- Read keyboard input
- Move the sensor in a desired way

In addition to providing numerical values for sensor readings, Webots can visualize these readings in a user-friendly interface. Figure 4.26 shows the plotted encoder readings when a sinusoidal motion was given to each motor. This feature helps in the development procedure. In order to obtain a better scan pattern, a 90° offset between each motor's motion was introduced. This phase difference can clearly be seen in that figure. This motion algorithm was then transferred to the real system to have the same motion, namely nutation, and thus the same scan pattern was obtained in the real system.

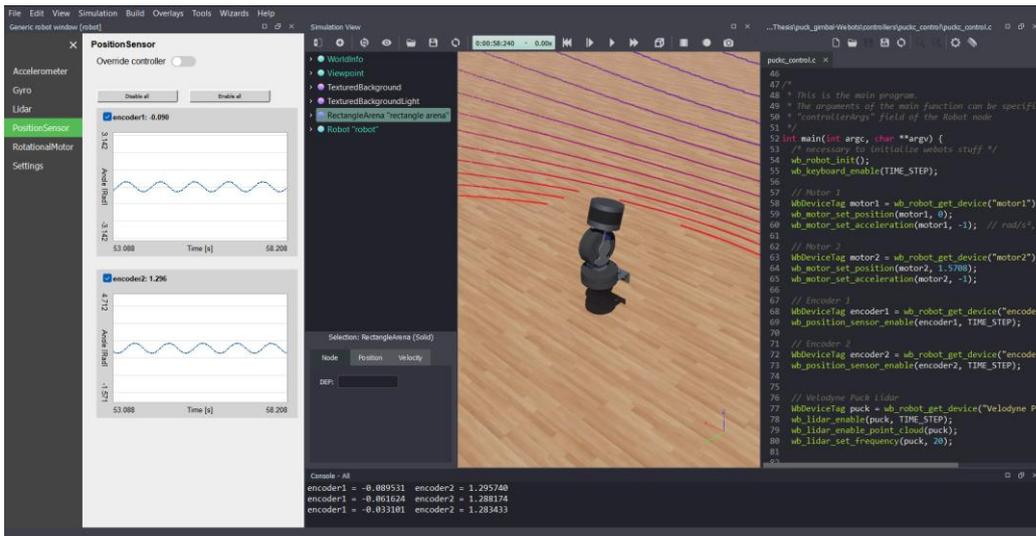


Figure 4.26 Webots Encoder Plots

CHAPTER 5

EXPERIMENTATION AND RESULTS

After the successful completion of the design procedures and software setup, the designed system has been tested to evaluate the mapping performance obtained by the system. In Figure 5.1, the test setup can be seen. The system is mounted on a heavy metal plate which creates a rigid base for the test setup.



Figure 5.1 Test Setup

Numbered items in Figure 5.1 are as follows: 1 - Base Metal Plate; 2 - Designed System; 3 - BaseCamBGC Pro; 4 - Velodyne Puck Interface Box; 5 - Power Supply.

5.1 Mapping an Indoor Environment

In the first qualitative experiment performed by using the designed system, an indoor environment is scanned, and its point cloud map is obtained. In all the figures that illustrate point cloud data, different colors represent the intensity of the returned laser beam for each point. This information may give an idea about the reflectivity of the surface, but it may also depend on the angle incidence. If the incidence angle for a laser beam is too large, the returned laser beam may be weak even for a surface with high reflectivity.

In the Figure 5.2 point cloud map of the environment captured by the sensor in stationary position as shown in Figure 5.1 can be seen. In this figure, point cloud map obtained is from a single rotation of the lasers in the sensor. Therefore, the gap between the points in the horizontal direction in addition to the large gap between the scan layers leads to a poor map.

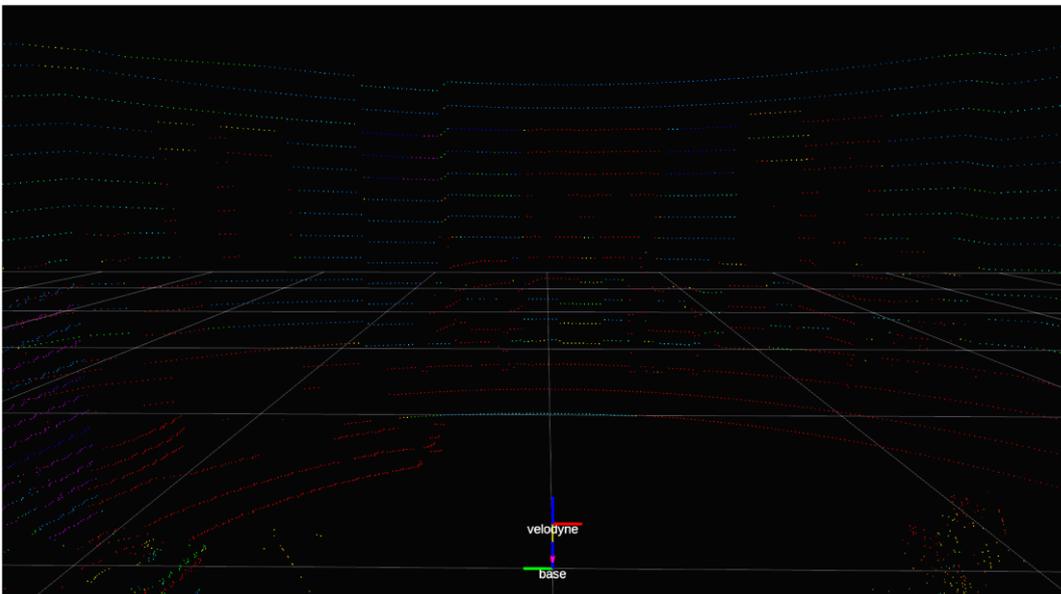


Figure 5.2 Stationary Instant Point Cloud Map

In the Figure 5.3, the sensor is kept stationary as before, however the point cloud map is obtained by accumulating the point cloud data for 1 second. In this scenario, the empty space between the points in each layer in the horizontal direction are mostly filled. This is possible due to the fact that the sensors' internal rotational speed varies slightly, which is an intentional feature of Velodyne Puck. However, still there is an empty space between the scan layers which greatly reduces the reliability of the obtained map.

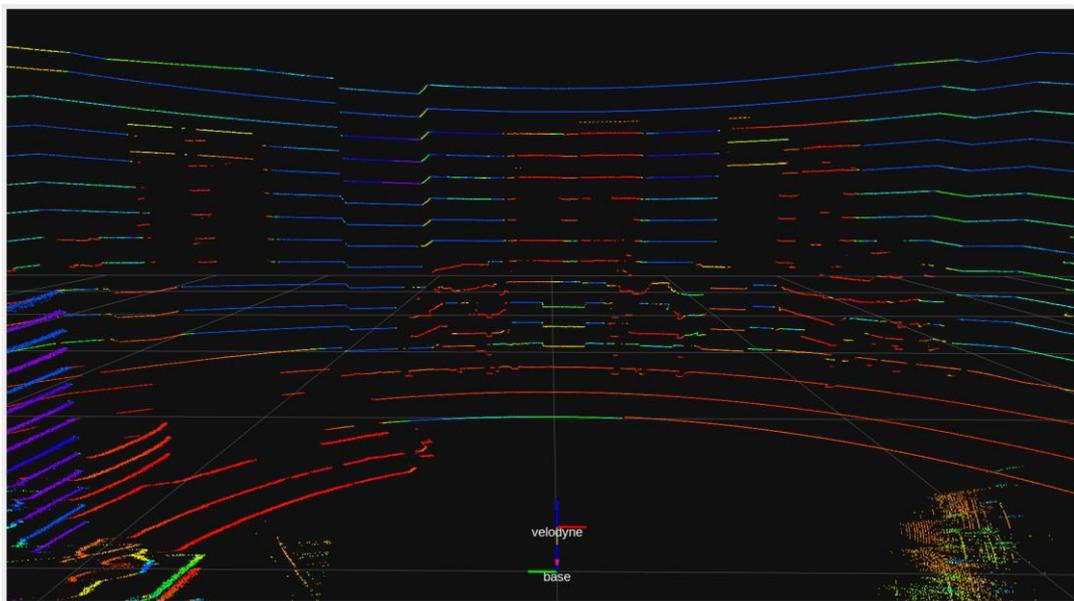


Figure 5.3 Stationary Accumulated Point Cloud Map

In the last scenario, the designed system was utilized, and the sensor was given a sinusoidal rotational motion around pitch and yaw axes. The rotational range is from -25° to $+25^\circ$ from the stationary position for each axis and the period of rotations was approximately 3.3 seconds. Point cloud map of the same environment with accumulation of the point cloud for 3 seconds is given in Figure 5.4.

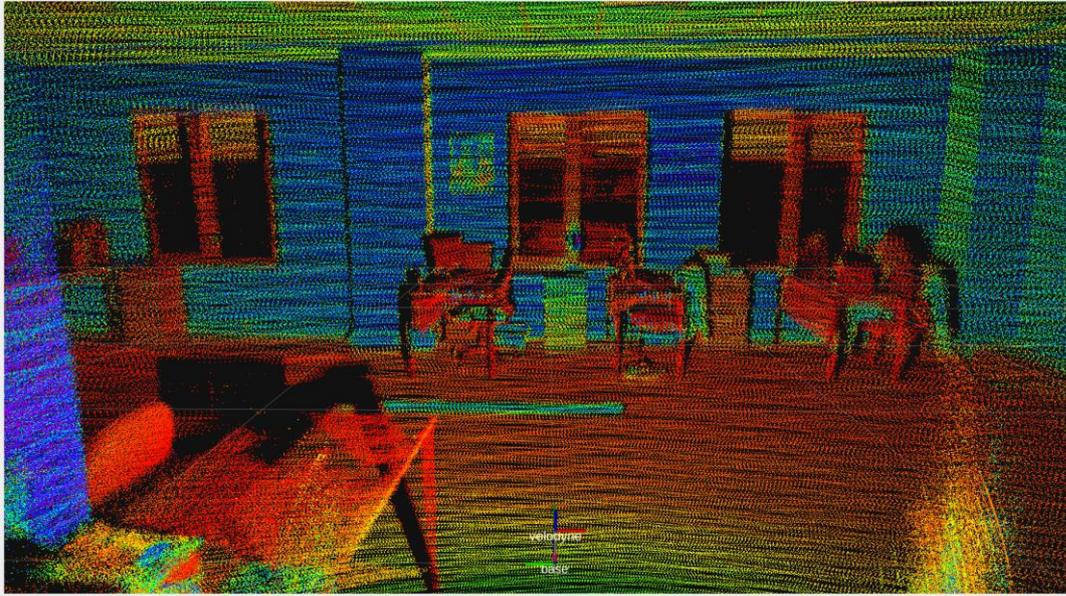


Figure 5.4 Accumulated Point Cloud Map Using the Designed System

The designed system allows us to scan the gaps between the layers and obtain a high density, homogenously distributed point cloud data. Additionally, the vertical field of view of the sensor is also increased. The system also allows us to change the vertical field of view by changing the motion limits of the system on the fly.

Contribution of an additional axis of rotation to the quality of the point cloud can be visualized in Figure 5.5 in which only the pitching motion was given to the same sensor. It can be seen from the figure that, there are empty spots in the point cloud pattern obtained by this motion. Additionally, unintentionally we get high density point cloud in some regions. This result is in compliance with the previous research [26] explained in the literature survey part of this thesis.

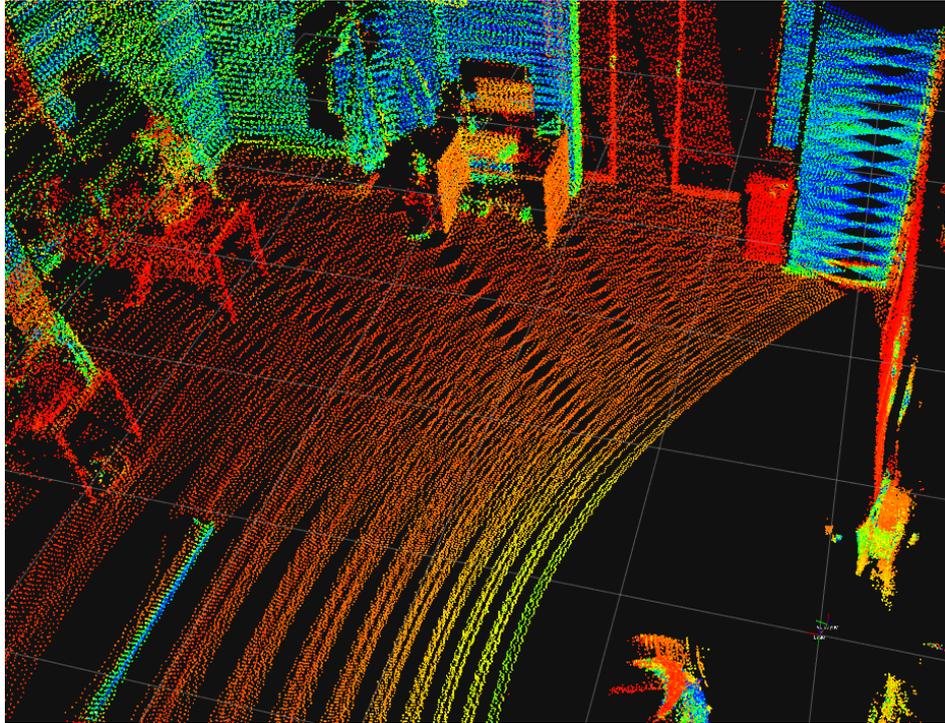


Figure 5.5 Lidar Pitching Only

An additional representation of the performance of the system may be done by providing the point cloud map of the whole building which was obtained using the same method. In Figure 5.6, the LiDAR sensor was kept stationary, and the point cloud data was accumulated for 5 seconds.

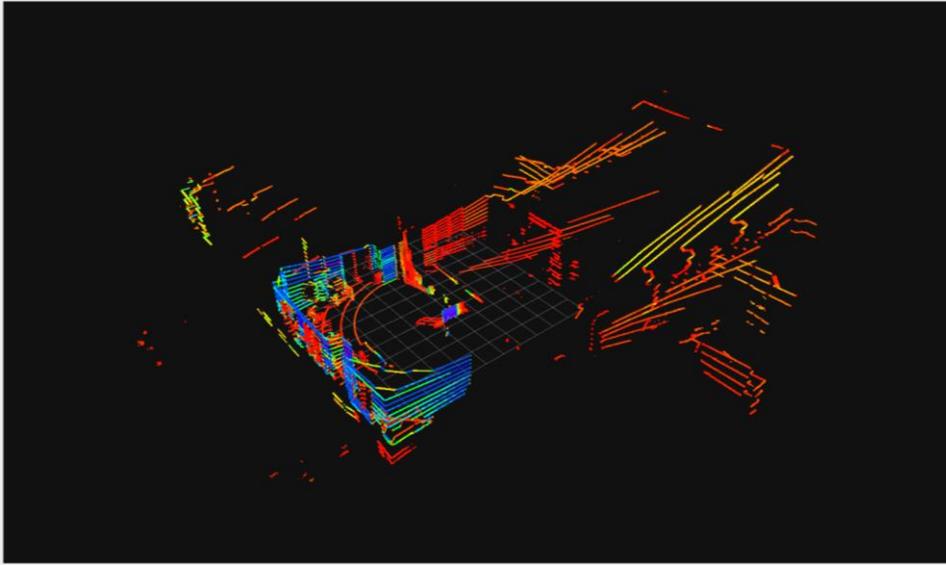


Figure 5.6 Point Cloud Map of the Building Using a Stationary LiDAR

In the Figure 5.7, the designed system was utilized to obtain the point cloud of the building by rotating the sensor in a previously described manner. In this scenario, point cloud data was accumulated for 5 seconds.

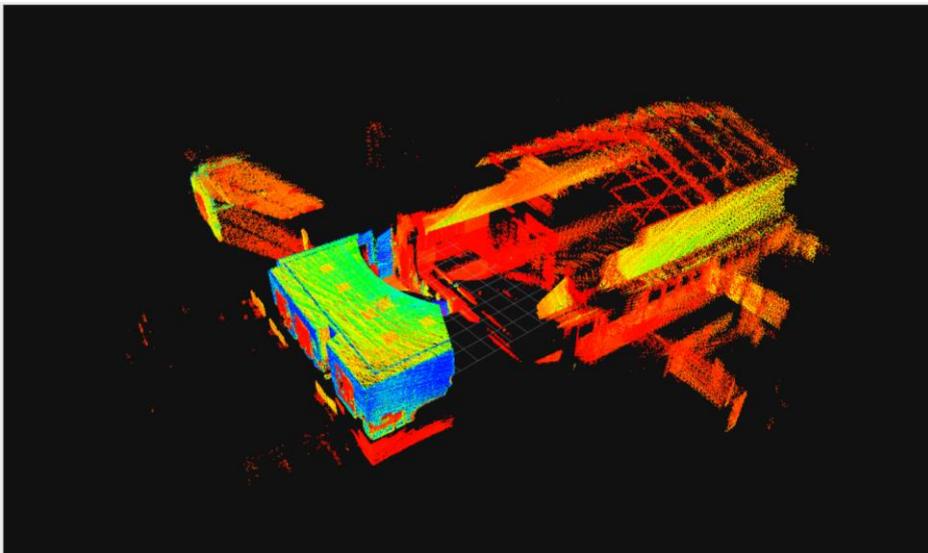


Figure 5.7 Point Cloud Map of the Building Obtained by the Designed System

For indoor mobile robotics applications, a complete map of the environment is critical for safe operation of a robot. In the point cloud measurements obtained in this study, the base of the sensor was kept stationary in the building. The designed system can be used on a mobile robot to completely map the environment with high density and provide a reliable source of information for SLAM algorithms.

5.2 High Density Mapping of an Obstacle

In order to identify obstacles in a point cloud map accurately, the point cloud obtained needs to have a high resolution. In this experimentation scenario, a cardboard box of dimensions $0.6 \times 0.6 \times 0.4$ m is placed 3.5 m away from the sensor. A picture taken from this scenario is given in Figure 5.8.



Figure 5.8 Placement of the Obstacle and the Sensor

In the first scan, the sensor is kept stationary, and the point cloud is accumulated over 5 seconds. Obtained point cloud map can be seen in Figure 5.9. It can be realized from this figure that, using this point cloud data, the presence of an obstacle may be detected, however it is almost impossible to identify the object. In this scenario, the

obstacle was placed 3.5 m away from the sensor. Noting that the Velodyne Puck has a scan range of 100 meters, it becomes much worse if an object that is farther away from sensor needs to be identified. At 100 meters the separation between the scan layers d_{layer} of the Velodyne Puck can be calculated as follows where 2° is the angular separation between each laser scan layer.

$$d_{layer} = (100 \text{ m}) \times \sin(2^\circ) = 3.49 \text{ m} \quad (5.1)$$

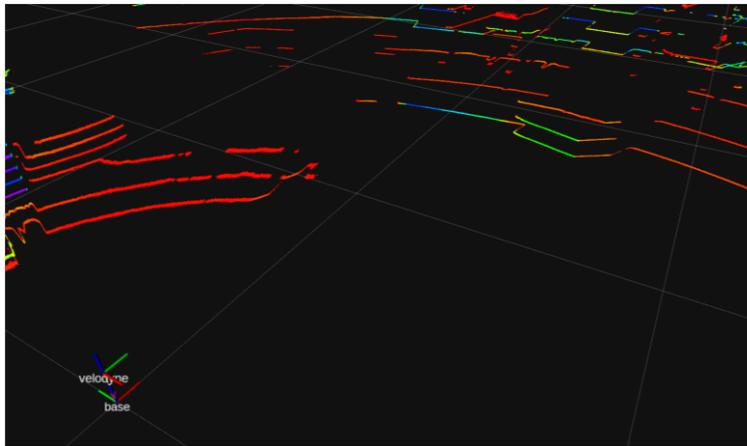


Figure 5.9 Obstacle Scan with Stationary LiDAR

Then the same object was scanned with the same motion described in Section 5.1 given to the sensor using the designed system. This time the point cloud was accumulated for 5 seconds to increase the density of the map obtained. The resulting point cloud map from this scan is given in Figure 5.10. It can be seen in this figure that the object was scanned with much higher point cloud density. The shape and the dimensions of the object can be extracted from the point cloud data and this point cloud data can safely be utilized in object recognition algorithms.

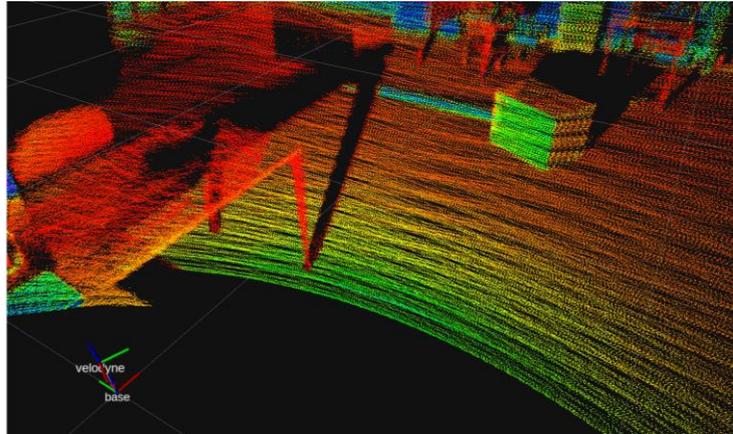


Figure 5.10 Obstacle Scan with Moving LiDAR

5.3 Controlled Density Variation

Various algorithms have been developed to control the motion of the designed system in such a way that the density of the point cloud obtained is varying as desired. It was aimed to obtain high density in certain regions, while still getting point cloud data from other regions. In Figure 5.11 a regular scan obtained by the system can be seen in which the point cloud density is homogeneously distributed.

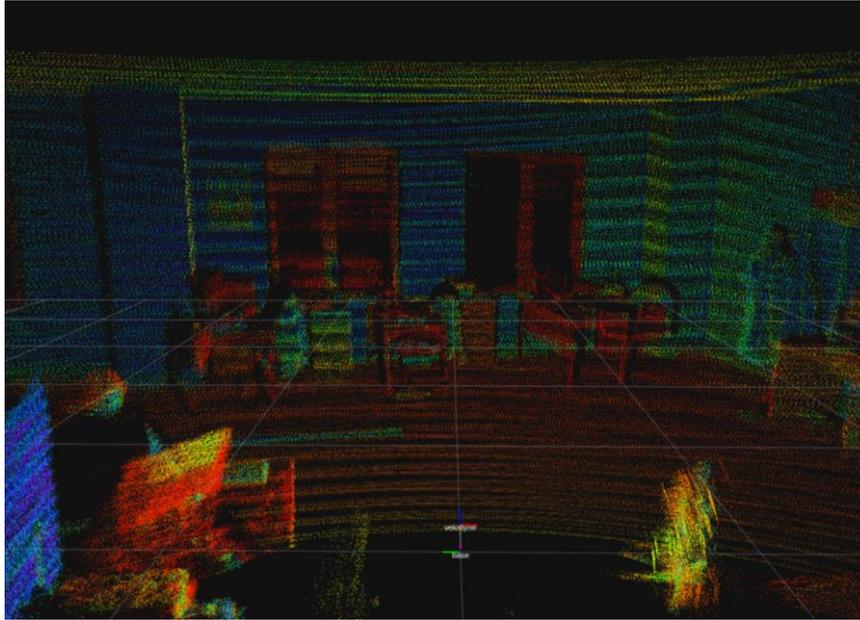


Figure 5.11 Constant Density Scan

On the other hand, in Figure 5.12 a point cloud data with emphasis on the center of the map with ± 16 degrees in the vertical. This high-density region is homogeneously distributed all around the sensor. This fact will also be shown clearly in the simulation results in the next section.

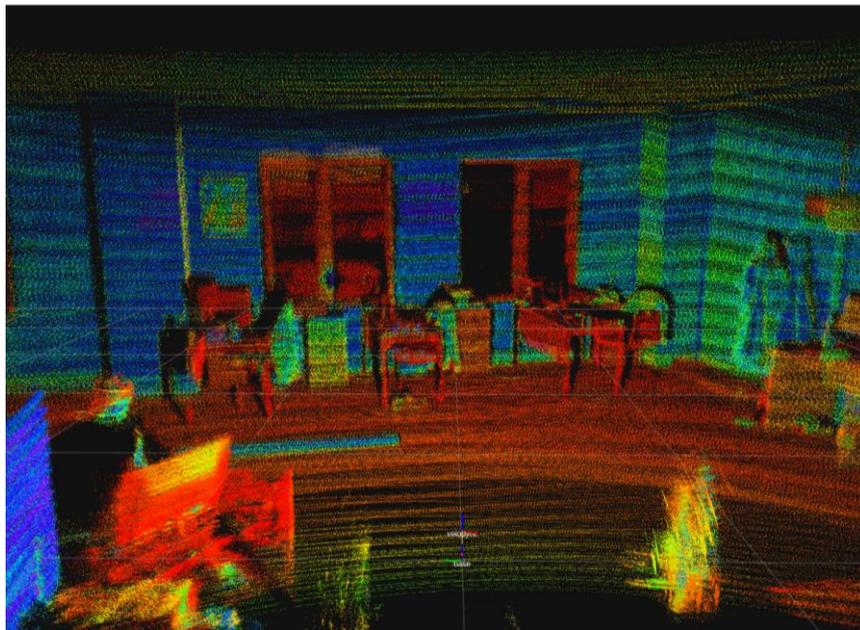


Figure 5.12 Increased Point Cloud Density at the Center

In another approach, the density of the point cloud was focused on the bottom part of the map. This could correspond to the sides of a vehicle from which the point cloud information is very crucial for safety reason. The result obtained by this motion algorithm can be seen in Figure 5.13. In this point cloud map, the emphasis is on the near bottom part of the sensor while still getting a full point cloud map of the environment.

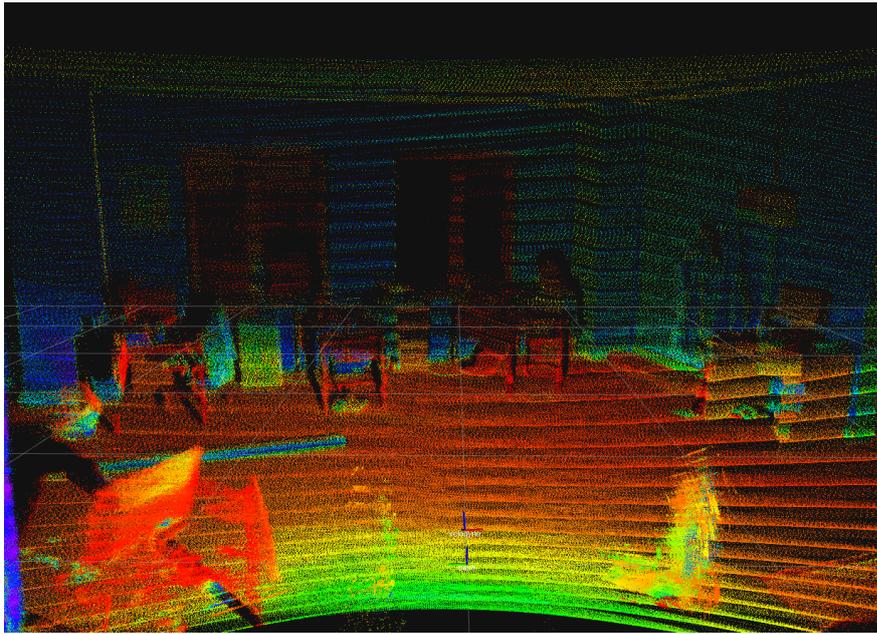


Figure 5.13 Increased Point Cloud Density at the Bottom

5.4 Robotic Simulation Results

Developed robotic simulation model of the designed system works similar to the real system thanks to its detailed description in the simulation environment. Rotations of each motor is controlled to control the position and orientation of the LiDAR sensor by using the developed ROS package. This package, additionally, publishes IMU sensor data, encoder readings and Velodyne Puck point cloud data, on different topics. IMU and encoder messages from the simulation environment are then used

to calculate the transformation between the base link and the LiDAR sensor. This transformation is calculated using the same mathematical model of the system and then this information is published as a ROS message which is used to transform the point cloud data. As a result, it was possible to test different motion algorithms on the simulated system and investigate the obtained point cloud map for each algorithm. These simulations allow us to see the coverage and the pattern of the point cloud map much clearly. In these simulations, the system is placed in a hollow sphere of 20 meters in diameter.

The designed system can achieve a full spherical scan of the environment. This can be seen in Figure 5.14. This scan was obtained in 2 seconds with a motion ± 90 degrees in each axis where the period of motion was 2 seconds.

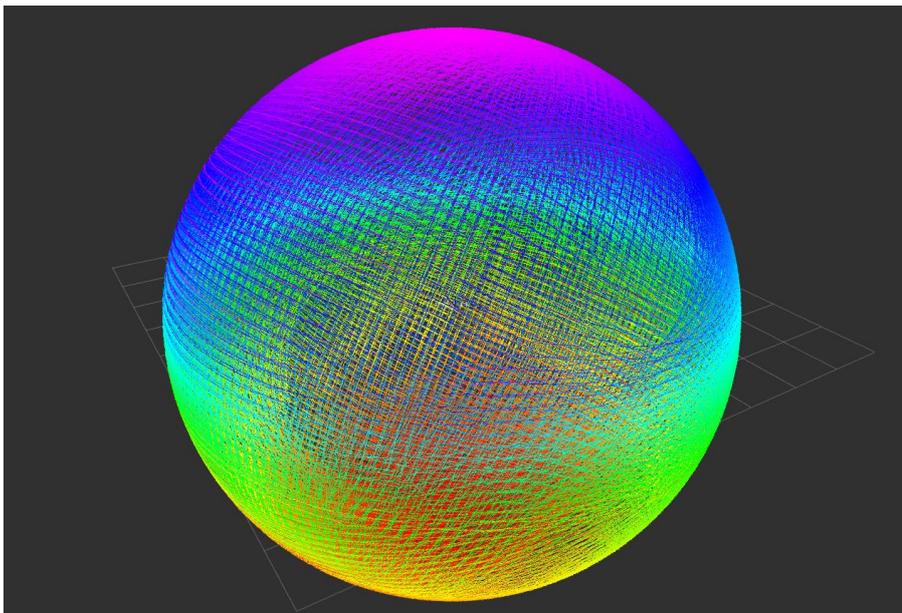


Figure 5.14 Simulated Scan Coverage

Motion control algorithms that can control the density of the point cloud map in certain regions were also tested in the simulated system in addition to the real-world tests given in the previous section. The results from this simulation allow us to see the point cloud data distribution more clearly in those scenarios. In Figure 5.15, point

cloud distribution obtained by focusing the density of point cloud to the horizontal center belt is clearly seen. Additionally, the homogeneity of the point cloud distribution in this high-density region can also be realized from this figure.

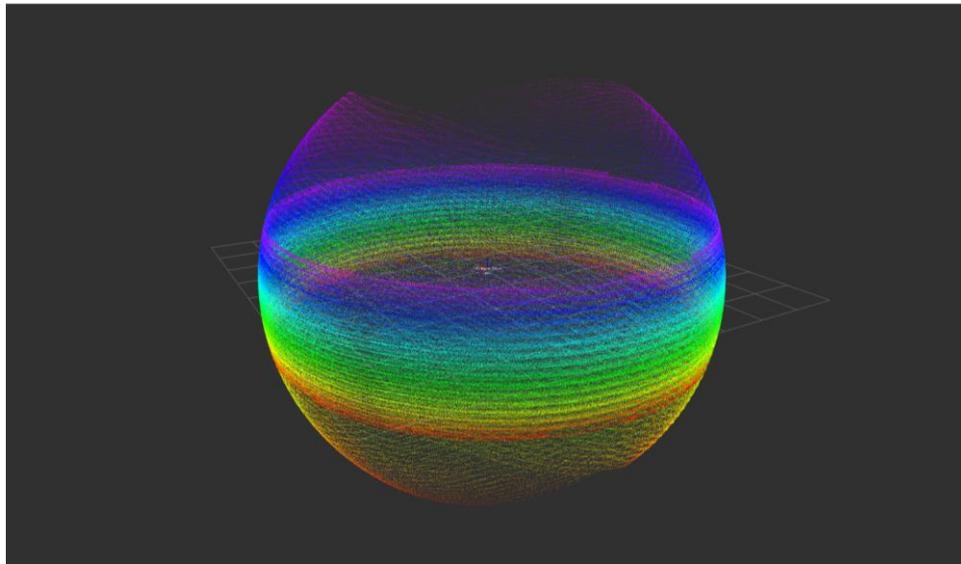


Figure 5.15 Simulated Increased Density at the Center

A final motion control algorithm to change the density distribution around the spherical scan region was tested in which a vertical belt region was focused to get more dense point cloud data in that region. The result is given in Figure 5.16.

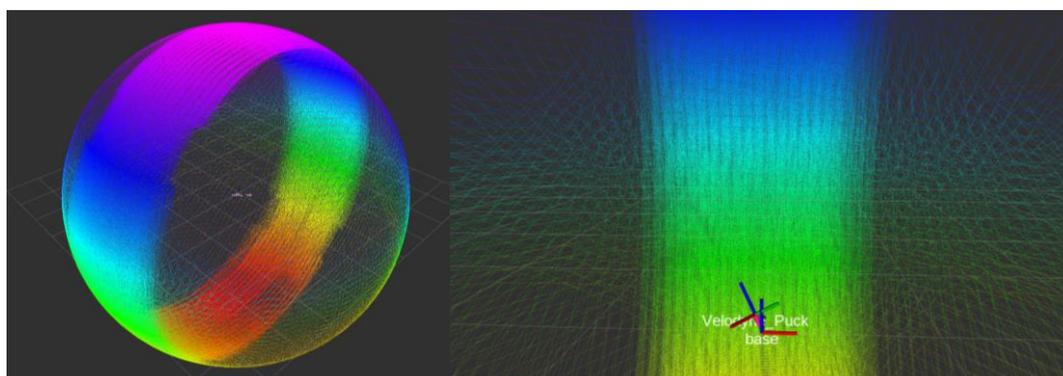


Figure 5.16 Vertically Focused Scan

5.5 Focused Region Scans

Motion control capability in two axes allows the manipulation of the point cloud distribution in such a way that the density of the point cloud obtained can be focused on specific regions using a similar approach used in [20]. In that paper, the focused region and the size of this region was manipulated by modulating the speed of rotation of the system. However, the shape of the region could not be changed. In our system, in addition to changing the position and the size of the focused region, the shape of this region can also be changed. Various methods have been implemented to increase the density of scans in specific regions. In the remainder of the chapter, results obtained in the robotic simulation environment by using these developed motion control algorithms to obtain high density scan regions are provided.

Focused Region with Constant Motion

Regular scanning motion to obtain a homogeneous full spherical scan of the environment can be manipulated by changing the yaw motion limits to increase the density of scan in a certain region. This region is pointed by the pitch axis and the azimuthal location of the region can be changed as desired. The algorithm is tested in Webots, and the result is given in Figure 5.17. In order to make the visualization easier, the sensor was utilized with a 180° horizontal field of view, resulting in an approximately half sphere.

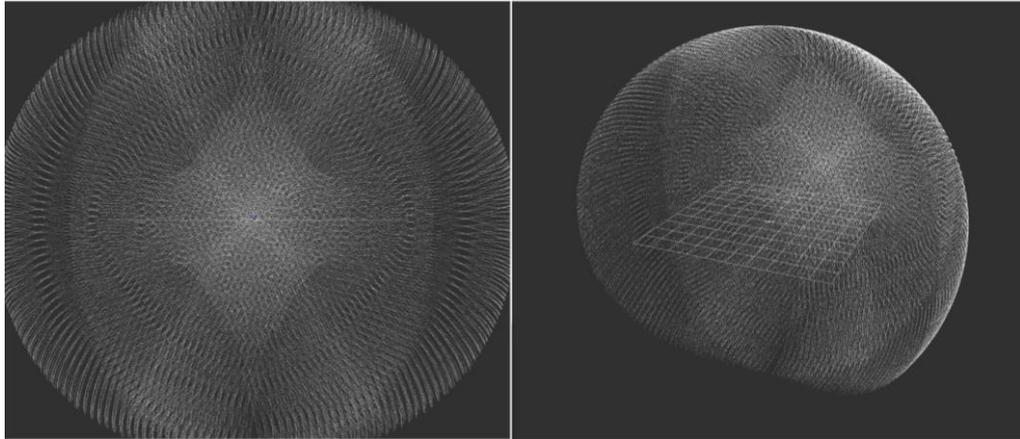


Figure 5.17 Constant Motion High Density Scan Region

Two-Band Focused Region

In this approach, high density scan is again obtained in the direction of the pitch axis. Throughout the full spherical scanning motion, two high density bands, separated by 90 degrees was obtained resulting in a focused rhomboid shaped region. In order to get these high-density bands, the motion of the system was slowed around ± 45 degree pitch angles and during this slow motion, the yaw axis was given an oscillatory motion to fill in the blank regions effectively in between the scan layers of the sensor. The azimuthal location of this focus region can be selected as desired around the sphere. The result obtained in the simulation environment is provided in Figure 5.18. Again, for clarity, the sensor was limited to 180° horizontal FOV.

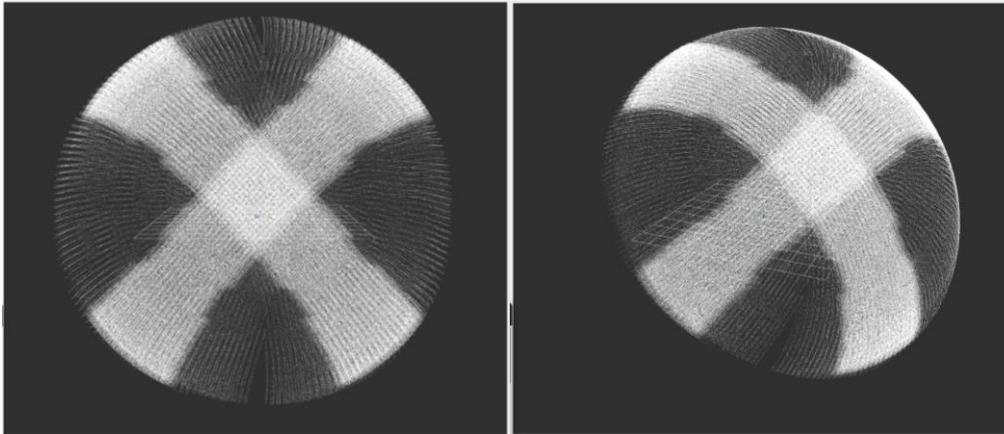


Figure 5.18 Two-Band Focused Region

Three-Band Focused Region

A similar approach used in the previous method is used with three bands intersecting to further increase the density of the point cloud in that specific region. In this scenario, a different shape is obtained in the scan region as can be seen in Figure 5.19. Similarly, for clarity, 180° horizontal FOV of the sensor is used.

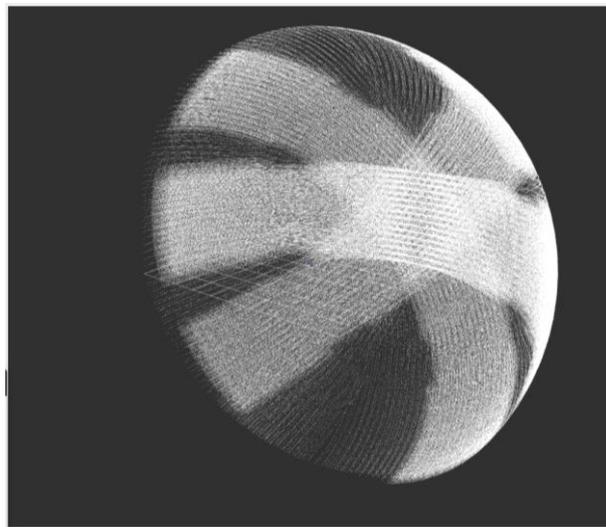


Figure 5.19 Three-Band Focused Region

Vertical Focus Region

Another motion control method allows the system to increase the density of scans in specific regions above or below the horizontal plane. In this method, a high-density, rhomboid like shape is obtained and this region can be positioned around the sphere with small distortions of the shape depending on the location. When the shape is near the horizontal plane, distortions become larger. Therefore, it works best for pitch angles greater than approximately 20 degrees from the horizontal plane. In other situations, one of the previous methods could be utilized effectively. One other important factor in this scan method is that the sensor needs to be operating at full 360 degrees horizontal field of view for the method to work, unlike previous methods. The result obtained in the simulated system is given in Figure 5.20. It should be noted that the pattern obtained is symmetric in the spherical scan region and thus, while one of the focused regions is below horizontal level the other is above it.

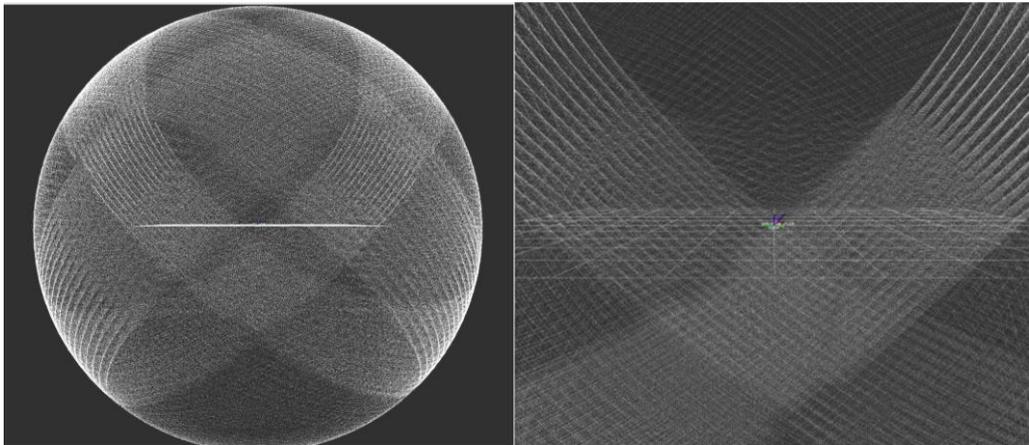


Figure 5.20 Vertical Focus Region

Rectangular Scan Region

A final motion control method for the designed system is developed to scan a rectangular region in any azimuthal direction. Since this region is placed on a spherical surface, its height and width are better represented in degrees as in spherical coordinates. The region is centered on the horizontal plane with approximately 32 degrees in vertical direction and adjustable width in the horizontal direction. In this scanning method, the sensor is utilized at 180 degrees horizontal FOV which halves the data transfer rate. This decreased data transfer rate becomes especially important in scenarios where multiple sensors need to be working at the same time without delay. In Figure 5.21, this high-density rectangular scan region can be seen in which the region is located at the front with a width of 30 degrees and a height of 32 degrees. It should also be noted that the two sides of the sensor is also scanned with a higher density than the other regions while the front rectangular region has the highest density.

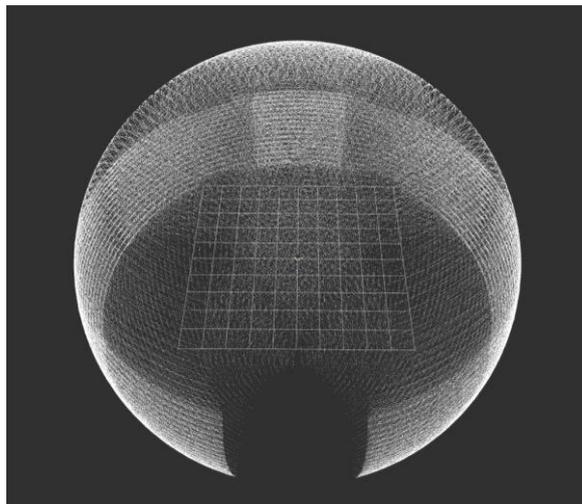


Figure 5.21 Rectangular Scan Region

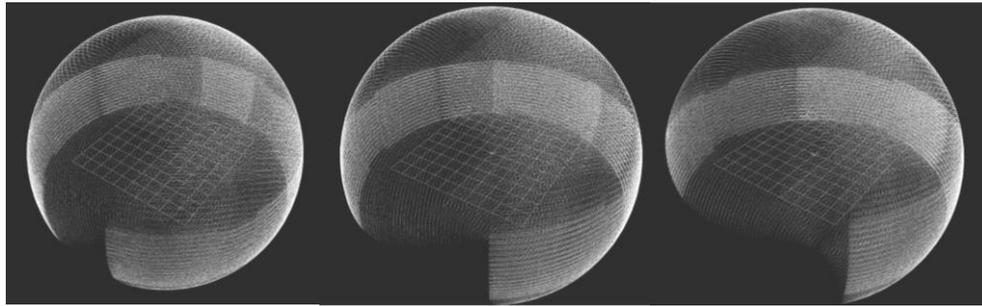


Figure 5.22 Rectangular Scan Region - 30°, 60° and 90° Wide

In Figure 5.23, the top view of the scan pattern with high density 30 degrees wide rectangular region is given. This figure illustrates the advantage of using this approach in robotics applications. For example, the sensor could be mounted in front of an unmanned vehicle. In that case, it is unnecessary to scan the back of the sensor and the front of the vehicle requires the most attention. The designed system with the developed motion control method could serve this purpose effectively by increasing the density of scans at the sides and further increasing the density in a rectangular region at the front of that vehicle.

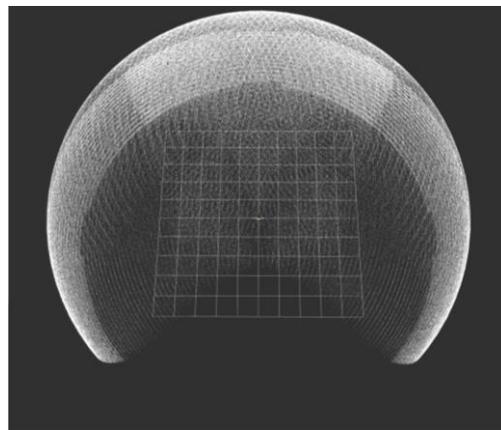


Figure 5.23 Rectangular Scan Region – Top View

CHAPTER 6

CONCLUSION

6.1 Conclusion

In this thesis, a 2-axis gimbal system that controls the position and the orientation of a Velodyne Puck LiDAR sensor has been developed and utilized to improve the mapping performance of the sensor. First, the mechanical design process has been carried out regarding the problem definition and the design requirements. The functionality of the system to be designed is divided into sub-functions and for each sub-function several solutions are proposed. After the evaluation of these solutions, the best concept is determined, and the 2-axis gimbal system is produced and assembled accordingly. Then, the forward and inverse kinematic modeling of the designed system has been performed using the Denavit-Hartenberg convention. This kinematic model is necessary to interpret the point cloud data obtained from the sensor in different positions and to position the sensor to the desired location. In order to transform the point cloud data from sensor's reference frame to the stationary base frame this mathematical model has been utilized.

After the mathematical modeling of the system, setup procedure of the BaseCamBGC Pro controller and necessary connections is explained. In this part, developed python package for serial communication with the controller is introduced. To be able to control the positions of the motors and read sensor data from the controller board, this python package was used. Additionally, PID auto-tuning procedure is also explained in detail. The performance of the tuned PID parameters is visualized by conducting several motion tests with the system.

Then the developed ROS workspace that constitutes the main software base for the system is explained in detail. In the system ROS 2 Foxy has been used as the

middleware suite. Developed ROS packages and available ROS packages used in the workspace are explained.

The designed 2-axis gimbal system is also simulated in Webots robotic simulator. Developed simulation model and the necessary controllers prepared for the simulation environment are presented. This robotic simulation model made it possible to quickly test the designed mechanical system and start the manufacturing processes immediately. In addition to that, ability to manipulate the simulation environment rapidly, and test the system in different scenarios proved to be a great advantage. Thanks to the simulated robotic model, it was possible clearly visualize LiDAR scan patterns obtained by the system.

Then the performance of the designed 2-axis gimbal for a Velodyne Puck is tested by utilizing the system in a mapping application of an indoor environment. It was shown qualitatively that the point cloud mapping of an environment with a stationary LiDAR sensor produces a point cloud map with many gaps in it, which cannot be used reliably. This would also make object identification using this point cloud data very hard. It was shown that, by using the designed system, a point cloud map of the environment with high density and higher field of view can be obtained. This point cloud data also provides a good source of information for object identification and SLAM algorithms.

Thanks to the motion control capability of the designed system in two axes, it was also possible to increase the density of scans in certain regions while still performing a full scan of the environment. Results that show the controlled density variation of the point using the developed motion control algorithms were presented. In addition to testing on the real system, these motions were also tested in the simulated robotic system. Simulation results provide us with a better understanding of how the point cloud data distribution is manipulated.

A final application with the designed system has been made in the simulated environment in which the motion of the system was controlled in such a way that the point cloud density is increased in certain regions. A similar approach is available in

the literature, however the improvement over this approach was also presented. It was possible to change the position, size, and the shape of this spot. This focused region can be used to scan certain objects of interest with high density if needed. And the ability to move this focused region allows the tracking of the object of interest that move relative to the sensor.

6.2 Future Work

The designed system provides a good and reliable method to obtain better point cloud data from the environment. This setup can be used for research focusing on algorithm development for processing point cloud data. These include SLAM, segmentation, and identification algorithms.

As it was explained, the designed system can be utilized in two different configurations without modifying the assembled system. However, in the second configuration several changes need to be made for the system to work. As a future work, the designed system can be configured in the second configuration and the performance comparison of the two configurations can be made.

In the present study, the modeling the dynamics of the system is not considered. Mathematical model of the system can be further developed to obtain a complete model of the system.

For a better cable management, use of slirings could be considered in the system. Since powerful motors are used in the current design, the resistance introduced by the cables against the rotations were negligible, however these cables introduce a weak point in the system and the usage of slirings would increase the lifetime of the system.

REFERENCES

- [1] S. Royo and M. Ballesta-Garcia, "An overview of lidar imaging systems for autonomous vehicles," *Appl. Sci.*, vol. 9, no. 19, Oct. 2019, doi: 10.3390/APP9194093.
- [2] T. Raj, F. H. Hashim, A. B. Huddin, M. F. Ibrahim, and A. Hussain, "A Survey on LiDAR Scanning Mechanisms," *Electronics (Switzerland)*, vol. 9, no. 5. Multidisciplinary Digital Publishing Institute, p. 741, Apr. 30, 2020, doi: 10.3390/electronics9050741.
- [3] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009, vol. 3, no. 3.2.
- [4] "ROS Documentation." <https://docs.ros.org/> (accessed Oct. 10, 2022).
- [5] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Sci. Robot.*, vol. 7, no. 66, May 2022, doi: 10.1126/scirobotics.abm6074.
- [6] C. K. Liu and D. Negrut, "The Role of Physics-Based Simulators in Robotics," *Annu. Rev. Control. Robot. Auton. Syst.*, vol. 4, no. 1, pp. 35–58, May 2021, doi: 10.1146/ANNUREV-CONTROL-072220-093055.
- [7] "Webots." <https://cyberbotics.com/> (accessed Oct. 18, 2022).
- [8] "Version R2019a - Webots Goes Open Source." <https://www.cyberbotics.com/doc/blog/Webots-2019-a-release> (accessed Oct. 19, 2022).
- [9] O. Michel, "Cyberbotics Ltd. Webots TM: Professional Mobile Robot Simulation." [Online]. Available: <http://www.cyberbotics.com>.
- [10] A. Farley, J. Wang, and J. A. Marshall, "How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion," *Simul. Model. Pract. Theory*, vol. 120,

p. 102629, Nov. 2022, doi: 10.1016/j.simpat.2022.102629.

- [11] T. Hegna, H. Pettersson, K. M. Laundal, and K. Grujic, “3D laser scanner system based on a galvanometer scan head for high temperature applications,” <https://doi.org/10.1117/12.888985>, vol. 8082, pp. 1195–1203, May 2011, doi: 10.1117/12.888985.
- [12] D. Wood and M. Bishop, “A novel approach to 3D laser scanning,” 2012.
- [13] A. Niewola and L. Podsedkowski, “A novel 3D laser scanner design for variable density scanning,” *12th Int. Work. Robot Motion Control. RoMoCo 2019 - Work. Proc.*, pp. 197–202, Jul. 2019, doi: 10.1109/ROMOCO.2019.8787369.
- [14] V. Vuthea and H. Toshiyoshi, “A Design of Risley Scanner for LiDAR Applications,” in *2018 International Conference on Optical MEMS and Nanophotonics (OMN)*, Jul. 2018, vol. 2018-July, pp. 1–2, doi: 10.1109/OMN.2018.8454641.
- [15] S. Bi, C. Yuan, C. Liu, J. Cheng, W. Wang, and Y. Cai, “A Survey of Low-Cost 3D Laser Scanning Technology,” *Appl. Sci.*, vol. 11, no. 9, p. 3938, Apr. 2021, doi: 10.3390/app11093938.
- [16] G. Aslan, “Design and Implementation of a Scanning Platform for Mobile Robotics,” Middle East Technical University, 2012.
- [17] M. Matsumoto and S. Yuta, “3D laser range sensor module with roundly swinging mechanism for fast and wide view range image,” *IEEE Int. Conf. Multisens. Fusion Integr. Intell. Syst.*, pp. 156–161, 2010, doi: 10.1109/MFI.2010.5604484.
- [18] K. Ohno, T. Kawahara, and S. Tadokoro, “Development of 3D laser scanner for measuring uniform and dense 3D shapes of static objects in dynamic environment,” *2008 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2008*, vol. 2008-January, pp. 2161–2167, 2008, doi: 10.1109/ROBIO.2009.4913337.

- [19] T. Yoshida, K. Irie, E. Koyanagi, and M. Tomono, “A sensor platform for outdoor navigation using gyro-assisted odometry and roundly-swinging 3D laser scanner,” *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 1414–1420, 2010, doi: 10.1109/IROS.2010.5652172.
- [20] T. Yoshida, K. Irie, E. Koyanagi, and M. Tomono, “3D laser scanner with gazing ability,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 3098–3103, 2011, doi: 10.1109/ICRA.2011.5980385.
- [21] M. G. Ocando, N. Certad, S. Alvarado, and Á. Terrones, “Autonomous 2D SLAM and 3D mapping of an environment using a single 2D LIDAR and ROS,” in *Proceedings - 2017 LARS 14th Latin American Robotics Symposium and 2017 5th SBR Brazilian Symposium on Robotics, LARS-SBR 2017 - Part of the Robotics Conference 2017*, Dec. 2017, vol. 2017-Decem, pp. 1–6, doi: 10.1109/SBR-LARS-R.2017.8215333.
- [22] “Velodyne Lidar.” <https://velodynelidar.com/> (accessed Nov. 22, 2022).
- [23] A. Pfrunder, P. V. K. Borges, A. R. Romero, G. Catt, and A. Elfes, “Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D LiDAR,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 2601–2608, doi: 10.1109/IROS.2017.8206083.
- [24] T. Neumann, E. Dülberg, S. Schiffer, and A. Ferrein, “A Rotating Platform for Swift Acquisition of Dense 3D Point Clouds,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9834 LNCS, Springer Verlag, 2016, pp. 257–268.
- [25] T. Neumann, A. Ferrein, S. Kallweit, and I. Scholl, “Towards a Mobile Mapping Robot for Underground Mines,” in *Proceedings of 2014 PRASA, RobMech and AfLaT Int. Joint Symposium, Cape Town, South Africa*, 2014, no. November.

- [26] J. Morales, V. Plaza-Leiva, A. Mandow, J. Gomez-Ruiz, J. Serón, and A. García-Cerezo, “Analysis of 3D Scan Measurement Distribution with Application to a Multi-Beam Lidar on a Rotating Platform,” *Sensors*, vol. 18, no. 2, p. 395, Jan. 2018, doi: 10.3390/s18020395.
- [27] M. Cao, P. Su, H. Chen, S. Tang, and Y. Liu, “3-D Dense Rangefinder Sensor with a Low-Cost Scanning Mechanism,” *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, doi: 10.1109/TIM.2020.3016415.
- [28] H. Geng, Z. Gao, G. Fang, and Y. Xie, “3D Object Recognition and Localization with a Dense LiDAR Scanner,” *Actuators*, vol. 11, no. 1, p. 13, Jan. 2022, doi: 10.3390/act11010013.
- [29] J. Bula, M.-H. Derron, and G. Mariethoz, “Dense point cloud acquisition with a low-cost Velodyne VLP-16,” *Geosci. Instrumentation, Methods Data Syst.*, vol. 9, no. 2, pp. 385–396, Oct. 2020, doi: 10.5194/gi-9-385-2020.
- [30] *VLP-16 User Manual*. 2019.
- [31] “BaseCamBGC Pro.” <https://www.basecamelectronics.com/pro/> (accessed Nov. 23, 2022).
- [32] “AS5048A High-Resolution Position Sensor.” <https://ams.com/en/as5048a> (accessed Oct. 13, 2022).
- [33] M. Kemal Ozgoren, *Kinematics of General Spatial Mechanical Systems*. 2020.
- [34] *SimpleBGC32 Serial API protocol specification*. 2022.
- [35] *SimpleBGC 32bit 3-Axis Software User Manual*. 2022.
- [36] T. Foote, “tf: The transform library,” in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, Apr. 2013, pp. 1–6, doi: 10.1109/TePRA.2013.6556373.
- [37] “Velodyne ROS Drivers.” <https://github.com/ros-drivers/velodyne.git>

(accessed Dec. 23, 2022).

- [38] “Webots ROS2 Interface.” https://github.com/cyberbotics/webots_ros2.git
(accessed Dec. 23, 2022).