Institute for Computer Science XVII
Robotics

Bachelor's thesis

3D Mapping with Rolling Spheres Using a Livox Mid-360 LiDAR

Marawan Salah Abdelrahman Khalil Khalil

August 2025

First reviewer: Prof. Dr. Andreas Nüchter
Advisor: M.Sc Fabian Arzberger

Acknowledgements

I would like to thank God for granting me good health, strength, and blessings to complete this work. I am deeply grateful to my family for their continuous support and encouragement throughout my academic journey. I would also like to express my sincere gratitude to my supervisor, Prof. Dr. Andreas Nüchter, for giving me the opportunity to work on this exciting project, for his invaluable guidance and constructive feedback, and for providing the necessary resources and facilities to conduct this research. I am especially thankful to Fabian Arzberger for his assistance, inspiring discussions, and the valuable time we spent together working on research development and improvements. I would also like to thank Dr. Michael Blieber for his generous technical advice and support whenever I needed guidance. I am grateful to my friends and colleagues at the Robotics Department for their encouragement and camaraderie. Finally, I would like to thank my friends from Cairo and Würzburg for their unwavering support and friendship.

Abstract

Spherical robots offer unique advantages for mapping applications in hazardous or confined environments, unlike traditional wheeled or legged robots, thanks to their protective shells and omnidirectional mobility. Recent research has focused on developing more robust motion mechanisms, and some have incorporated technologies like LiDAR for 3D mapping, further expanding their capabilities in complex and unpredictable environments. However, the rapid evolution of LiDAR–inertial algorithms and modern hardware platforms presents new opportunities for spherical SLAM, while the integration of actuation with mapping and localization remains largely unexplored.

To address this, this work presents two complementary spherical mapping systems: a lightweight, non-actuated design and an actuated variant featuring internal pendulum-driven locomotion. Both systems are equipped with a Livox Mid-360 solid-state LiDAR sensor and run LiDAR-Inertial Odometry (LIO) algorithms in real-time on resource-constrained hardware (Raspberry Pi 5). We assess the mapping accuracy of these systems by comparing the resulting 3D point-clouds from the LIO algorithms to a ground truth map from a high-precision terrestrial laser scanner. A comparative evaluation highlights distinct performance characteristics: the non-actuated sphere achieves higher mapping accuracy, while the actuated system provides denser point-clouds and shows potential for autonomous navigation. Furthermore, the performance of state-of-the-art LIO algorithms deteriorates due to the high dynamic movement introduced by the spherical locomotion, leading to globally inconsistent maps and sometimes unrecoverable drift.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent decades, robotics has evolved from a niche discipline into a transformative technology impacting a wide range of industries, including manufacturing, healthcare, space exploration, and personal assistance. Among the diverse types of robotic systems, spherical robots have recently begun to attract increased attention from researchers. These robots represent a relatively unconventional design compared to more familiar, rotation-restricted systems such as UAVs, handheld devices, and wheeled vehicles. Unlike traditional wheeled or legged robots, spherical robots offer several key advantages, including omnidirectional movement, enhanced maneuverability in unpredictable environments, and improved protection for internal components. These features make them well-suited for a variety of applications such as surveillance, inspection, and environmental monitoring in hazardous conditions. Their spherical design enables them to access environments that are difficult or impossible for other robotic systems to navigate, such as steep tunnels, underground mines, narrow passageways, and other confined or dangerous spaces. The sealed outer shell of a spherical robot provides full protection against dust, hazardous chemicals, liquids, and external impacts.

Recent research has focused on developing more robust motion mechanisms [8–10] and incorporating technologies like LiDAR for 3D mapping, further expanding their capabilities in complex and unpredictable environments[1, 11, 12]. However, the rapid evolution of LiDAR–inertial algorithms and modern hardware platforms presents new opportunities for spherical SLAM, while the integration of actuation with mapping and localization remains largely unexplored. To address this, we present and evaluate two custom-built spherical robots (see Fig. 1.1) equipped with LiDAR-based SLAM systems. Both systems integrate FAST-LIO2[13], FAST-LIVO2[14], and DLIO[15], which are state-of-the-art LiDAR-inertial odometry algorithms. The primary contribution of this work is the comparative analysis of these algorithms' performance on spherical platforms and their validation against terrestrial laser scanning ground truth data. This evaluation provides insights into the practical applicability of different SLAM approaches for spherical robots and establishes benchmarks for future developments in this emerging field.

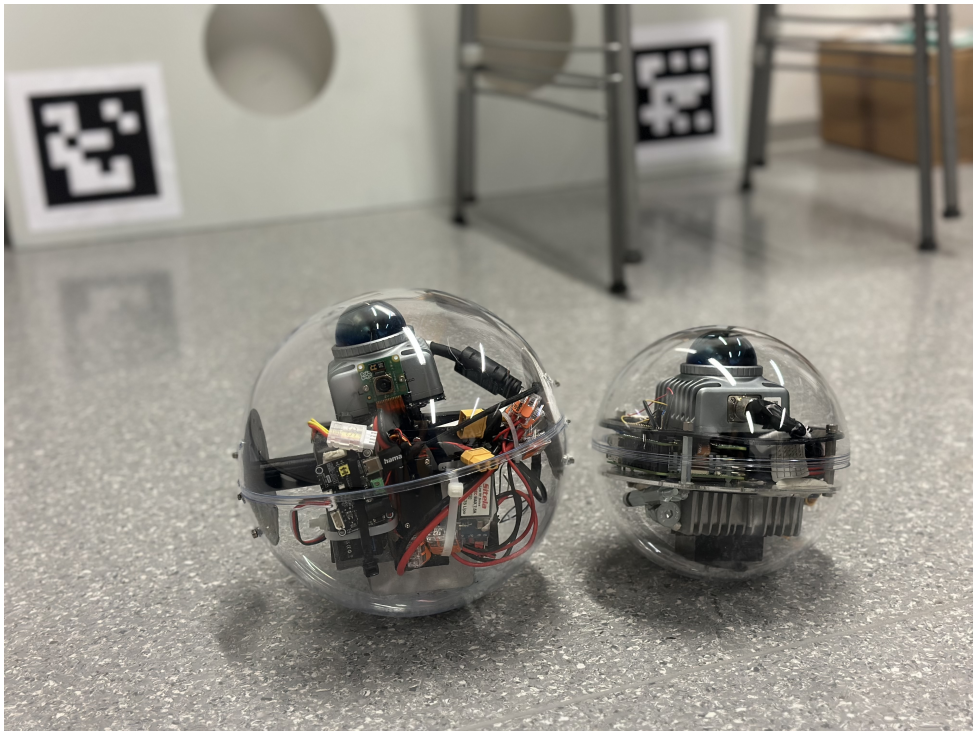


Figure 1.1: (Left:) 20 cm diameter actuated sphere. (Right:) 16 cm diameter non-actuated sphere. A video where both spheres are moving is available at `https://youtube.com/shorts/lxTF85HK-zY`.

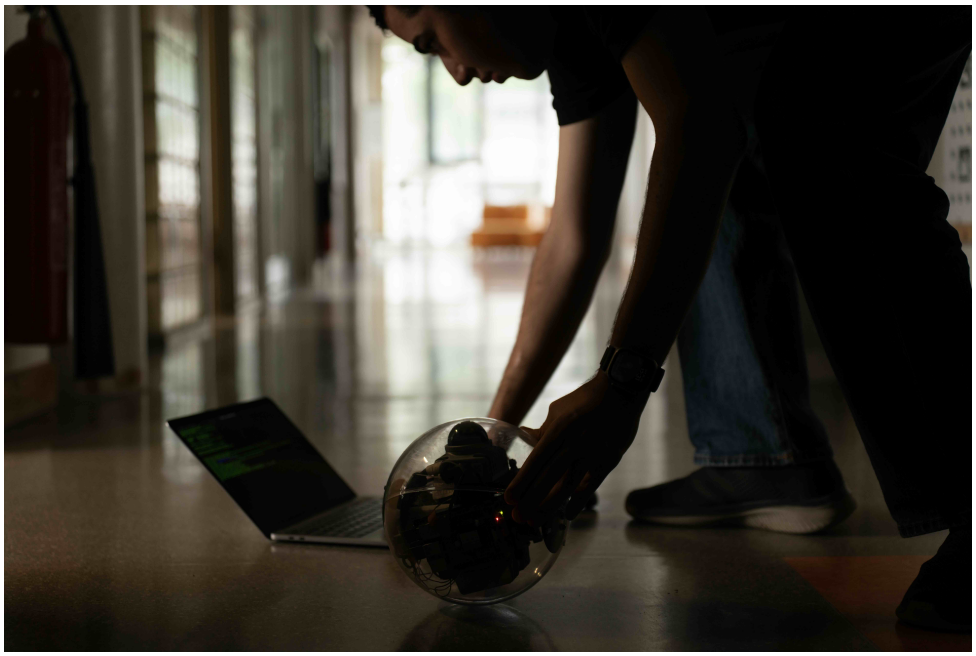


Figure 1.2: Actuated Sphere being tested and evaluated in Computer Science building at the University of Würzburg.

1.1 Outline

In the next Chapter, we review recent SLAM developments for spherical platforms, with emphasis on LiDAR-inertial methods, followed by advancements in spherical robot design. In Chapter 3, we provide an overview of the fundamental concepts and technologies that relates to our work, including Inertial Measurement Units (IMUs), LiDAR technology, spherical robot types, an introduction to SLAM algorithms and Control Strategies. In Chapter 4, we provide a description of the hardware implementation of the two spherical robots. Subsequently, we describe the software integration, as well as the motion control mechanisms used for the actuated sphere. Finally, we provide a comprehensive evaluation of the mapping accuracy by comparing the maps of the proposed systems with a ground truth map obtained using high-precision terrestrial laser scanning (TLS).

Chapter 2

State of the Art

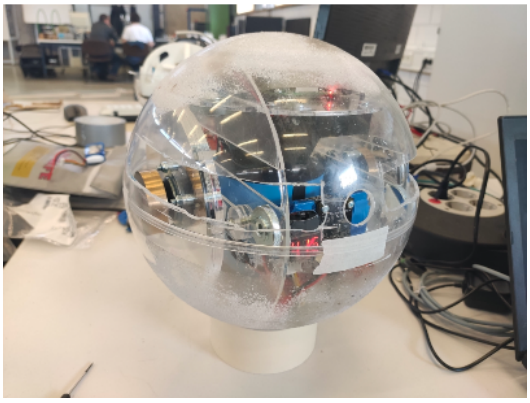
This chapter reviews the current landscape of spherical robotics, focusing on two critical aspects: SLAM algorithms adapted for rolling platforms and locomotion mechanisms that enable controlled motion. The unique constraints of spherical robots—enclosed sensors, aggressive rolling dynamics, and operation in confined spaces—drive both the algorithmic choices for perception and the mechanical strategies for mobility. Section 2.1 surveys spherical SLAM with a focus on LiDAR–inertial methods. Section 2.2 reviews various locomotion mechanisms.

2.1 Spherical SLAM

Spherical Simultaneous Localization and Mapping (SLAM) is an emerging area within mobile robotics, offering promising solutions for robust mapping in constrained or hazardous environments. Unlike traditional platforms, spherical robots encapsulate their sensors within a protective shell and rely on rolling locomotion. This unique configuration introduces both opportunities and significant challenges for SLAM algorithms. One of the earliest spherical SLAM prototypes was L.U.N.A. [16], which employed a 2D LiDAR and IMU inside a rolling spherical shell. The design validated the feasibility of spherical SLAM. It also featured actuation through internal flywheels, using an IBCOAM (Impulse by Conservation of Angular Momentum) mechanism.

A major milestone in this field was the DAEDALUS project [11], funded by the European Space Agency, which proposed a fully enclosed spherical robot for the autonomous exploration of lunar lava tubes. The robot was proposed to be equipped with LiDAR and internal actuators, and its design focused on resilience to lunar regolith and harsh environmental conditions. A core challenge in spherical SLAM is the aggressive and off-centered rotation induced by rolling locomotion. These motions produce high angular velocities and dynamic behavior across all principal axes. This significantly degrades pose estimation accuracy and leads to error accumulation in the map. This problem is further compounded by the absence of magnetometer use—often intentionally excluded due to the unreliability of magnetic field data in planetary environments. This leads to uncorrected yaw drift in IMU-based odometry, especially during prolonged navigation.

To address these issues, Arzberger et al. [1, 2, 12] introduced specialized filtering techniques for spherical systems. Their Delta Filter is a lightweight, real-time, multi-trajectory



(a) L.U.N.A. robot with internal flywheel actuation [16].

(b) DAEDALUS robot for lunar exploration [17].

Figure 2.1: Spherical robots concepts

pose estimation method that fuses unreliable trajectories, such as those from IMUs and stereo visual-inertial odometry (VIO), into a more robust estimate, without requiring explicit sensor uncertainty modeling. The filter operates on pose changes ("deltas"), uses a probabilistic weighting scheme for translation estimation, and applies rotational interpolation via spherical linear interpolation (Slerp). A follow-up Kalman Filter design extended this approach by incorporating a covariance-aware model, enhancing pose estimation accuracy during rapid and complex motion.

Figure 2.2: (Left:) A sphere platform featuring a 'Hesai Pandar-XT32'[1]. (Right:) A 'Livox Mid-100' Spherical platform from [2]

While recent studies (e.g. [1]) have compared their results with state-of-the-art SLAM systems such as DLIO[15] and FAST-LIO2[13], the field continues to evolve rapidly. More recent SLAM algorithms (e.g., FAST-LIVO2 [14]), advanced LiDAR sensors like the MID-360 and RoboSense Airy, and modern single-board computers such as the Raspberry Pi 5 with PCIe support and Gigabit Ethernet are pushing the boundaries of what is possible. In this context,

we introduce our first prototype, the non-actuated sphere, which leverages the processing power of the Raspberry Pi 5 16GB RAM model. This design enables real-time SLAM on a compact spherical platform, offering improved power efficiency, extended battery life, and reduced overall system cost.

2.2 Spherical locomotion

Spherical locomotion has attracted significant research interest in recent years, driven by the pursuit of optimal mobility mechanisms. Although spherical robots may appear mechanically simple, a wide variety of locomotion strategies have been developed—and continue to emerge—requiring sophisticated control systems to manage the complex dynamics of rolling locomotion and internal mass distribution.

One of the most well-known approaches is the Internal Driving Unit (IDU), or differential drive, used in commercial robots like the Sphero Bolt+ and BB-8. Akella et al.[18] analyzed BB-8's internal wheel-driven system and demonstrated that effective control was achieved using only two actuators. However, they highlighted a key limitation: the robot's geometry prevents controlled sliding and reduces effectiveness on inclined or uneven terrain.

In contrast, Zevering et al. [16] introduced L.U.N.A., a spherical robot designed for autonomous 3D mapping in lunar caves. Rather than using wheels or rods, L.U.N.A. relies on internal flywheels to generate motion via the IBCOAM method. This approach enables a compact form factor and protects internal electronics, providing advantages particularly well suited to harsh and remote terrain. The robot demonstrated reliable motion on soft surfaces such as sand and rubber. However, limitations remain, including vibrational instability caused by unbalanced flywheels, reduced performance on inclined low-friction surfaces, and pose estimation errors due to unsynchronized sensor data.

In a following study, Zevering et al. [19] proposed a rod-driven spherical robot, also targeting lunar cave exploration. This design uses external linear actuators to push against the environment to induce motion. While it improves adaptability to rugged terrain and sharp obstacles, it introduces new challenges, such as oscillatory behavior from fixed-speed actuators, limited effectiveness on slopes, and the need for higher-power actuation when traversing dusty or soft ground.

Beyond differential and flywheel-based designs, several researchers have explored pendulum-driven locomotion due to its mechanical simplicity, energy efficiency, and natural stability. Oevermann et al. [8], Ren et al. [9], and Kolbari et al. [10] developed spherical robots that use an internal heavy pendulum as the primary driving mechanism. While their configurations differ in terms of shell design and target applications, they share a common reliance on pendulum-based locomotion and have demonstrated robust movement across a variety of terrains. A notable example is RoboBall [8], which features a novel soft pressurized shell and a two-degree-of-freedom internal pendulum. This robot successfully navigates gravel, grass, steep inclines, and even floats and maneuvers on water. However, the deformable shell introduces complex dynamic behavior. The researchers addressed this using a Linear Quadratic Regulator (LQR) for steering and a model-based proportional controller for driving. Ren et al. [9] experimented with both robust servo LQR (RSLQR) and PID-based stabilization strategies to enhance motion control

and responsiveness, whereas Kolbari et al. [10] adopted only PID control for stabilization.

RoboBall's experiments further revealed that internal pressure and shell deformation significantly affect dynamic behavior—especially due to the presence of a "dead zone" where balance control becomes unstable during motion. Taken together, these pendulum-based locomotion strategies highlight the value of combining mechanical simplicity with robust control to enable adaptive movement in unpredictable environments. While a variety of innovative locomotion methods have been proposed for spherical robots, including flywheel-, rod-, and pendulum-driven systems, few have been evaluated in conjunction with advanced LiDAR-inertial odometry algorithms under real-world motion dynamics, as briefly discussed in Section 1. This paper addresses this gap by introducing our second prototype, a pendulum-driven spherical robot designed to achieve both robust locomotion and accurate, real-time mapping performance within the constraints of a compact platform.

Chapter 3

Fundamentals and Theoretical Background

3.1 Inertial Measurement Unit (IMU)

An Inertial Measurement Unit (IMU) is an electronic device—either electromechanical or solid-state—that measures the specific force, angular rate, and, in some cases, the magnetic field of the object or body to which it is attached. A 6-axis IMU typically employs accelerometers and gyroscopes, while a 9-axis IMU additionally incorporates a magnetometer. IMUs are widely used in aerospace, robotics, satellite navigation, and various other applications.

Micro-Electro-Mechanical Systems (MEMS) technology is commonly used in the production of IMUs. This technology enables the manufacturing of components that are extremely small and lightweight, with low power consumption, while also offering high reliability and cost efficiency. [20, 21]

3.1.1 Accelerometer

An accelerometer in an IMU is a device that measures the linear acceleration of an object along the x, y, and z axes. From acceleration data, velocity and position can be derived by integrating the measured acceleration values over time. MEMS accelerometers measure the forces acting on them, which may arise from motion, gravity, or vibrations.

The accelerometer contains a proof mass connected to its frame via mechanical springs. These springs allow the proof mass to move along a direction known as the sensitivity axis. By monitoring the displacement of the proof mass from its original position, the accelerometer can determine the applied acceleration [3].

This displacement is typically measured using electrical capacitance. The device includes multiple differential capacitors, consisting of stationary electrodes positioned on either side of the proof mass, to accurately detect its movement, as illustrated in Fig.3.1.

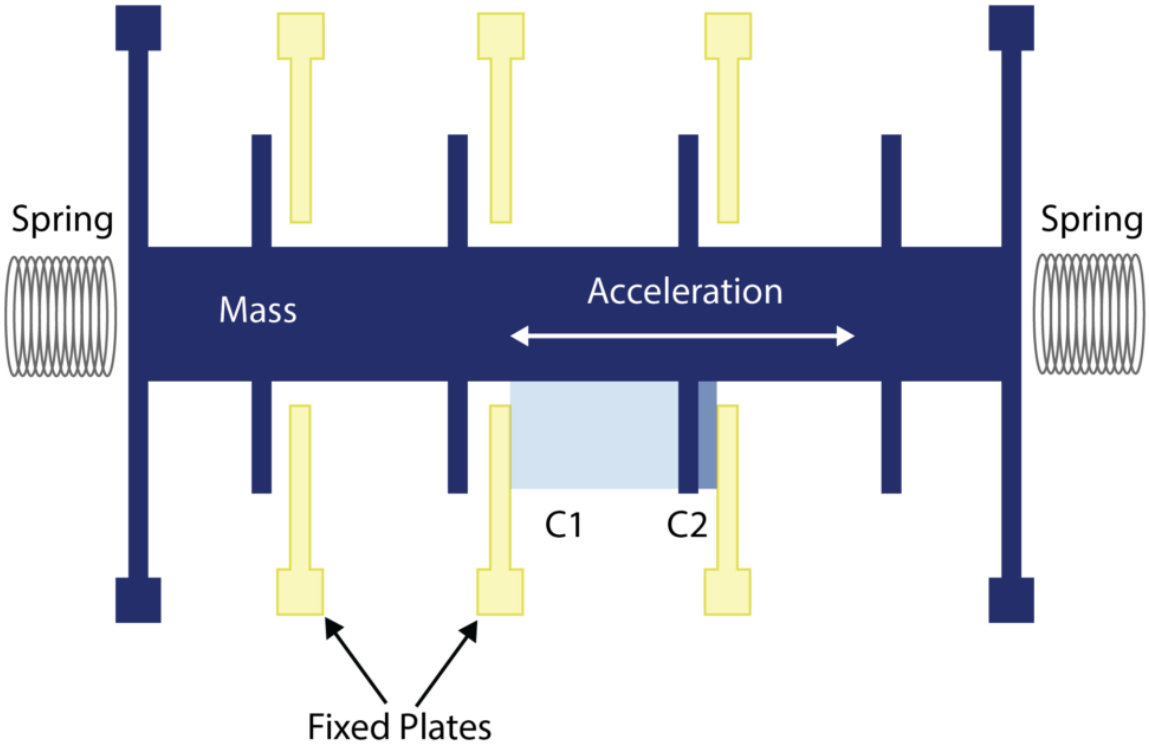


Figure 3.1: Inside a MEMS accelerometer. The proof mass has fixed plates "fingers" between capacitor electrodes. At rest, they're centered, giving zero acceleration. Movement during acceleration shifts the fingers, changing capacitance, which is used to calculate acceleration[3]

3.1.2 Gyroscope

Gyroscopes are devices used to measure or maintain rotational motion. The unit of angular velocity (a measure of how fast an object rotates) is typically expressed in degrees per second (°/s) or revolutions per second (RPS). Gyroscopes have a wide range of applications, including automotive rotation detection, inertial navigation, robotics, and motion tracking.

A triple-axis MEMS (Micro-Electro-Mechanical Systems) gyroscope can measure rotation around three perpendicular axes: x, y, and z. The gyroscope sensor within a MEMS device is extremely small—typically between 1 and 100 micrometers, comparable to the thickness of a human hair. When the gyroscope is rotated, a small resonating mass inside the sensor shifts in response to changes in angular velocity. This movement generates minute electrical signals, which are then amplified and processed by a host microcontroller to determine the rotational

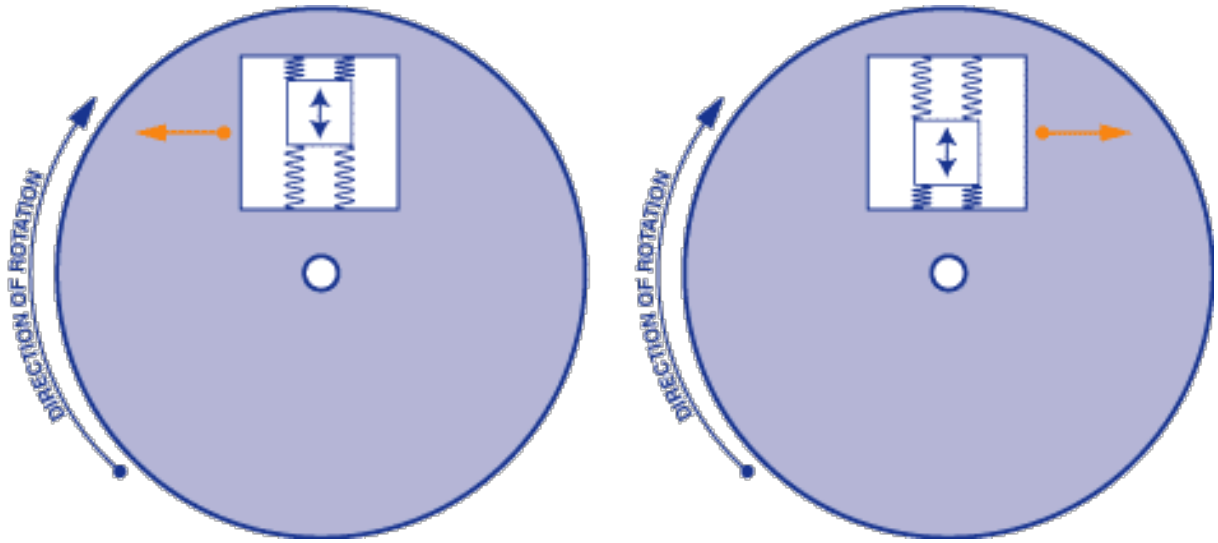


Figure 3.2: Inside a MEMS gyroscope. The small resonating mass shifts in response to angular velocity changes, generating electrical signals [4].

motion [4, 22].

3.1.3 Magnetometer

A *magnetometer* is a sensor that measures the strength and direction of a magnetic field, most commonly the Earth's magnetic field. In an IMU, magnetometers are primarily employed to estimate the heading, i.e., the orientation relative to magnetic north. This is achieved using a three-axis magnetometer, where three orthogonal sensing elements measure the magnetic field components along the x, y, and z axes. From these measurements, the magnetic heading is computed and, when fused with gyroscope and accelerometer data, a north-referenced orientation can be obtained [20].

Several physical principles can be exploited in the construction of magnetometers, including fluxgate, anisotropic magnetoresistance (AMR), and giant magnetoresistance (GMR). For MEMS-based magnetometers, however, the *Hall effect* is the most widely adopted mechanism. In this principle, a voltage difference is generated across a conductor when a magnetic field is applied perpendicular to the direction of current flow, allowing for direct measurement of magnetic field strength [23].

Despite their usefulness, magnetometers also present certain limitations. Their high sensitivity makes them susceptible to *electromagnetic interference (EMI)* from nearby electronic devices as well as distortions caused by surrounding ferromagnetic materials. These disturbances, commonly referred to as *soft-iron* and *hard-iron* effects, can significantly reduce measurement accuracy. Consequently, proper calibration techniques, such as ellipsoid fitting or offset compensation, are necessary to ensure reliable heading estimation [24].

3.2 LiDAR Technology

Light Detection and Ranging (LiDAR) is a transformative sensing technology that enables accurate 3D perception of the environment. Unlike traditional cameras, which passively capture reflected light, LiDAR actively emits laser pulses and measures the time-of-flight of the returned signals. By repeating this process thousands of times per second, LiDAR builds dense point clouds that capture both the geometry and scale of the surrounding environment [25].

Several LiDAR principles are commonly used:

Range-Finding LiDAR

The most fundamental form, range finders, emit laser pulses and measure the time difference between emission and reception. This allows accurate distance measurements with high spatial resolution, particularly effective for small-scale targets due to the short laser wavelength compared to radar [26].

Differential Absorption LiDAR (DIAL)

DIAL systems measure atmospheric composition by exploiting the absorption characteristics of gases. Two laser beams are used: one at a wavelength corresponding to the absorption peak of a target gas (*on-line*), and another at a nearby non-absorbing wavelength (*off-line*). The ratio of the returned signals reveals the presence and concentration of the gas [26].

Doppler LiDAR

Doppler LiDAR utilizes the frequency shift (Doppler effect) of the returned signal to measure the velocity of moving particles such as aerosols. This technique is widely used for wind profiling, turbulence detection, and monitoring hazardous atmospheric conditions around launch sites [26].

3.2.1 Hybrid Solid-State LiDAR Technology

Recent advancements in LiDAR move away from traditional mechanical spinning sensors toward solid-state and hybrid designs. These approaches reduce moving parts, improve robustness, and enable compact form factors suitable for mobile robots and drones. Instead of relying on repetitive scan lines, some hybrid LiDARs employ non-repetitive scanning patterns that gradually fill in the environment with dense coverage over time. This results in rich point clouds while maintaining lower cost and power consumption compared to high-end mechanical systems [27, 28].

An example of such technology is the **Livox Mid-360**, a hybrid solid-state LiDAR optimized for robotics and SLAM. It offers a full 360° horizontal field of view and employs a non-repetitive scanning pattern, making it highly effective for real-time mapping and obstacle avoidance in autonomous systems. Its balance of compact size, robustness, and dense perception capability makes it a suitable choice for research and mobile robotic applications [29].

3.3 Spherical Robot Types

Spherical robots are well-known for their unique design compared to traditional wheeled or legged robots. They have various ways to be designed for locomotion as described in Section 2.2. In this section, we will describe the most common types of spherical robots based on their actuation mechanisms which can be broadly categorized into : **Barycentric** , **Conservation of Angular Momentum** , **rod-driven** and **Shell deformation**.

3.3.1 Barycentric

Barycentric spherical depends on the displacement of the center of mass to achieve rolling motion. This is achieved by using internal wheels or other mechanisms to shift the center of mass, allowing the sphere to roll in the desired direction. [30]

Hamester-driven

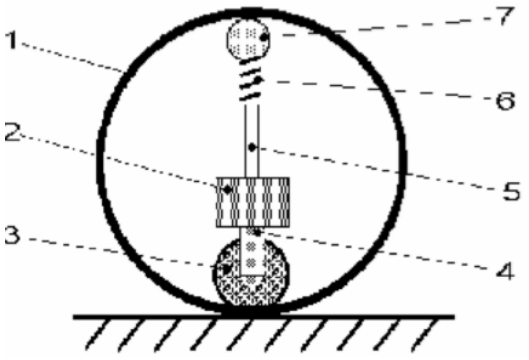
one of the earliest form of barycenter offset locomotion "hamster" design, as it mimics the movement of a hamster running inside a toy sphere. In this concept, a small wheeled robot is placed within the shell, and its weight and motion generate the force needed to roll the structure. Different internal vehicles can be used, such as single- or multi-wheeled robots, with four-wheel differential-drive systems offering additional maneuverability and even the ability to turn in place. The design is relatively simple to build and operate, but challenges arise from wheel slippage, friction losses, and instability when the internal robot becomes airborne over uneven surfaces. These issues can lead to loss of traction, reduced momentum, and tracking errors, limiting its effectiveness for tasks requiring precise navigation. [31]

Spring-Loaded locomotion

In this design, a spring–rod mechanism is mounted on the top of the internal robot and pressed against the inner surface of the spherical shell. The spring ensures that the robot's driving wheels remain in continuous contact with the shell, providing stable traction and controlled motion. To minimize friction at the contact point, a three-degree-of-freedom (DOF) ball bearing is integrated at the end of the spring. This bearing allows smooth sliding along the inner shell surface while accommodating multi-directional movement, thus improving efficiency, reducing energy loss, and enhancing overall maneuverability within the spherical structure. [5, 31, 32]

pendulum-driven Mechanism

A pendulum-driven spherical robot is a type of *barycentric sphere robot* that relies on shifting its internal center of mass for locomotion. The robot consists of a spherical shell, a central shaft, an internal pendulum, and typically two motors (or a dual-axis torque motor), as illustrated in Fig. XX. The pendulum, mounted near the inner surface of the shell, is connected to the center of the sphere and actuated by a motor or high-precision servo. By swinging the pendulum to the left or right, the robot shifts its center of mass, causing the shell to roll laterally. Meanwhile, the dual-axis torque motor controls forward and backward motion by controlling the pendulum's



(a) Structure of the Rolling Robot. 1. robot body (case), 2. controlling box, 3. driving wheel, 4. steering axis, 5. supporting axis, 6. spring, 7. balance wheel [32]

Figure 3.3: Spring-loaded spherical robot actuation.

orientation. This design provides a relatively simple and energy-efficient method of achieving mobility, though its torque is limited by the pendulum's displacement and gravitational force.[8, 10, 33]

3.3.2 Conservation of Angular Momentum

Instead of moving the center of mass for the rolling motion, the motion of this sphere is achieved by using reaction wheel. We will focus on the flywheel-driven mechanism, which is a specific implementation of the conservation of angular momentum principle.[30]

Flywheel-driven

The flywheel mechanism in spherical robots operates on the principle of conservation of angular momentum (COAM), where a rapidly spinning internal rotor generates torque to drive motion. By accelerating, decelerating, or reorienting the flywheel, reaction torques are imparted to the spherical shell, enabling controlled movement. More advanced implementations use Control Moment Gyroscopes (CMGs), in which a high-speed flywheel mounted on a gimbal produces torque in an orthogonal axis when rotated, thus amplifying output power. One of examples that follow this mecahnism is L.U.N.A[16] via the Impulse by Conservation of Angular Momentum (IBCOAM) method.

Compared to barycenter-offset systems, flywheel-based designs are not limited by the internal center of mass and can achieve higher torques, improved agility, and even holonomic motion with multi-axis configurations. However, these systems introduce challenges such as precession-induced instability, nonlinear dynamics, and high energy demands, requiring sophisticated control strategies for stable and precise operation.[31]

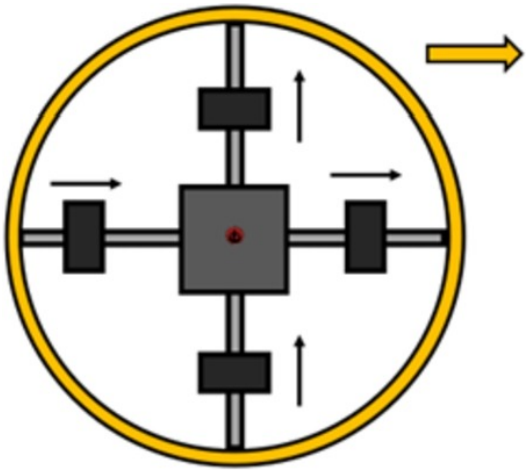


Figure 3.4: Flywheel-based simplified 2D top view of one example of flywheel mechanism, where the flywheel's angular momentum is used to generate torque for movement. [5]

3.3.3 Rod-Driven Mechanism

The rod-driven mechanism for spherical robots generates locomotion by extending and retracting internal rods positioned within the sphere. There are two primary ways in which these rods can initiate rotation. The first is the *pushing approach*, where rods on the side opposite to the intended rolling direction extend outward to make contact with the ground. By exerting force against the surface, they generate a reaction torque that propels the sphere in the desired direction. The second method is the *leverage approach*, in which the weight of the rods themselves is used to shift the robot's center of mass. When extended without contacting the ground, the displaced weight creates torque that induces rotation. In practice, a combination of pushing and leverage is often employed, where the rods provide both active ground interaction and passive weight shifting, resulting in more efficient and controllable locomotion. However, experiments have shown several challenges: on soft or sandy surfaces the mechanism struggles to generate enough torque due to actuator limitations, while on solid ground poorly chosen extension angles can lead to unstable or oscillatory motion. Moreover, leverage alone often fails to move the robot effectively, and even combined push-and-leverage control may cause stalling or uncontrolled rolling, requiring manual intervention. [19]

3.4 Introduction to SLAM

Simultaneous Localization and Mapping (SLAM) addresses the problem of enabling a robot to navigate in an unknown environment while building a consistent map of that environment. A robot typically starts at an initial position and moves through an environment with uncertain motion dynamics. Over time, this uncertainty grows, making it increasingly difficult for the robot to accurately estimate its pose in global coordinates.

To overcome this challenge, the robot relies on its onboard sensors—such as cameras, LiDAR, radar, and IMU—which are inherently affected by noise. Using these noisy measurements, the

robot incrementally builds a representation of the environment (a map) while simultaneously estimating its position relative to that map in real time [34].

Researchers have developed a variety of SLAM algorithms to address this problem with improved accuracy, robustness, and computational efficiency. Among the widely studied methods are *Visual-Inertial Odometry (VIO)*, *LiDAR-Inertial Odometry (LIO)*, and their hybrid variant *Visual-LiDAR-Inertial Odometry (VLIO)*. These approaches form the foundation of many modern SLAM systems.

3.4.1 Visual-Inertial Odometry (VIO)

VIO fuses visual data (from monocular, stereo, or RGB-D cameras) with inertial data from an IMU. The visual input provides rich environmental features, while the IMU provides high-frequency motion estimates [35, 36].

Some of its advantages include:

- Works well in texture-rich environments.
- IMU compensates for fast motions and reduces scale ambiguity in monocular systems.
- More lightweight than LiDAR-based methods.

However, VIO also has some limitations:

- Struggles in low-light or feature-poor environments.
- Sensitive to camera calibration and rolling-shutter effects.
- Drift can accumulate without loop closure.

3.4.2 LiDAR-Inertial Odometry (LIO)

LIO integrates LiDAR scans with IMU measurements. LiDAR provides highly accurate geometric information about the surroundings, while the IMU contributes motion constraints. By combining these two sources of information, LIO can estimate the robot's pose more accurately and robustly, even in challenging environments [13, 37].

Some of its advantages include:

- Good performance in environments with poor lighting or visual features.
- Provides accurate 3D geometry and range data.
- Less sensitive to environmental changes compared to vision-based methods.

However, LIO also has some limitations:

- LiDAR sensors are expensive and power-hungry.
- Struggles in featureless areas (e.g., tunnels with uniform walls).
- Lower spatial resolution compared to cameras.

3.4.3 LiDAR-Inertial-Visual Odometry (LIVO)

LIVO combines the strengths from VIO and LIO by integrating data from both vision and LiDAR, together with IMU. Visual features improve robustness in environments where LiDAR degeneracy occurs, while LiDAR provides reliable geometry where vision fails (e.g., in poor lighting) [14, 38].

Some of its advantages include:

- Strong robustness by exploiting complementary sensor modalities.
- Reduced degeneracy problems (both visual and LiDAR weaknesses are compensated).
- Often achieves state-of-the-art accuracy in SLAM benchmarks.

However, VLIO also has some limitations:

- High computational complexity and sensor fusion challenges.
- Requires precise extrinsic calibration among LiDAR, camera, and IMU.
- Higher hardware cost and system integration complexity.

3.5 Control Strategies

Control strategies play a crucial role in various robotic applications and spherical robots are no exception. These strategies are essential for achieving precise motion control, stability, and adaptability to dynamic environments. There are two types of control strategies: open-loop and closed-loop control. An **open-loop system** relies solely on the system's model, without input from past performance or supporting measurements. In contrast, a **closed-loop system** uses feedback—such as errors or results from previous operations—to continuously adjust and improve its performance. Fig. 3.5 illustrates the difference between the two systems.[6, 7]

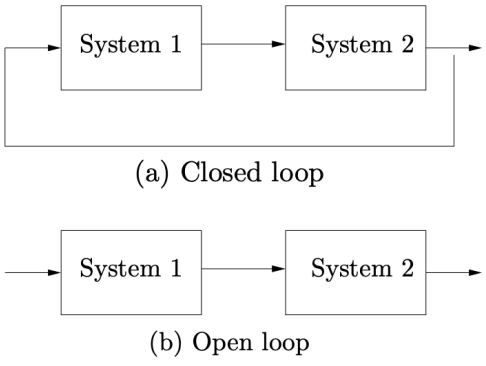


Figure 3.5: Comparison of closed-loop and open-loop control systems.[6]

In this section, we will discuss two widely used control strategies: Proportional-Integral-Derivative (PID) control and Linear Quadratic Regulator (LQR) control. These methods are particularly relevant for spherical robots, where maintaining balance and achieving accurate movement is critical since they are designed to roll in all directions.

3.5.1 PID Control

Proportional–integral–derivative (PID) control is one of the most widely used methods of implementing feedback (closed-loop systems) in applications ranging from industrial processes and manufacturing to simple balancing robots [6]. A PID controller combines three types of control actions: proportional, integral, and derivative. The feedback correction term is obtained by applying individual gain values to each of these control components, as illustrated in Figure 3.6.

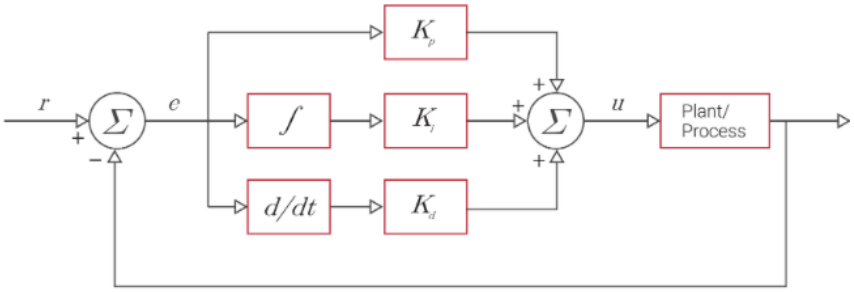


Figure 3.6: PID Controller Block Diagram. [7]

Considerable research has been devoted to determining the optimal gain values for these terms across a broad spectrum of systems.

Proportional Control (P): The proportional term applies a gain factor K_p to the instantaneous error $e(t)$, which is defined as the difference between the desired state and the measured state:

$$u_P(t) = K_p e(t) \tag{3.1}$$

Proportional control is the simplest form of feedback control to implement. However, when used alone, it generally results in a steady-state error in the system.

Integral Control (I): The integral term accumulates the error over time and scales it by a gain factor K_i:

$$u_I(t) = K_i \int_0^t e(\tau)\, d\tau \tag{3.2}$$

When combined with proportional control, the integral component effectively acts as a low-pass filter, compensating for steady-state errors. Nevertheless, excessive integration can lead to integral windup, causing the system to overshoot the target value.

Derivative Control (D): The derivative term acts on the rate of change of the error and is multiplied by a gain factor K_d:

$$u_D(t) = K_d \frac{de(t)}{dt} \tag{3.3}$$

This allows the controller to anticipate future error trends. The faster the error changes, the stronger the influence of the derivative term. When tuned appropriately, derivative control helps prevent overshoot and enhances system stability.

Combined PID Control: The total control input is obtained by summing the three components:

$$u(t) = u_P(t) + u_I(t) + u_D(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \tag{3.4}$$

This combined feedback correction term is applied to the system input—commonly referred to as the plant—to drive the dynamics toward the desired state. The gain values K_p, K_i, and K_d can be tuned to achieve the desired system response. In practice, this tuning is typically performed through iterative testing and experimentation. When implemented on digital controllers or microcontrollers, the same principles apply, but the PID algorithm must be expressed in discrete form to account for fixed sampling intervals and computational constraints, as shown in Eq. 3.5 [7, 39].

Discrete PID Control: While the continuous-time formulation of PID control (Eq. 3.4) provides the theoretical basis, most real-world applications rely on a discrete implementation, since digital controllers operate with fixed sampling intervals. The discrete form of the PID controller is given by

$$u[k] = K_p \cdot e[k] + K_i \cdot \sum_{i=0}^{k} e[i] \cdot \Delta t + K_d \cdot \frac{e[k] - e[k-1]}{\Delta t} \tag{3.5}$$

where $u[k]$ represents the control signal (e.g., the servo motor command) at the k-th time step, $e[k]$ is the error between the target angle θ_{target} and the current angle θ_{current}, and K_p, K_i, K_d are the proportional, integral, and derivative gains, respectively. The controller operates at a fixed sampling period Δt, which replaces the continuous integral and derivative operations with their discrete equivalents.

In practice, the discrete controller enables real-time implementation on microcontrollers and embedded systems. Enhancements such as introducing a deadband around the setpoint can suppress small oscillations, while resetting the integral term within this region helps prevent windup. Moreover, manual control inputs may be superimposed on the PID output, and the resulting command is constrained within actuator limits to ensure safe and stable operation. Compared to the continuous formulation, the discrete PID explicitly accounts for sampling and computational constraints, making it more suitable for digital hardware implementations

3.5.2 Linear Quadratic Regulator (LQR)

The *Linear Quadratic Regulator* (LQR) is considered one of optimal control theory, particularly notable for its analytical tractability in stabilizing linear time-invariant systems with a quadratic cost functional. For a system modeled in state-space form as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u},$$

the infinite-horizon cost to be minimized is taken as

$$J = \int_0^\infty \left(\mathbf{x}^T \mathbf{Q}\, \mathbf{x} + \mathbf{u}^T \mathbf{R}\, \mathbf{u} \right)\, dt,$$

where $\mathbf{Q} = \mathbf{Q}^T \succeq 0$ and $\mathbf{R} = \mathbf{R}^T \succ 0$ are the state and control weight matrices [40].

By assuming the optimal cost-to-go function $J^*(\mathbf{x})$ is quadratic, $J^*(\mathbf{x}) = \mathbf{x}^T\mathbf{S}\mathbf{x}$, and substituting this into the Hamilton–Jacobi–Bellman condition, one derives an algebraic Riccati equation:

$$\mathbf{A}^T\mathbf{S} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} + \mathbf{Q} = \mathbf{0}.$$

The solution $\mathbf{S}$ subsequently yields the optimal state-feedback law

$$\mathbf{u}^*(\mathbf{x}) = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}\,\mathbf{x} = -\mathbf{K}\mathbf{x},$$

where $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}$ is the gain matrix [40].

LQR's appeal lies in its closed-form solution, robustness guarantees, and minimal reliance on computational resources, making it especially valuable as a baseline controller in robotic systems. Recent studies have shown that actuated spherical robots increasingly adopt LQR-based control rather than classical PID approaches, since LQR can explicitly handle multi-variable state coupling and achieve smoother stabilization. This researches highlights LQR's suitability for nonlinear robotic platforms where traditional PID controllers face some limitations[9].

Chapter 4

Hardware Design

This chapter describes the hardware design and structural implementation of two spherical robots developed for 3D mapping: the non-actuated sphere and the actuated sphere.

4.1 Non-Actuated Sphere

4.1.1 Design Considerations

The non-actuated sphere was modeled using CAD software, drawing inspiration from the design by Arzberger et al. [1]. Its structure features two flat discs stacked vertically and secured with pillar screws, leaving a gap of approximately 28 mm between them for internal components.

The diameter of the discs was calculated using the Pythagorean theorem 4.1 to ensure proper spacing for adding all internal components needed.

$$d = \sqrt{h^2 + r^2} \tag{4.1}$$

where d is the disc diameter, h is the vertical spacing between the discs, and r is the horizontal offset required to enclose internal components symmetrically.

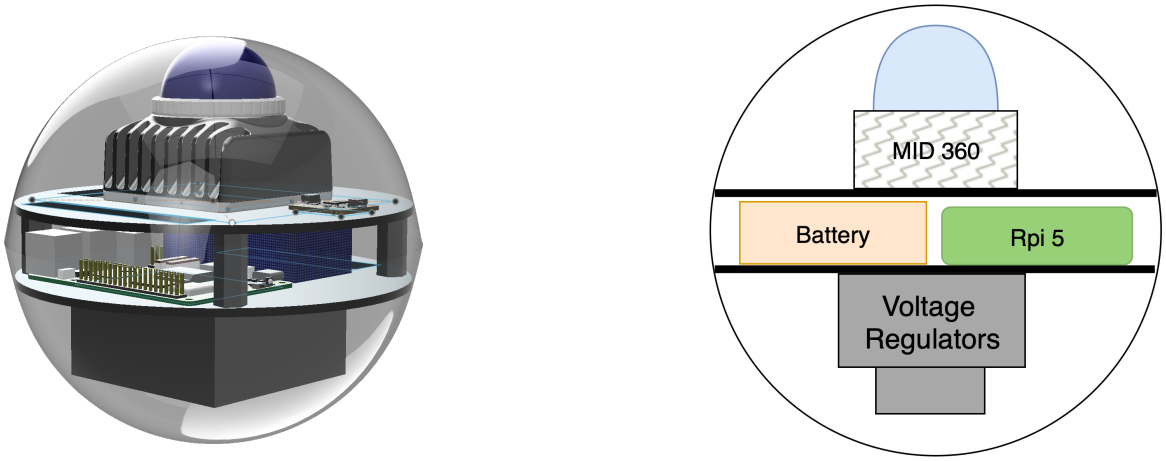
To enhance structural integrity and safety, fillets were applied to the edges of the discs. Each disc has a uniform thickness of 3 mm. The overall external diameter of the sphere is 16 cm, making it compact and lightweight (less than 1 kg, including battery and electronics). A key requirement was achieving stable and predictable rolling behavior. To this end, the center of mass was aligned with the geometric center of the sphere by strategically placing small metal weights. This alignment prevented wobbling and ensured smooth motion when the sphere was manually pushed by hand or foot.

4.1.2 Fabrication Methods

Two fabrication approaches were explored for the discs:

- **3D Printing with PLA,** which enabled rapid prototyping and iterative testing.
- **Laser Cutting with Acrylic,** which provided greater rigidity.

Both materials demonstrated sufficient durability during initial testing.



(a) 3D CAD model **(b)** Simplified 2D Model

Figure 4.1: Schematic model and design of the non-actuated sphere. It must be pushed manually.

Table 4.1: Hardware components and placement in non-actuated sphere

Layer	Components
Top	Livox Mid-360 LiDAR, BNO-085 IMU
Middle	Raspberry Pi 5 (16 GB, 256 GB SSD, cooling fan)
	2200 mAh 3S LiPo battery
Bottom	12V 10A voltage regulator
	5V 5A voltage regulator (for Raspberry Pi 5)

4.1.3 Component Integration

The internal layout of the non-actuated sphere was divided into three layers (Table 4.1). Components were arranged to balance weight distribution and minimize cable routing complexity.

Table 4.1 summarizes the components used in the non-actuated sphere and their locations within the sphere.

Each component was carefully selected to ensure its compatibility with the sphere's design and operational requirements:

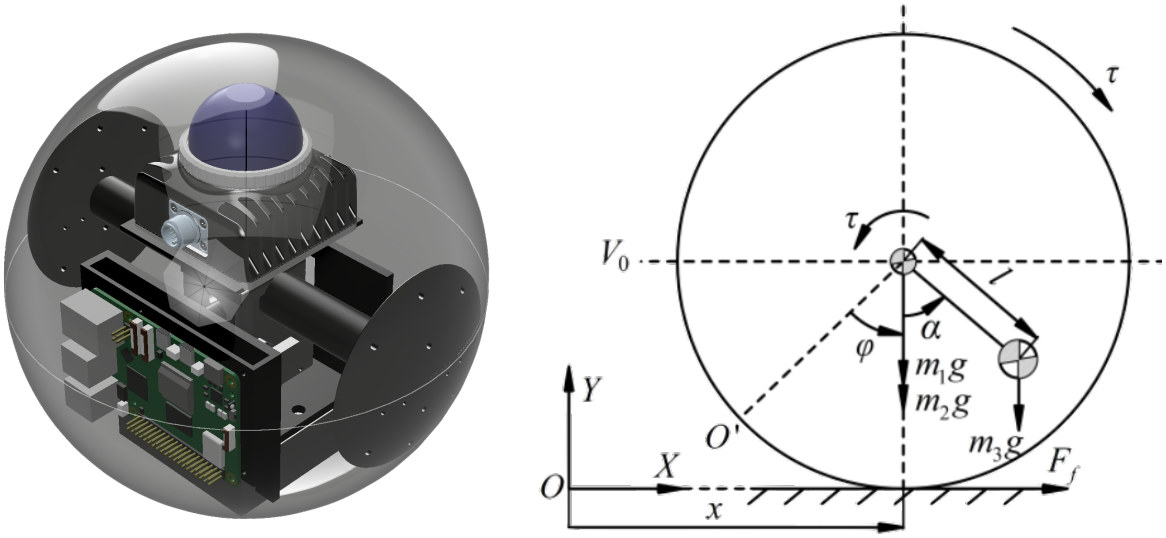
- **Livox Mid-360 LiDAR:** was mounted on the top layer to maintain an unobstructed 360° field of view.
- **Raspberry Pi 5:** was placed in the middle layer, providing sufficient processing power for real-time SLAM.
- **LiPo Battery:** a 2200 mAh 3S battery was used to power the system, ensuring a balance between weight and runtime.

- **Voltage Regulators:** a 12V 10A voltage regulator was used to power the LiDAR, while a 5V 5A regulator powered the Raspberry Pi 5.

The Livox Mid-360 is connected to the Raspberry Pi via Ethernet, while the BNO-085 IMU communicates through the I2C interface.

A 3D CAD model of the assembled structure is shown in Fig. 4.1. Simple 2D Drawings of the sphere with dimensions are shown in Fig. 4.1b.

The complete structural design of the non-actuated sphere is shown in section 8.1.1. For reproducibility, the complete CAD and 2D design files are made openly available in our GitHub repository [41].



(a) 3D CAD model **(b)** Pendulum model [42]

Figure 4.2: Schematic model and design of the actuated sphere. The sphere uses a servo motor for actuation through pendulum-driven locomotion. A second servo shifts the center of mass laterally by moving the battery to enable curved trajectories.

4.2 Actuated Sphere

4.2.1 Design Considerations

The actuated sphere was designed using CAD software, with its final structure illustrated in Fig. 4.2. As described in Section 2.2, the design was inspired by pendulum-driven locomotion mechanisms reported in related works [8, 9].

Unlike the non-actuated sphere, this version have an internal actuation system that enables autonomous locomotion. At its core, a **Waveshare Smart Continuous Servo (ST3215)** provides forward and backward motion by dynamically shifting the internal center of mass. A second **PWM-controlled servo** produces lateral displacement of the pendulum, enabling left and right turns.

The simplified dynamic model of the robot, shown in Fig. 4.2b, captures the essential motion parameters:

- robot horizontal position x,
- shell rotation angle ϕ,
- pendulum swing angle α,
- applied motor torque τ.

The physical properties considered in the design include:

- shell mass m_1,
- driver mass m_2,
- counterweight mass m_3,
- gravitational acceleration g,
- pendulum rod length l.

These parameters guided the selection of servo specifications and overall component placement to balance actuation efficiency with structural stability.

4.2.2 Fabrication Method

The structural components were fabricated from **PLA filament**, produced by 3D printing in multiple sections and assembled into the final spherical form. PLA was selected for this prototype due to its light weight, ease of fabrication, and sufficient strength for housing the actuation system.

4.2.3 Component Integration

The internal components were distributed across specific positions within the sphere, as summarized in Table 4.2. Placement was chosen to maximize balance and minimize unnecessary torque on the servos:

- The **primary servo motor** was centrally mounted, ensuring stable pendulum motion.
- The **PWM servo** was aligned laterally to allow controlled sideward displacement.
- The **LiPo battery** was placed at the rear.
- The **counterweight** was positioned at the front to balance the mass of the battery.
- Voltage regulators were distributed between the pendulum module and rear compartment to reduce thermal buildup and electrical noise.
- Sensors were mounted on the top for clear environmental perception (LiDAR and Pi Camera).

The detailed CAD drawings of the actuated sphere are presented in 8.1.2. These include orthographic projections, component placement, and disc dimensions, providing a complete overview of the mechanical layout.

Table 4.2: Hardware components and placement in the actuated sphere

Position	**Components**
Center	Waveshare Smart Continuous Servo (model ST3215)
	Forward/backward actuation
	PWM Servo – left/right pendulum control
Rear	2200 mAh 3S LiPo battery
	5V 5A voltage regulator (for Raspberry Pi 5)
	6V UBEC (for servo motors)
Pendulum Module	12V 20A voltage regulator – powers the entire system; positioned near the shell for stability
Front	Raspberry Pi 5 (16 GB model)
Top	Livox Mid-360 LiDAR
	Pi Camera V3 (12 MP)

Chapter 5

Software Design

As shown in the system state diagram in Fig. 5.1, both the actuated and non-actuated spheres follow the same state transitions, with the exception of the control subsystem, which is unique to the actuated variant. All software components are deployed on Ubuntu 24.04.2 LTS ARM and ROS2 Jazzy.

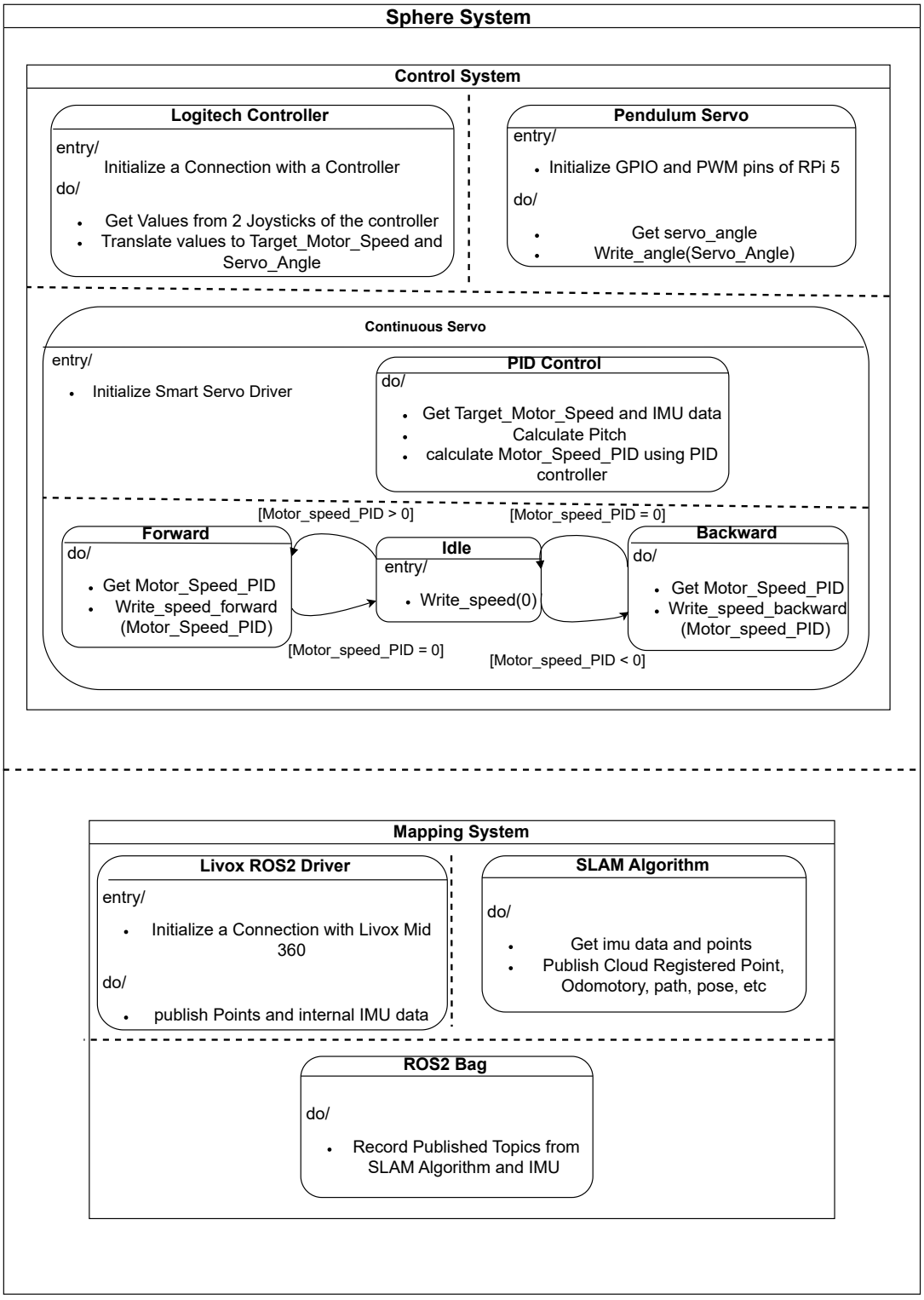


Figure 5.1: State chart of the Sphere System showing the control and mapping subsystems.

5.1 Real-time Performance Optimization

To ensure getting the most out of the Raspberry Pi 5, we applied several performance optimizations by using cooling fans on the Raspberry Pi and taking advantage of the PCIe bus to connect an NVMe HAT for improved data throughput.

One of the difficulties we faced was the limiting CPU performance of the Raspberry Pi 5 when not using the official 5V 5A power supply, as no official portable version exists. The official power supply is required to run the Raspberry Pi 5 at full performance [43]. The Raspberry Pi 5 requires a minimum of 5V at 5A to operate at maximum CPU frequency and avoid performance throttling.

To address this, we used a 5V 5A voltage regulator to power the Raspberry Pi 5, and disabled the power delivery (PD) protocol to allow for higher power draw. This modification enabled the system to maintain peak performance during intensive SLAM computations while operating on battery power within the spherical platform.

5.2 Mapping System

Both spheres incorporate a real-time mapping system. The following LiDAR-Inertial Odometry (LIO) frameworks were evaluated and integrated:

- **FAST-LIO2**
- **DLIO**
- **FAST-LIVO2** (LIO-only mode for real-time mapping)

Some packages such as `FAST-LIVO2` needed modification to meet the constraints and processing capabilities of the Raspberry Pi 5 and ROS2 Jazzy. The mapping system is designed to operate in real-time, processing LiDAR scans and IMU data published from the Livox Mid-360 and BNO-085 sensors, respectively. IMU data was collected from both the BNO-085 and Livox Mid-360, but only the Livox IMU data was used for mapping, as it is more reliable, which will be discussed in Section 6. The mapping system is implemented with several bash scripts that orchestrate the mapping pipeline, which processes the data using the selected LIO framework and publishes the resulting map in ROS bag format for later evaluation and comparison against ground truth data. The source code, configuration files, and instructions for building the software are available in our GitHub repository [41]. All mapping computations are performed onboard in real-time.

Figures 5.2, 5.3, and 5.4 show the Rqt graphs of the mapping system for each LIO framework, respectively. Tables 5.1, 5.2, 5.3, and 5.4 summarize the key ROS2 topics used in the mapping system for each framework and Point-Cloud visualization.

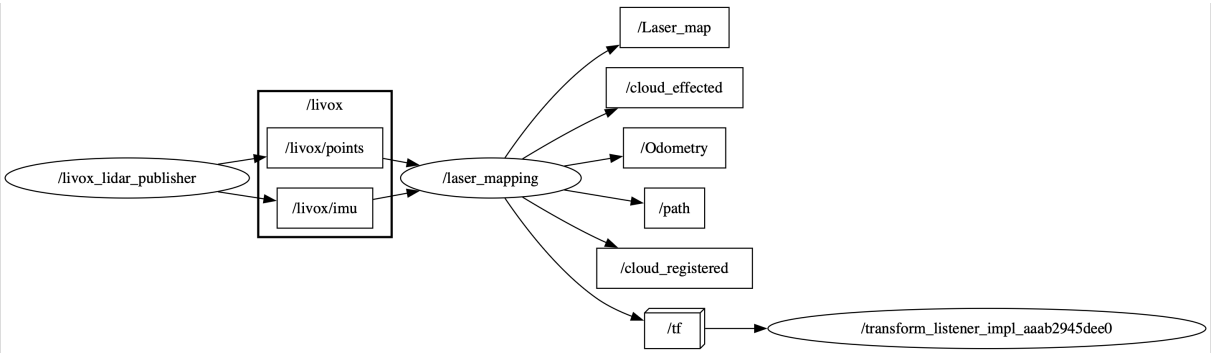


Figure 5.2: Mapping System Rqt Graph(FAST-LIO2)

Table 5.1: Livox ROS2 Topics

Topic	Publisher	Subscribers	Type	Description
`/livox/imu`	`livox_lidar_publisher`	`waveshare_servo_node`	`sensor_msgs/Imu`	IMU data from internal IMU of Livox Mid-360
`/livox/points`	`livox_lidar_publisher`	`laser_mapping`, `dlio_odom_node`	`sensor_msgs/PointCloud2`	Point cloud data from Livox Mid-360 LiDAR

Table 5.2: FAST-LIO2 ROS2 Topics

Topic	Publisher	Subscribers	Type	Description
`/Odometry`	`laser_mapping`	rviz or rosbag	`nav_msgs/Odometry`	publishes pose (position + orientation) and velocity of the robot
`/cloud_registered`	`laser_mapping`	rviz or rosbag	`sensor_msgs/PointCloud2`	PointCloud data output from FAST-LIO2
`/path`	`laser_mapping`	rviz or rosbag	`nav_msgs/Path`	Visualizes the full trajectory of the robot
`/tf`	`laser_mapping`	rviz or rosbag	`tf2_msgs/TFMessage`	Publishes the robot's pose as a transform between frames (body -> camera_init)

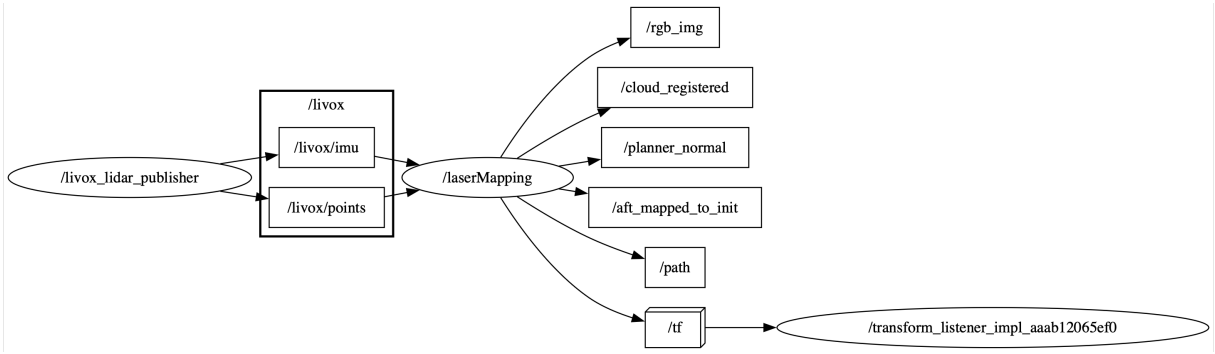


Figure 5.3: Mapping System Rqt Graph(FAST-LIVO2)

Table 5.3: FAST-LIVO2 ROS2 Topics

Topic	Publisher	Subscribers	Type	Description
`/Odometry`	`laserMapping`	rviz or rosbag	`nav_msgs/Odometry`	publishes pose (position + orientation) and velocity of the robot
`/cloud_registered`	`laserMapping`	rviz or rosbag	`sensor_msgs/PointCloud2` `livox_ros_driver2/CustomMsg`	PointCloud data output from FAST-LIVO2
`/path`	`laserMapping`	rviz or rosbag	`nav_msgs/Path`	Visualizes the full trajectory of the robot
`/tf`	`laserMapping`	rviz or rosbag	`tf2_msgs/TFMessage`	Publishes the robot's pose as a transform between frames (aft_mapped -> camera_init)

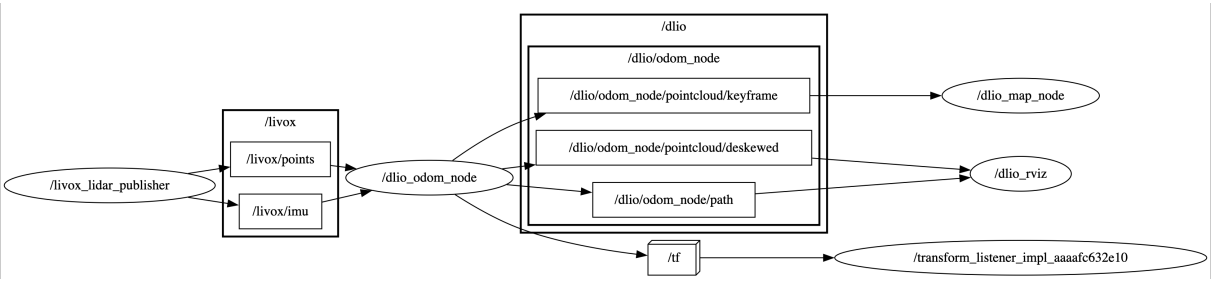


Figure 5.4: Mapping System Rqt Graph(DLIO)

Table 5.4: DLIO ROS2 Topics

Topic	Publisher	Subscribers	Type	Description
`/dlio/odom_node/pointcloud/deskewed`	`dlio_odom_node`	rviz or rosbag	`sensor_msgs/PointCloud2`	PointCloud data output from DLIO
`/dlio/odom_node/path`	`dlio_odom_node`	rviz or rosbag	`nav_msgs/Path`	Visualizes the full trajectory of the robot
`/tf`	`dlio_odom_node`	rviz or rosbag	`tf2_msgs/TFMessage`	Publishes the robot's pose as a transform between frames (odom -> base_link)

5.3 Actuator System (Actuated Sphere Only)

The actuated sphere features an internal movement control system, operated using a Logitech F710 controller. The controller's two joysticks are mapped to distinct control tasks:

- **Servo control:** One joystick provides input to the continuous rotation servo. These inputs are scaled and passed through a discrete Proportional-Integral-Derivative (PID) controller to regulate pitch by minimizing the error between the desired target angle and the current angle measured by the IMU.
- **Mass shifting:** The second joystick adjusts the angle of an internal weight to shift the center of mass left or right, enabling directional control of the sphere's rolling behavior.

This control strategy allows for fine-grained movement and stabilization, combining feedback-based servo control with physical mass displacement for maneuvering.

The control system is implemented using three separate ROS2 packages but they are in same workspace, each responsible for a specific aspect of the control logic:

- `waveshare_servo`: Implements the PID control logic for the continuous servo motor and manages forward/backward movement.

- `logitech_controller`: Handles joystick input from the Logitech F710 controller, translating joystick movements to control commands for pwm servo and waveshare servo operation and mass shifting.
- `servo_pwm_node`: Manages PWM signal generation for the servo motor using wiringPi with rapid GPIO access which is responsible for pendulum control and mass shifting.

These packages are integrated to initialize using a single command from a master ROS package called `Khonsu`, which enables system activation with a single command.

Figures 5.5 shows the Rqt graph of the actuator system. Table 5.5 summarizes the key ROS2 topics used in the actuator system.

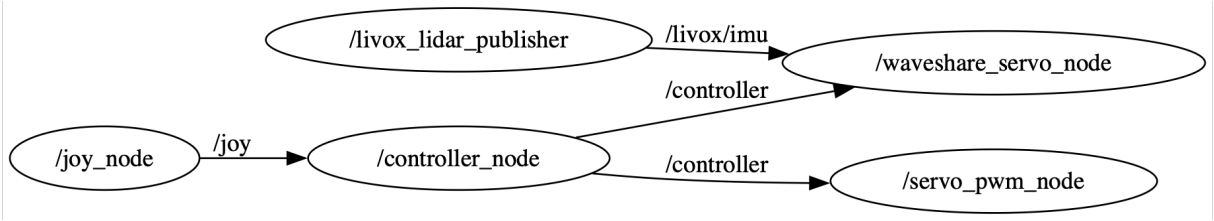


Figure 5.5: Actuator System Rqt Graph

Table 5.5: Actuator System ROS2 Topics

Topic	Publisher	Subscribers	Type	Description
`/joy`	`joy_node`	`controlled_node`	`sensor_msgs/Joy`	Joystick input data from Logitech F710 controller
`/controller`	`controller_node`	`waveshare_servo_node`, `servo_pwm_node`	`geometry_msgs/Twist`	Control commands for servo motors and mass shifting

Chapter 6

Evaluation and Results

In this Chapter, we present the evaluation and results of both spherical robots in terms of their mapping accuracy and stability. All experiments were conducted in a controlled indoor environment within the Computer Science building at the University of Würzburg.

6.1 Evaluation Setup

The evaluation consisted of mapping selected indoor areas—specifically, a corridor, a hall, and the upper floor—using both the non-actuated and actuated spherical robots. For the non-actuated sphere, the robot was manually moved by hand and by foot. Each mapping run used one of three SLAM algorithms: FAST-LIO2, DLIO, and FAST-LIVO2 (LIO mode). The experiment was repeated three times—once for each algorithm—under the same environmental conditions. The same procedure was carried out with the actuated sphere, with the key difference being that it was moved using a controller rather than manually. Again, each of the three SLAM algorithms was executed along similar paths within the same locations. We used 3DTK [44] to process the resulting point-clouds, which were evaluated by comparing them against ground truth data obtained using a Riegl VZ-400 terrestrial laser scanner (TLS) as shown in Fig. 6.1a.

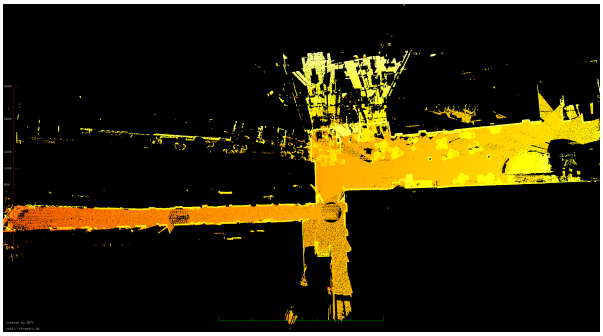
The error metric used was the point-cloud root mean square error (RMSE).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |p_i - q_i|^2} \tag{6.1}$$

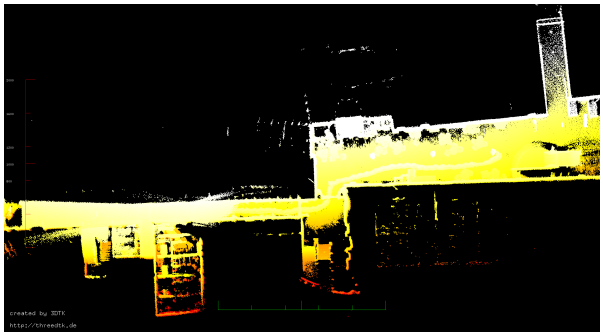
where p_i and q_i are the corresponding points in the model and data point-clouds, and N is the number of corresponding points. The RMSE measures the average distance between corresponding points in the two point clouds, providing an indication of the mapping accuracy.

(a) Riegl VZ-400 **(b)** Ground truth point-clouds

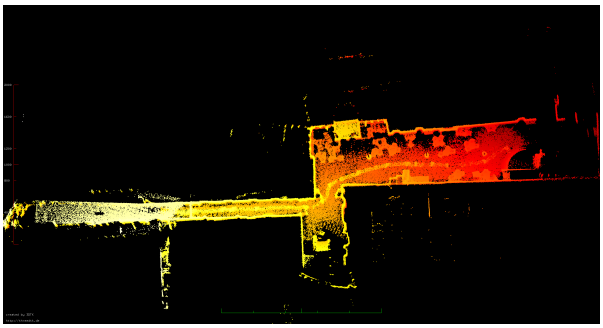
Figure 6.1: Evaluation setup, showing the Riegl VZ-400 terrestrial laser scanner (TLS) and the resulting point-cloud used as ground truth in the evaluation.



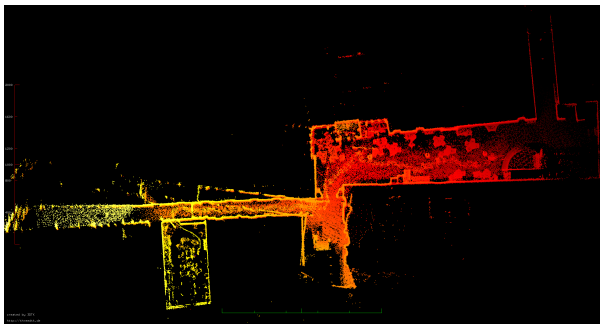
(a) Riegl Map Result



(b) Actuated FAST-LIVO2



(c) Actuated FAST-LIO2

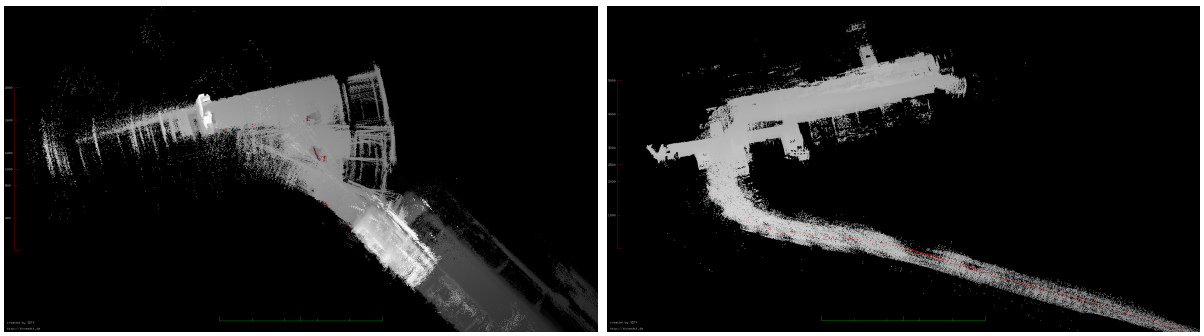


(d) Non-actuated FAST-LIO2

Figure 6.2: Bird's-eye view of a cross section of resulting point-clouds (color indicates height where red means higher).

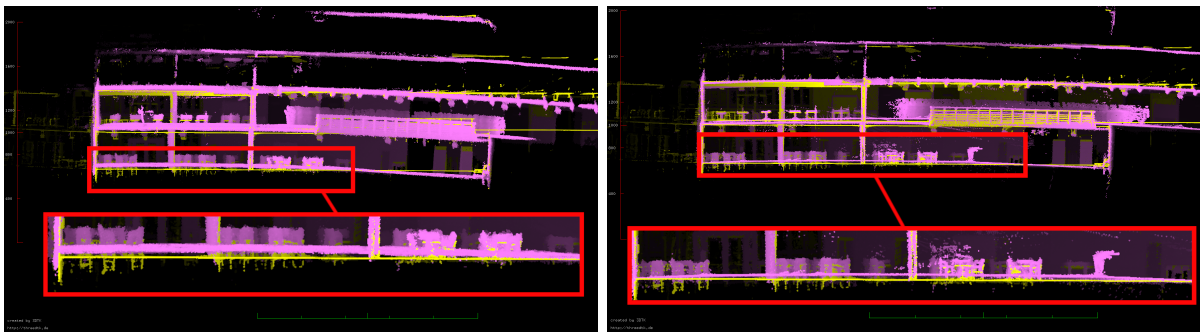
6.2 Drift and Bending

The LIO algorithms exhibited small drift and bending over time, which is particularly noticeable in Fig. 6.4. We were unable to obtain a satisfactory map using DLIO from the non-actuated sphere as shown in Fig. 6.3a. Furthermore, sometimes the algorithms drifted without recovering after a fast motion, as shown in Fig. 6.3b. We attribute this to the underlying motion models used in the LIO algorithms, which were designed for more conventional systems.



(a) DLIO point-cloud **(b)** FAST-LIO2 point-cloud

Figure 6.3: Example failed cases from the non-actuated sphere where the LIO algorithm could not recover due to fast angular motion. This results in huge drift and inconsistent mapping.



(a) DLIO point-cloud **(b)** FAST-LIVO2 point-cloud

Figure 6.4: Cross section comparison. Yellow: Ground truth, Magenta: Mentioned algorithm. The red boxes indicate noticeable bending of the ground plane.

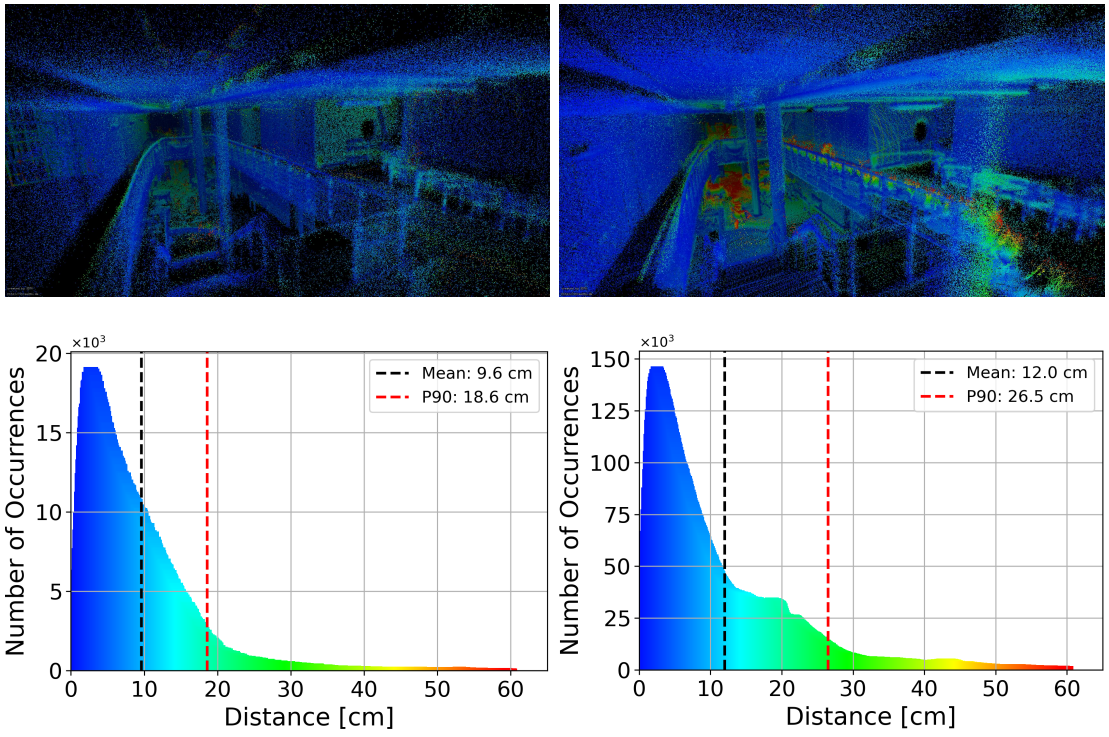
6.3 Mapping Accuracy

We evaluate the point-cloud error for each mapping run using the three SLAM algorithms. Table 6.1 summarizes the results. The statistical analysis reveals significant differences in mapping performance across configurations. The non-actuated sphere with FAST-LIO2 achieved the lowest mean error (9.60 cm) and RMSE (13.09 cm), indicating superior accuracy. This is unexpected since the motion of the non-actuated sphere is governed by larger rotations around all principal axes and more aggressive dynamics. We attribute the overall better mapping performance of the non-actuated sphere to its smaller shell radius and missing locomotion mechanism structure. Thus, we were able to place the laser scanner closer to the center of the sphere.

Table 6.1: Statistical analysis of point-cloud mapping accuracy.

Metric	FAST-LIO2		FAST-LIVO2(LIO-Mode)		DLIO	
	Non-act.	**Act.**	**Non-act.**	**Act.**	**Non-act.**	**Act.**
Points ($\times 10^6$)	1.99	4.43	16.36	46.39	-	88.01
Mean[1] **(cm)**	**9.60**	10.73	**12.05**	12.93	-	**13.71**
Std Dev (cm)	8.91	9.96	11.33	12.58	-	13.26
RMSE[1] **(cm)**	**13.09**	14.64	**16.54**	18.04	-	**19.08**
P95[2] **(cm)**	**26.76**	30.47	**37.33**	41.04	-	**42.36**
P90[2] **(cm)**	**18.58**	22.57	**26.48**	31.63	-	**34.70**

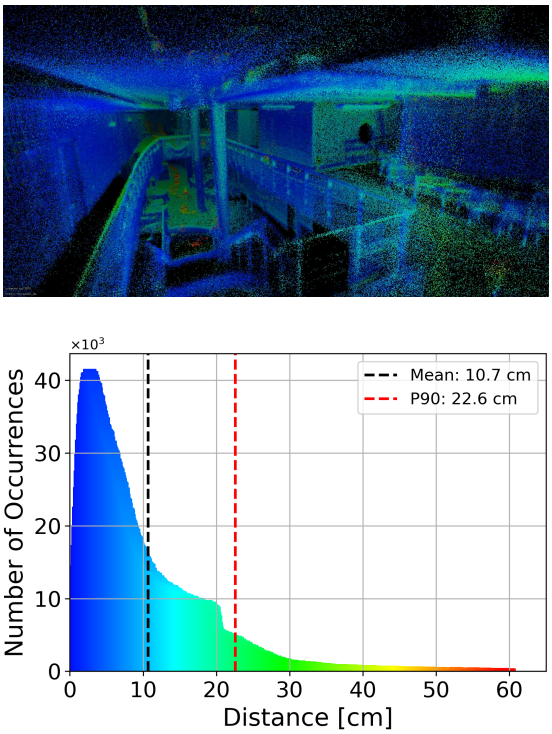
(-) denotes that the algorithm failed.
[1] Refers to the point-to-point errors to ground truth.
[2] Refers to the percent of points having point-to-point errors to ground truth smaller than the given value.



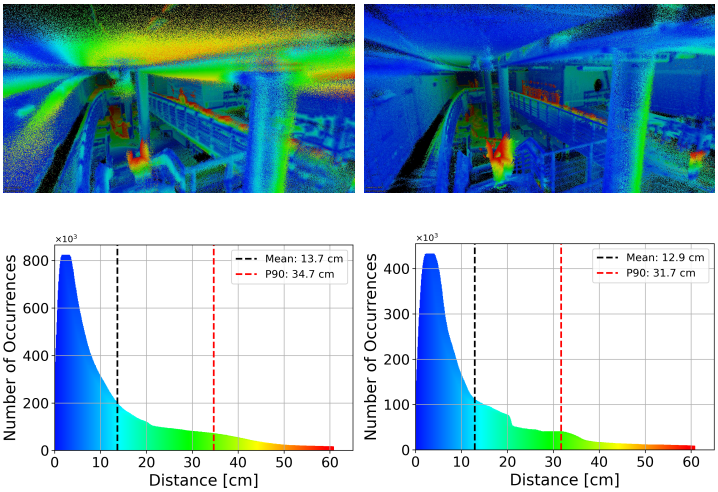
(a) Non-Act. FAST-LIO2 **(b)** Non-Act. FAST-LIVO2

(c) Color Mapping

Figure 6.5: Point-cloud Results and Error Distribution Analysis. A fly through video of the point-clouds is available at `https://youtu.be/Ere4UjPg-gk`. The first row shows the point-clouds generated by comparing the RIEGL map with each algorithm, while the second row displays the corresponding error distribution Analysis(Part 1).



(a) Act. FAST-LIO2



(b) Act. DLIO **(c)** Act. FAST-LIVO2

Figure 6.6: Error Distribution Analysis(Part 2)

Chapter 7

Conclusion

In this paper, we presented the design and evaluation of two spherical robots for 3D mapping applications: a non-actuated, and a self-actuated sphere. The self-actuated sphere uses a pendulum-based locomotion mechanism, enabling controlled movement and stabilization in addition to mapping capabilities. This is, to the best of our knowledge, the first prototype of a self-actuated spherical robot performing online LIO. The mapping accuracy of both systems was evaluated in a controlled indoor environment using ground truth point-clouds. The non-actuated sphere using FAST-LIO2 surprisingly achieved the lowest mean error and RMSE among all configurations. We attribute this to the placement of the LiDAR sensor, which is closer to the center compared to the actuated sphere. All tested algorithms produce bent point-clouds, and sometimes unrecoverable drift due to the rotationally aggressive motion of the system.

In future work, we plan to incorporate motion models into the LIO algorithms that reflect the motion of the sphere better. Furthermore, we want to explore alternative actuation mechanisms. The actuated sphere, in particular, can be developed further for autonomous navigation and exploration tasks, leveraging its enhanced locomotion capabilities. Additionally, integrating higher-resolution LiDAR sensors—such as the Robosense Airy—could improve mapping quality without increasing sphere's form factor. Pose estimation accuracy was not directly addressed in this paper, as establishing accurate ground truth trajectories would require a separate measurement setup, which was beyond the scope of this study. Overall, this work contributes to the growing field of spherical robots and provides a foundation for future advancements in autonomous 3D mapping and mobile perception.

Chapter 8

Appendix

8.1 CAD Drawings

8.1.1 Non-Actuated Sphere

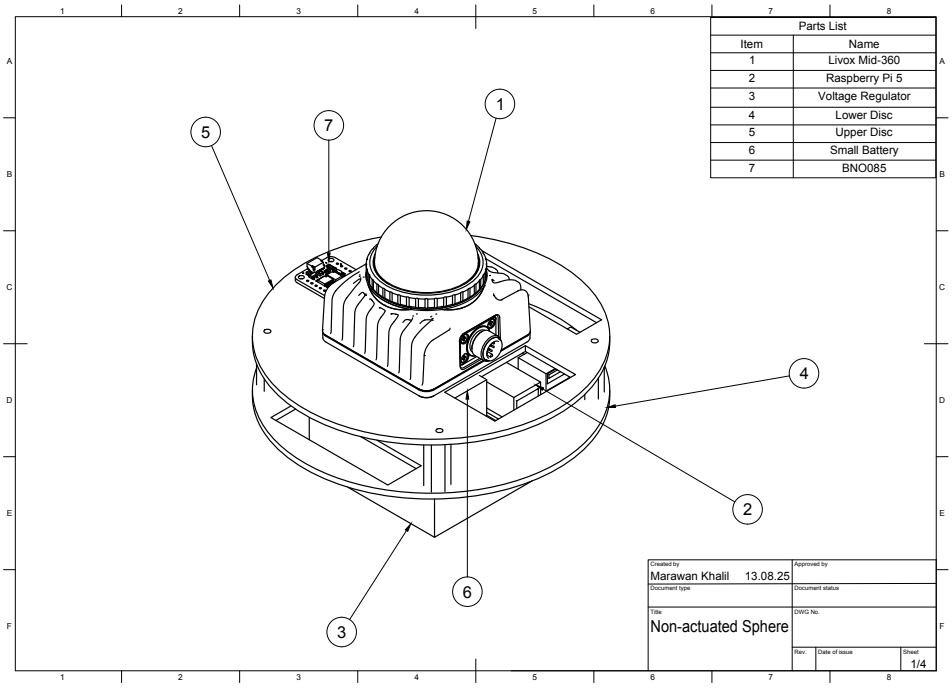


Figure 8.1: Physical design and components

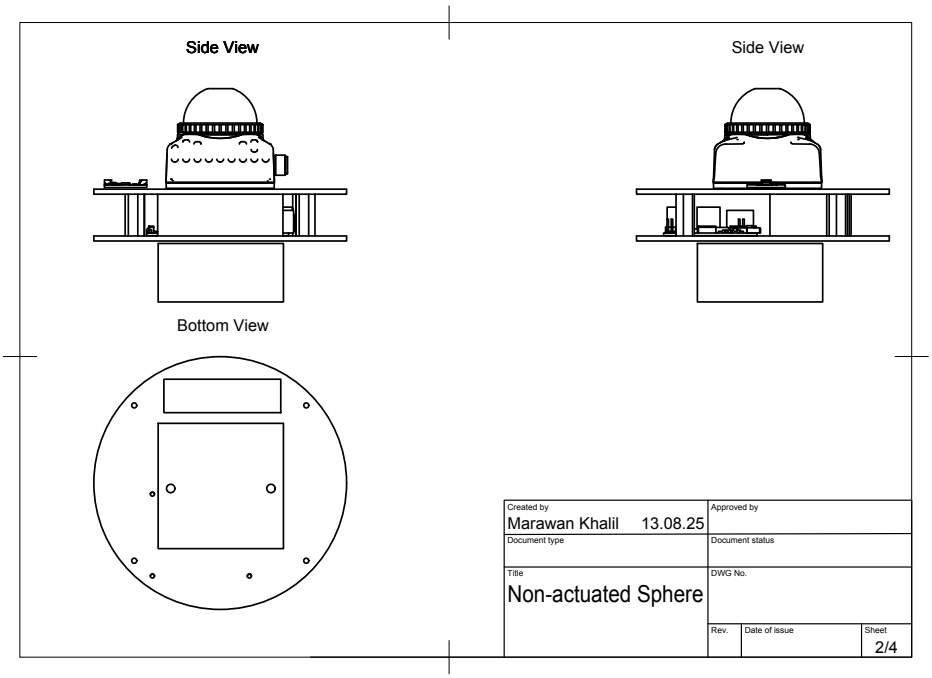


Figure 8.2: Orthographic projection of the non-actuated sphere

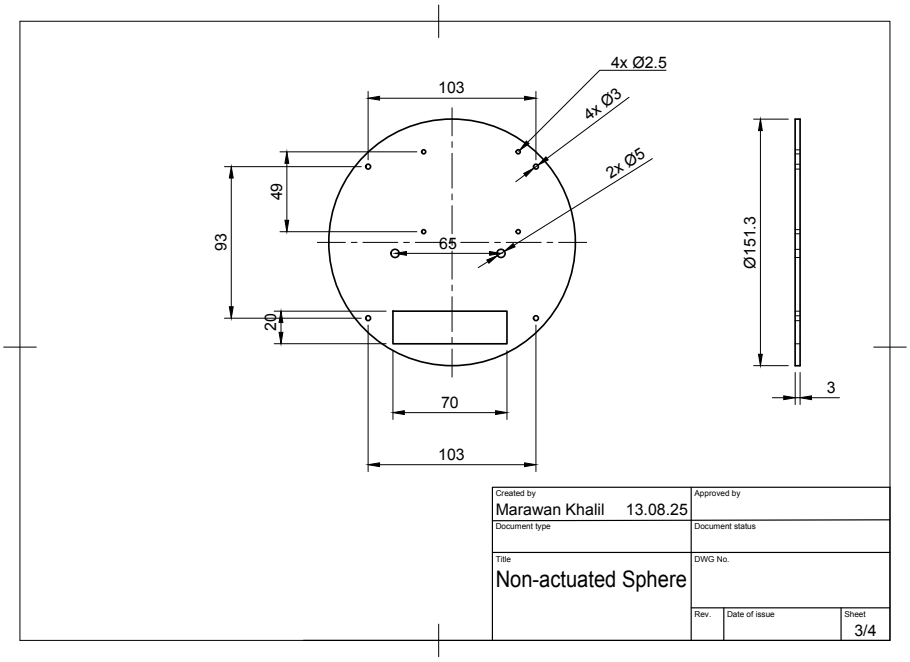


Figure 8.3: Upper Disc dimensions

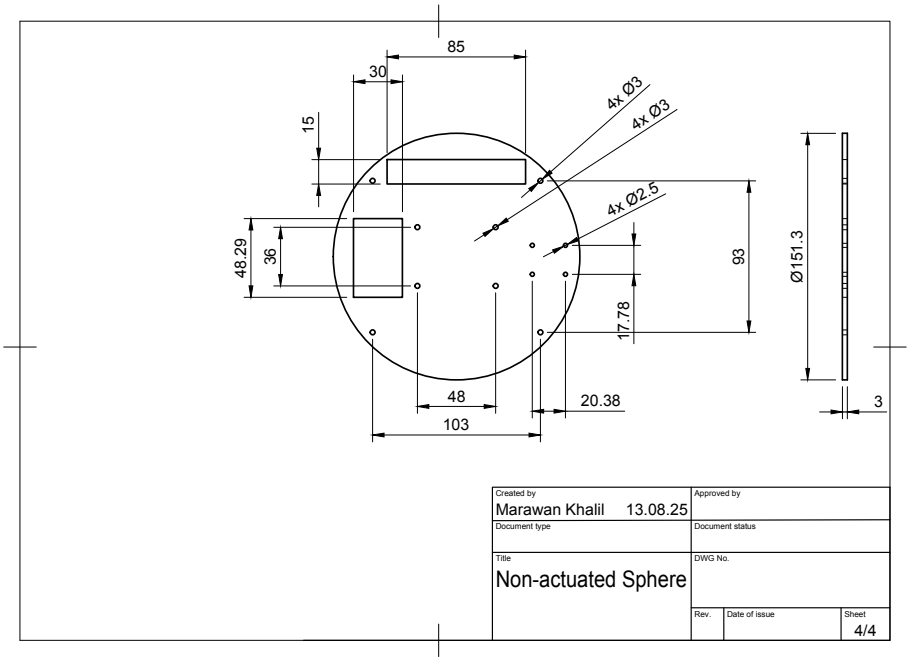


Figure 8.4: Lower Disc dimensions

8.1.2 Actuated Sphere

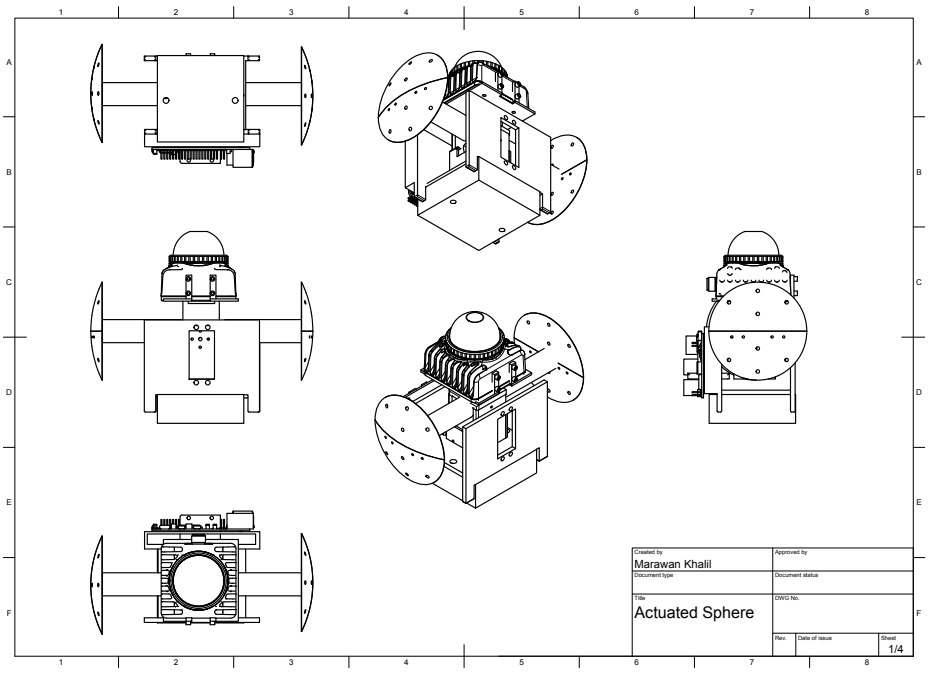


Figure 8.5: Physical design

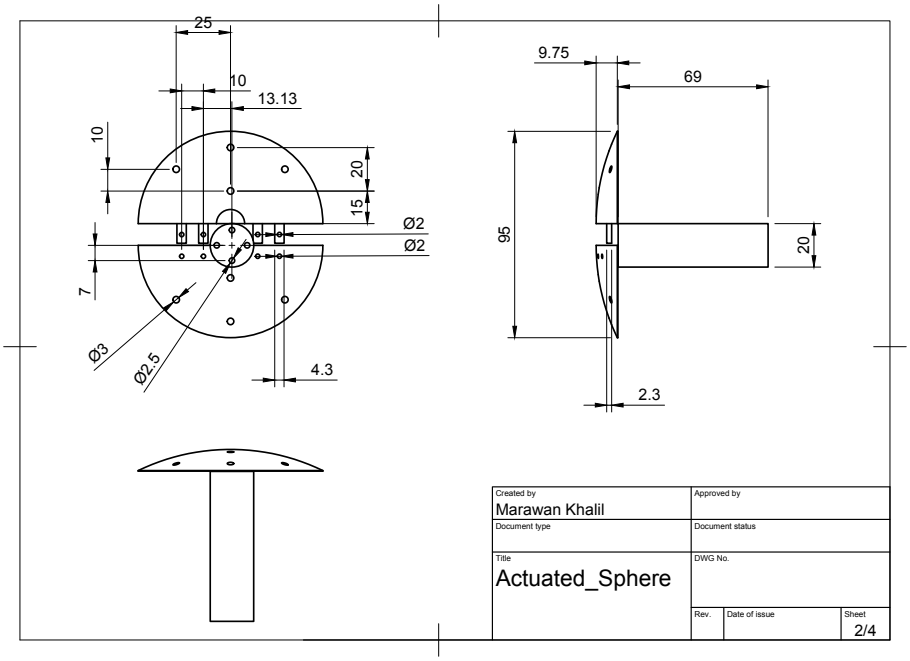


Figure 8.6: Orthographic projection of shell's hook

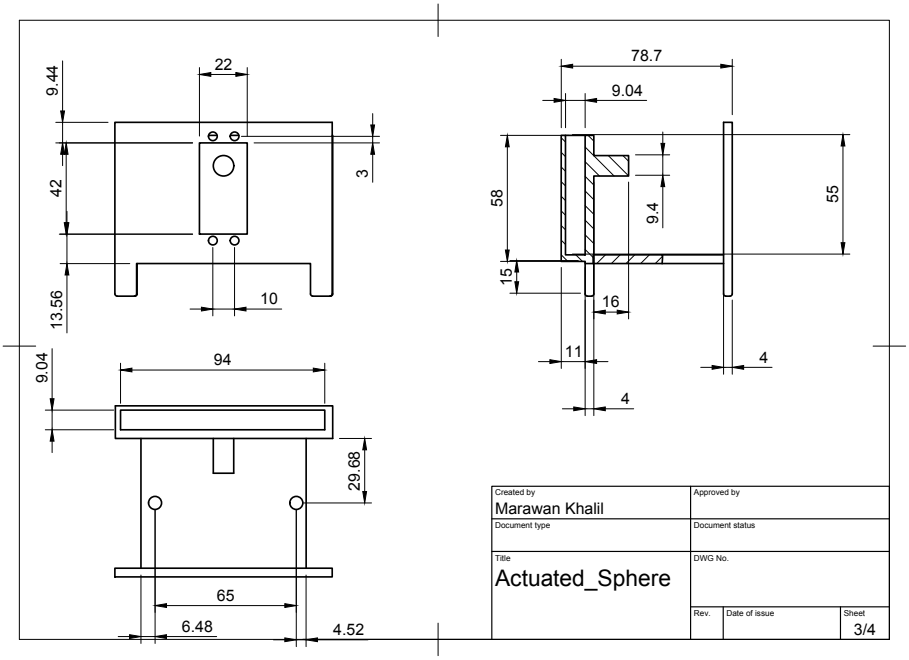


Figure 8.7: Orthographic projection of the Main body

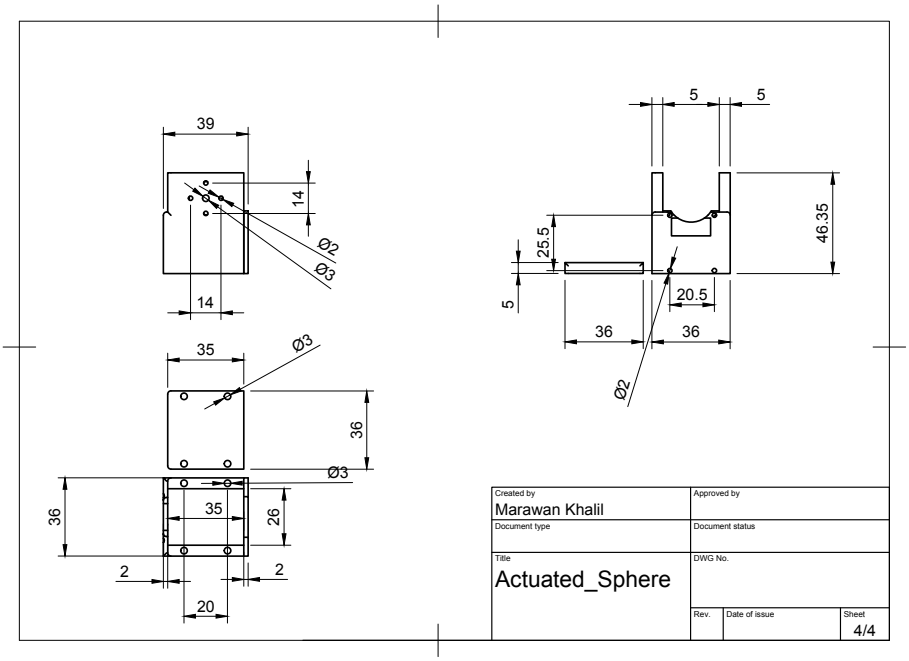


Figure 8.8: Orthographic projection of internal servo case

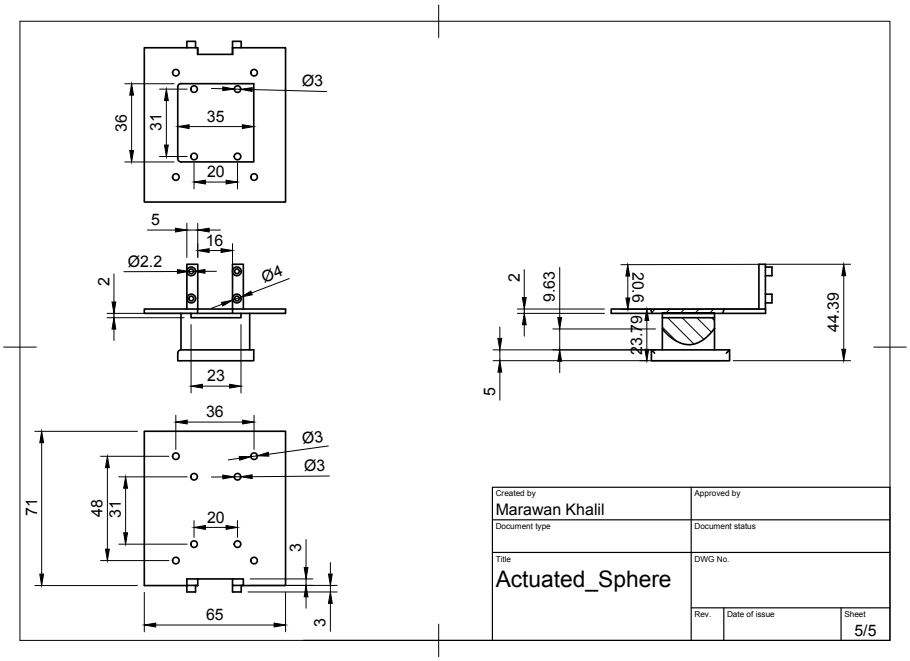


Figure 8.9: Orthographic projection of LiDar

Bibliography

[1] Fabian Arzberger, Tim Schubert, Fabian Wiecha, Jasper Zevering, Julian Rothe, Dorit Borrmann, Sergio Montenegro, and Andreas Nüchter. Delta- and kalman-filter designs for multi-sensor pose estimation on spherical mobile mapping systems. *Robotics and Autonomous Systems*, 184:104852, 2025.

[2] Fabian Arzberger, Fabian Wiecha, Jasper Zevering, Julian Rothe, Dorit Borrmann, Sergio Montenegro, and Andreas Nüchter. Delta filter - robust visual-inertial pose estimation in real-time: A multi-trajectory filter on a spherical mobile mapping system. In *2023 European Conference on Mobile Robots (ECMR)*, pages 1–8, 2023.

[3] Ericco International. Mems accelerometers for analyzing mechanical vibrations, March 2024.

[4] SparkFun Electronics. Gyroscope, n.d. Available at: `https://learn.sparkfun.com/tutorials/gyroscope/all`, accessed on 2025-05-15.

[5] Marek Bujňák, Rastislav Pirník, Dušan Nemec, Júlia Kafková, Aleš Janota, and Pavol Kuchár. Spheridrive: A spherical robot with an innovative 3-wheeled platform and omnidirectional wheels. *Results in Engineering*, 25:104351, 2025.

[6] Karl Johan Astrom and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, USA, 2008.

[7] VectorNav Technologies LLC. Math fundamentals: Controls, n.d. Available at: `https://www.vectornav.com/resources/inertial-navigation-primer/math-fundamentals/math-controls`, accessed on 2025-05-10.

[8] Micah Oevermann, Derek Pravecek, Garrett Jibrail, Rishi Jangale, and Robert O. Ambrose. Roboball: An all-terrain spherical robot with a pressurized shell. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13502–13508, 2024.

[9] Wei Ren, You Wang, Haoxiang Liu, Song Jin, Yixu Wang, Yifan Liu, Ziang Zhang, Tao Hu, and Guang Li. Spherical robot: A novel robot for exploration in harsh unknown environments. *IET Cyber-Systems and Robotics*, 5(4):e12099, 2023.

[10] Hamidreza Kolbari, Alireza Ahmadi, Mohsen Bahrami, and Farzam Janati. Impedance estimation and motion control of a pendulum-driven spherical robot. pages 6–11, 2018.

[11] Angelo Pio Rossi, Francesco Maurelli, Vikram Unnithan, Hendrik Dreger, Kedus Mathewos, Nayan Pradhan, Dan-Andrei Corbeanu, Riccardo Pozzobon, A Bredenbeck, et al. Daedalus-descent and exploration in deep autonomy of lava underground structures. 2021.

[12] Fabian Arzberger, Anton Bredenbeck, Jasper Zevering, Dorit Borrmann, and Andreas Nüchter. *Towards spherical robots for mobile mapping in human made environments*, volume 1. 2021.

[13] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry, 2021.

[14] Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, Yunfan Ren, Rong Wang, Fanle Meng, and Fu Zhang. Fast-livo2: Fast, direct lidar-inertial-visual odometry, 2024.

[15] Kenny Chen, Ryan Nemiroff, and Brett T. Lopez. Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction, 2023.

[16] Jasper Zevering, Anton Bredenbeck, Fabian Arzberger, Dorit Borrmann, and Andreas Nüchter. *L.U.N.A. - A Laser-Mapping Unidirectional Navigation Actuator*, pages 85–94. 03 2021.

[17] European Space Agency. Lunar cave explorer, 2021. Available at: `https://www.esa.int/ESA_Multimedia/Images/2021/03/Lunar_cave_explorer`, accessed on 2025-05-10.

[18] Prithvi Akella, Oliver O'Reilly, and Koushil Sreenath. Controlling the locomotion of spherical robots or why bb-8 works. *Journal of Mechanisms and Robotics*, 11, 12 2018.

[19] Jasper Zevering, Dorit Borrmann, Anton Bredenbeck, and Andreas Nuchter. The concept of rod-driven locomotion for spherical lunar exploration robots. pages 5656–5663, 10 2022.

[20] Advanced Navigation. Inertial measurement unit (imu) an introduction, n.d. Available at: `https://www.advancednavigation.com/tech-articles/inertial-measurement-unit-imu-an-introduction/#h-the-technology-behind-imu-sensors`, accessed on 2025-05-10.

[21] Stanford GPS Lab. Inertial measurement unit (imu) — development & testing, n.d. Available at: `https://gps.stanford.edu/research/current-and-continuing-gpspnt-research/inertial-measurement-unit-imu-development-testing`, accessed on 2025-05-10.

[22] A.M. Shkel and Yusheng Wang. *Pedestrian Inertial Navigation with Self-Contained Aiding*. 08 2021.

[23] Dejan. What is mems? accelerometer, gyroscope and magnetometer with arduino, 2015. Available at: `https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyrocope-magnetometer-arduino/`, accessed on 2025-05-10.

[24] VectorNav Technologies LLC. Theory: Mems operation, n.d. Available at: https://www.vectornav.com/resources/inertial-navigation-primer/theory-of-operation/theory-mems, accessed on 2025-05-10.

[25] Leah A. Wasser, NEON (National Ecological Observatory Network). The basics of lidar – light detection and ranging – remote sensing, 2024. Available at: https://www.neonscience.org/resources/learning-hub/tutorials/lidar-basics, accessed on 2025-05-10.

[26] Dr. Michael Bleier. Advanced sensory systems and sensor data processing, n.d. Available at: https://wuecampus.uni-wuerzburg.de/moodle/course/view.php?id=73665, accessed on 2025-05-10.

[27] Global GPS Systems. Explore the different types of lidar technology: From mechanical to solid-state lidar, n.d. Available at: https://globalgpssystems.com/lidar/explore-the-different-types-of-lidar-technology-from-mechanical-to-solid-state-lidar/, accessed on 2025-05-10.

[28] GovComm, Inc. Top lidar systems compared: Solid-state, mechanical & hybrid, n.d. Available at: https://govcomm.us/lidar-systems/, accessed on 2025-05-10.

[29] Livox. Mid-360 downloads (user manual, quick start guide, 3d model, firmware, software), 2025. Available at: https://www.livoxtech.com/mid-360/downloads, accessed on 2025-05-10.

[30] Aminata Diouf, Bruno Belzile, Maarouf Saad, and David St-Onge. Spherical rolling robots—design, modeling, and control: A systematic literature review. *Robotics and Autonomous Systems*, 175:104657, 2024.

[31] Richard Chase and Abhilash Pandya. A review of active mechanical driving principles of spherical robots. *Robotics*, 1(1):3–23, 2012.

[32] A. Halme, T. Schonberg, and Yan Wang. Motion control of a spherical mobile robot. In *Proceedings of 4th IEEE International Workshop on Advanced Motion Control - AMC '96 - MIE*, volume 1, pages 259–264 vol.1, 1996.

[33] Mehdi Roozegar and Mohammad J. Mahjoob. Modelling and control of a non-holonomic pendulum-driven spherical robot moving on an inclined plane: simulation and experimental results. *IET Control Theory & Applications*, 11(4):541–549, 2017.

[34] Cyrill Stachniss, J Leonard, and Sebastian Thrun. *Springer Handbook of Robotics, 2nd edition.* 01 2016.

[35] Inertial Labs, Inc. Vio revolution in navigation and positioning, 2024. Available at: https://inertiallabs.com/vio-revolution-in-navigation-and-positioning/, accessed on 2025-05-10.

[36] Sherif A.S. Mohamed, Hashem Haghbayan, Tomi Westerlund, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. A survey on odometry for autonomous navigation systems. *IEEE Access*, PP:1–1, 07 2019.

[37] Dongjae Lee, Minwoo Jung, Wooseong Yang, and Ayoung Kim. Lidar odometry survey: recent advancements and remaining challenges. *Intelligent Service Robotics*, 17(2):95–118, 2024. Available at: `https://doi.org/10.1007/s11370-024-00515-8`, accessed on 2025-05-10.

[38] Zikang Yuan, Jie Deng, Ruiye Ming, Fengtian Lang, and Xin Yang. Sr-livo: Lidar-inertial-visual odometry and mapping with sweep reconstruction. *IEEE Robotics and Automation Letters*, 9(6):5110–5117, 2024.

[39] Karl Johan Åström and Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. ISA - The Instrumentation, Systems and Automation Society, 1995.

[40] Russ Tedrake. Underactuated robotics: Linear quadratic regulator, 2022. Accessed: 2025-08-17.

[41] Marawan Khalil and JMUWRobotics. 3d mapping sphere ros2. `https://github.com/JMUWRobotics/3D_Mapping_Sphere_ROS2`, 2025. GitHub repository.

[42] Yao Cai, Qiang Zhan, and Xi Xi. Neural network control for the linear motion of a spherical mobile robot. *International Journal of Advanced Robotic Systems*, 8, 01 2011.

[43] Raspberry Pi Foundation. Raspberry pi documentation - power supply, 2024. Accessed: 2025-06-13.

[44] JMUWRobotics. 3DTK - The 3D Toolkit. `https://github.com/JMUWRobotics/3DTK`, 2025. GitHub repository.

Proclamation

Hereby I confirm that I wrote this thesis independently and that I have not made use of any other resources or means than those indicated.

Würzburg, August 2025