

DESIGN AND DEVELOPMENT OF A SMART ROBOT FOR DISASTER RELIEF APPLICATIONS

A Final Year Project Report

Presented to

**SCHOOL OF MECHANICAL & MANUFACTURING
ENGINEERING**

Department of Mechanical Engineering

NUST

ISLAMABAD, PAKISTAN

In Partial Fulfillment of the
Requirements for the Degree of
Bachelor of Mechanical Engineering

by

SHEIKH MUSTAJAB AHMED

MUHAMMAD BILAL

KHIZAR ALI KHAN

SHAYAN KHAN MASOOD

Graduation Se 2024

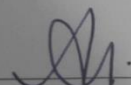
EXAMINATION COMMITTEE

We hereby recommend that the final year project report prepared under our supervision by:

SHEIKH MUSTAJAB AHMED	331657
MUHAMMAD BILAL	337422
SHAYAN KHAN MASOOD	339012
KHIZAR ALI KHAN	343498

Titled: "DESIGN AND DEVELOPMENT OF A SMART ROBOT FOR DISASTER RELIEF APPLICATIONS" be accepted in partial fulfillment of the requirements for the award of MECHANICAL ENGINEERING degree with grade A

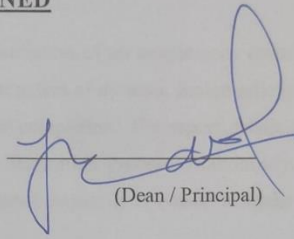
Supervisor: Dr Muhammad Tauseef (Assistant Professor), Affiliation <u>SMME</u>	 Dated: <u>Aug 2, 2024</u>
Committee Member: () Affiliation	Dated:
Committee Member: () Affiliation	Dated:


(Head of Department)

12-8-24
(Date)

COUNTERSIGNED

Dated: 12-8-2024



(Dean / Principal)

ABSTRACT

This document outlines the investigation and construction of an autonomous dynamic videography rolling robot. It encompasses the examination of dynamic design principles and the selection of suitable materials for project completion. The report details the application of acquired knowledge in the actual fabrication process. Additionally, it emphasizes the project's necessity and the significant impact it is poised to make in advancing the contemporary world.

ACKNOWLEDGMENTS

This project owes its successful completion to the timely guidance and support provided by our professors, lab supervisors, and friends. We express our gratitude to our initial project supervisor, Dr. Usman Bhutta, for accepting our proposal and providing consistent guidance, ensuring that we adhered to the project timeline. Our sincere thanks go to our current advisor, Dr. Muhammad Tauseef, for his support during crucial times and overall project management.

Special recognition is extended to Lab Engineer Ali Hassan, who offered valuable advice in designing the electronic circuit for project control. We are also grateful to Dr. Jawad Khan, our ex-co-supervisor, for granting us access to the additive manufacturing facility at the Aerial Robotics Lab in SINES.

A special acknowledgment goes to Mr. Mehran from the Aerial Robotics Lab at SINES, whose assistance in selecting and managing controls for the motor, battery, camera, and Wi-Fi systems was indispensable for the project's progress.

Lastly, we extend our heartfelt thanks to our friends and family members for their unwavering moral support, without which navigating this challenging journey would have been significantly more difficult.

ORIGINALITY REPORT

The concepts and content outlined in this report are original and have been independently developed by us. In instances where external content is incorporated, proper references are provided to duly acknowledge the contributions that have contributed to the completion of this project.

TABLE OF CONTENTS

ABSTRACT	3
ACKNOWLEDGMENTS	4
ORIGINALITY REPORT	5
LIST OF FIGURES	8
CHAPTER 1: INTRODUCTION	110
CHAPTER 2: LITERATURE REVIEW	112
Material Selection Procedure for Outdoor and Submersible Robot:	40
CHAPTER 3: METHODOLOGY	43
Buoyancy calculations	50
Motor calculations	64
Modelling the robots' dynamics in MATLAB Simulink	67
Model Results	70
Control System with arduino.....	74
Manufacturing Process: Building the Smart Rolling Robot	76
DATA ACQUISITION	78
Transferring Video Feed with ESP32 Controllers:	78
CHAPTER 4: RESULTS and DISCUSSION Initial Design	87
1. ABS (Acrylonitrile Butadiene Styrene):	98
2. ASA (Acrylonitrile Styrene Acrylate):	98
3. PETG (Polyethylene Terephthalate Glycol):	99

4. PLA (Polylactic Acid):	99
Motor Selection	101
CHAPTER 5: REVISED DESIGN	105
CHAPTER 6: CONCLUSION AND RECOMMENDATION	119
References	121

LIST OF FIGURES

Figure 1 Internal working of Barrycenter offset	115
Figure 2 Hamster ball design	116
Figure 3 Structure of Spring Loaded design	118
Figure 4 Structure of BHQ	20
Figure 5 HIT and Internal driving mechanism	20
Figure 6 A commercialized pendulum-driven robot, Rotundus	22
Figure 7 Rear view of Roball's steering mechanism	22
Figure 8 Mechanical structure of dual pendulum robot designed by Zhao: (1) Motor A; (2) Motor B; (3) Ballast B; (4) Ballast A; (5) Spring; (6) Linear Bearing; (7) Guide; (8) Outer Shell	23
Figure 9 Axes of rotation of pendulums for Kisbot II. (a) Side view (b) Front view	24
Figure 10 Picture of Sphere robot and its system of shifting masses	25
Figure 11 Image of spherical robot after reconfiguration	26
Figure 12 Movement types of Kisbot I	27
Figure 13 Image of hopping robot and its jumping mechanism	27
Figure 14 Soccer ball type robot movement	29
Figure 15 Breakaway view of the soccer ball robot	29
Figure 16 Mechanical breakdown of Gyrover	32
Figure 17 Schematic of rotor-based bob presented by S. Guanghui	33
Figure 18 V. Joshi's diametrically opposed rotor pair design	33
Figure 19 BHQ-5, a pendulum type robot with a CMG in place of a bob.....	34
Figure 20 BCO	45
Figure 21 Mass Inertia	46

Figure 22 Buoyancy diagram	50
Figure 23 Simulink Model	67
Figure 24 MATLAB Code	69
Figure 25 Velocity time Curve	70
Figure 26 Acceleration time Curve	70
Figure 27 Torque RPM Curve	71
Figure 28 Arduino Board	74
Figure 29 3D Printer	76
Figure 30 ESP32-CAM	78
Figure 31 Camera Mount	83

LIST OF TABLES

Thickness vs Von Mises 1	52
JGA25-371 Data sheet 2	99

LIST OF EQUATIONS

Equation 1	48
Equation 2	48
Equation 3	57
Equation 4	59
Equation 5	59

Equation 660

Equation 760

CHAPTER 1: INTRODUCTION

Smart Rolling Robot, a groundbreaking innovation in disaster response technology, is designed to excel in flood surveys with advanced sensors and AI capabilities. Its unique spherical design allows for seamless navigation both in water and on land, making it a versatile and efficient solution for flood disaster scenarios.

Field tests have demonstrated the Smart Rolling Robot's ability to swiftly navigate through submerged structures, showcasing its potential to revolutionize flood disaster response efforts. Equipped with state-of-the-art sensors, it autonomously detects survivors and facilitates timely communication, filling a crucial gap in current disaster management capabilities.

The rapid and precise flood disaster response provided by the Smart Rolling Robot addresses a critical need in disaster-prone areas. By efficiently surveying flood-stricken areas and autonomously identifying survivors, it enables emergency responders and disaster relief agencies to deploy resources more effectively, ultimately saving lives and reducing the impact of disasters on affected communities.

Emergency responders, disaster relief agencies, and communities in flood-prone regions stand to benefit greatly from the Smart Rolling Robot's capabilities. With potentially millions of people affected by flood disasters globally each year, the need for swift and efficient response measures is paramount. The Smart Rolling Robot offers a scalable solution that can be deployed in various locations, providing invaluable support to those in need during times of crisis. References could be added to this

CHAPTER 2: LITERATURE REVIEW

History of spherical robots

Spherical robots represent a compelling realm of research with unique features that captivate scholarly interest. These robots can be engineered to withstand harsh environments, operate holonomically (where holonomy denotes a robotic system whose orientation doesn't impact its desired travel direction), and swiftly rebound from collisions in a non-destructive manner.

The utilization of spherical robots has commenced in diverse domains, including underwater experiments, child development studies, and security reconnaissance. Some researchers are exploring the creation of robot swarms for task execution. Due to the inherent nature of a sphere, a robot with a spherical shell naturally follows the path of least resistance. An ideal spherical robot would possess genuine holonomy, enabling movement in any direction without altering its orientation. However, the current research focuses on spherical robots predominantly centered on internal mechanics and corresponding control systems. No single design has emerged dominantly, resulting in a broad spectrum of robotic characteristics and capabilities stemming from diverse internal driving mechanisms.

Originally, the initial vehicles were diminutive toys fueled by springs with a fixed axis of rotation. During that period, patents primarily concentrated on devising methods to store and transform spring energy using diverse mechanical solutions. A pivotal early challenge in these toys' evolution was integrating steering capabilities. Typically, the drive system of a spherical robot resides within its shell. To enable the outer shell's rotation, the drive system must transfer power, involving components like gears or electromagnetic devices. Achieving true holonomy poses a challenge, demanding the development of an internal drive mechanism capable of providing omnidirectional output torque to a sphere that can rotate arbitrarily, irrespective of the sphere's or drive mechanism's orientation. This necessitates the inner mechanics to rotate independently in three dimensions from the outer shell, presenting a complex design challenge with various potential solutions.

Numerous variations of internal propulsion devices have emerged, each balancing torque against holonomy, control against speed, etc. Some systems are straightforward, while others feature intricate designs accompanied by complex control algorithms. Teams have even explored propulsion through physical transformations of the outer shell, potentially leading to an entirely new category of robots. This review delves into concepts and novel variations of internal-drive mechanisms for spherical robots, beginning with designs grounded in the barycenter offset concept. The majority of early and current designs hinge on the principle of shifting the equilibrium of a sphere, with the most prevalent method being the displacement of the sphere's center of gravity.

In 1906, B. Shorthouse tackled this challenge by patenting a design allowing manual adjustment of the internal counterweight, causing the toy to follow a curved trajectory instead of a straight line (U.S. Patent 819,609). Subsequently, various mechanisms were patented to create self-propelled balls with distinct rolling paths, including irregular ones.

In 1957, J.M. Easterling innovated a design that substituted the mechanical spring power source with a battery and an electric motor (U.S. Patent 2,949,696). This ushered in the era of electric motors, employing mechanical solutions previously used in spring-driven inventions. The incorporation of shock and attitude sensing, using mercury switches, enabled control over motor operation, rolling direction, and the addition of light and sound effects.

In 1974, McKeehan introduced an active second freedom for a motorized ball, allowing the ball to alter its axis of rotation upon colliding with an obstacle, utilizing additional motors. This breakthrough paved the way for the development of radio-controlled ball robots (introduced in 1985 in U.S. Patent 4,541,814) and eventually, computer-controlled ball robots.

As toy cars gained popularity in the mid-1980s, they were frequently incorporated into spherical robots, providing a fully steerable 2-degree-of-freedom rolling toy (U.S. Patent 4,438,588). Spherical robots, also known as sphere robots, have a relatively brief history, with substantial research undertaken in the last two decades. One of the earliest spherical

robots, TITAN VII, was developed in 1991 by Hirose and his team at the Tokyo Institute of Technology, featuring six legs and multiple cameras for exploration in challenging terrain.

Subsequent researchers explored the potential of spherical robots. In 1998, Murata et al. developed the "Rolling Robot," comprising modules that could be combined into various shapes, designed for movement across diverse environments such as stairs and uneven surfaces. In 2005, Shimoyama et al. created "Puyu," a spherical robot with three nested spheres, including a camera in the innermost sphere, designed for hazardous environments like nuclear power plants.

A noteworthy advancement occurred when Kuniyoshi et al. introduced the "Spherical Drive System," consisting of a spherical shell with multiple arms extendable for directional movement, developed for search and rescue missions. In recent years, researchers have continued to innovate new types of spherical robots for various applications. For instance, in 2016, the University of California, Berkeley, researchers developed the Rolling Robot with Articulated Mechanisms (RRAM), a flexible-bodied robot designed to explore rough terrain, showcasing adaptability in navigating obstacles.

In conclusion, while the history of spherical robots is relatively short, the ongoing development of this technology demonstrates significant potential for diverse applications, including exploration, search and rescue operations, and inspection of hazardous environments. [15-19]

1. **Barycenter Offset (BCO)**

The term "barycenter offset" in the context of spherical robots refers to the intentional adjustment of a robot's center of mass (barycenter) to achieve a desired motion. Imagine a robotic sphere initially in a state of equilibrium. When the internal mechanisms of the sphere are set in motion, the distribution of mass within the ball changes, causing it to roll and find a new equilibrium position. Through precise timing and control methods, the robot

can navigate smoothly in its environment. However, a key limitation of this approach is the constrained maximum output torque, as the center of gravity cannot be shifted beyond the sphere's shell. To illustrate, envision a pendulum inside the sphere, a common and straightforward design. A simplified two-dimensional model demonstrates the torque generated and mechanically applied to the outer shell as a weighted bob of specific mass swings on an armature about a support rod positioned along the center axis of the robot. As the bob rotates, the center of mass also rotates, leading to the robot's rolling motion until it reaches equilibrium.

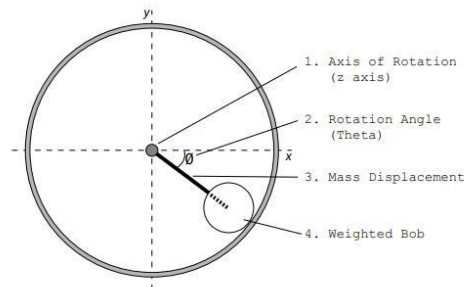


Figure 1

The maximum value of the torque that can be applied is τ_{max}

$$= mgr * \sin(\theta)$$

Where τ is the output torque about the z-axis (Figure), mg is the weight of the bob (Figure), r is the displacement of the bob's centre of mass from the shell's centre of mass (Figure), and $\sin(\theta)$ corresponds to the rotation angle from the horizontal (Figure). What follows are variations of the barycentre offset designs. [14] [20-24]

Types of BCO:

I. Hamster ball

A variant of the barycenter offset system, often known as the hamster ball design [10], takes inspiration from the way a hamster moves within a toy ball. This design involves placing a small-wheeled robot, typically a compact remote-control car, inside the ball. The weight of

the robot generates the necessary force for propulsion as it moves. The ball's navigation follows a non-holonomic pattern, like a car, where altering the direction of travel requires changing the internal robot's heading. Both single-wheeled and multiwheeled vehicles can be employed, with a four-wheeled differential-drive system producing distinct motion curves compared to a single-wheeled vehicle. The inclusion of a four-wheel drive system allows for differential drive capabilities, enabling the robot to turn in place and imparting holonomic characteristics to the vehicle. Additionally, this design is relatively straightforward to model, fabricate, and control. Control simplicity is maintained, especially for tasks that do not demand highly precise tracking, as the maneuverability mirrors that of a basic remote-control car.

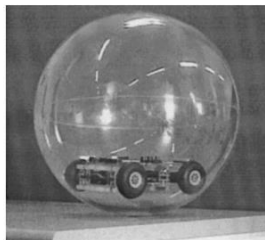


Figure 2

A significant drawback of this design is the tendency for some slippage to occur in the internal robot or driving mechanism. Nevertheless, implementing a closed-loop control system in conjunction with suitable internal tracking sensors can calculate and potentially alleviate this slippage issue. Beyond the energy loss and control complexities stemming from friction, another limitation of this design arises when the robot becomes airborne due to vibrations or encountering bumps. During airborne phases, traction between the shell and the internal robot's wheels diminishes to zero, resulting in a loss of momentum for the shell. This can adversely affect positional tracking. While sensors and a robust control system can somewhat address this issue, it remains unacceptable in tasks where precise navigation accuracy is paramount. [24]

II. Internal Drive Unit:

To address the challenge of shifting in internal robots, certain barycenter offset designs have implemented a system ensuring constant contact between the robot's wheels and the outer shell. This can be achieved through either a spring-loaded or fixed mechanism. In the spring-loaded design, a rod and spring are connected to the top of the internal robot, pressing against the shell to maintain continuous wheel contact. A 3-degree-of-freedom (DOF) ball bearing, attached on top of the spring, allows it to travel along the inner shell surface with minimal friction. A notable advantage of this constant contact setup is the ease of controlling the ball's mean speed by adjusting the motor wheel speed. At low speeds, the directional control is reasonably accurate. Additionally, the nature of the Internal Drive Unit (IDU) permits the use of either a sealed or honeycomb outer shell, making it adaptable to various conditions. While it serves as an excellent research platform, especially in academia, it faces challenges in real-world scenarios.

Despite being a commonly employed low-cost design, the control of an IDU-based robot's heading becomes challenging at high speeds. Issues such as slipping between the wheels and the shell, as well as slipping between the shell and the surface it travels on, can arise. Minimizing slippage between the wheel and shell requires adjusting the tension in the spring-loaded system, but a tighter fit increases friction forces throughout the robot. Moreover, an IDU system lacks the ability to utilize stored momentum; if the wheels stop, the robot may exhibit erratic behavior. When navigating inclines, the system must use power to keep the wheels spinning, limiting its ability to roll down small inclines without assistance. Conversely, rolling down steep inclines without controlled power may result in unpredictable movement. Achieving optimal balance is crucial in the design of the IDU system, as an off-axis center of mass could lead to undesired patterns of movement. [2426]

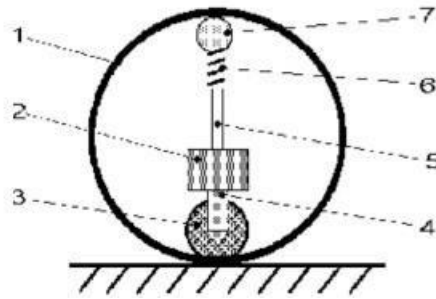


Figure 3

Structure of Spring-Loaded Design:

1. Robot body (case)
2. Controlling box
3. Driving wheel
4. Steering axis
5. Supporting axis
6. Spring
7. Balance wheel

III. Universal Wheel

Another design applying the principles of barycenter offset is BHQ-3, named after its dynamic model derived from the Boltzmann-Hamel equation. Combining features of the hamster wheel and the previous Internal Drive Unit (IDU) design, BHQ-3 can be visualized as a universal wheel system. The internal drive mechanism rotates freely inside the robot, facilitated by attached wheels. BHQ-3's IDU is engineered to remain stable on bumpy terrains, ensuring minimal internal mechanics shift. Two DC drive motors govern the robot: one controls the IDU's orientation, and the other manages the drive wheel's speed. This setup enables the ball to move with a zero turning radius, enhancing holonomy compared

to previous designs. Beyond barycenter offset maneuvers, the robot's velocity is regulated by the angular velocity of the driving wheel. Faster wheel rotation results in increased translational velocity. BHQ-3 exhibits versatility, capable of traveling in water, sand, and ascending small slopes. However, potential drawbacks include energy loss from sponge wheels friction and limitations in rolling unpowered down a slope, contingent on the motors and control method employed.

The HIT Spherical Robot is engineered to achieve independent steering and driving mechanisms. In contrast to pendulum-based designs where steering and turning are interdependent, creating a non-holonomic robot, HIT ensures mechanical independence for holonomic movement. Controlled by two motors—one for turning and one for driving—HIT's turning motor rotates the entire inner assembly along a rim at the equator, while the driving motor shifts the robot's center of gravity, propelling it in the desired direction. [26]

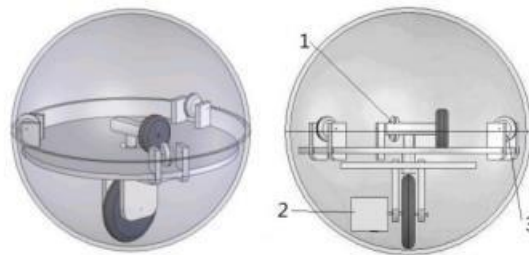


Figure 4

Structure of BHQ-3: 1–Motor, 2–Motor, 3–Sponge wheels



Figure 5

A picture of HIT and its internal driving mechanism

IV. Pendulum driven:

A widely adopted design in both industry and academia involves a pendulum-driven mechanism. In this model, a fixed shaft runs through the center of the outer shell, featuring a pendulum and bob that rotates around the shaft. The rotation of the pendulum shifts the center of mass outward, initiating the rolling motion of the shell. Horizontal movement of the pendulum along the equator results in a corresponding shift of the center of mass, causing the robot to turn in the respective direction.

An example of a commercially available pendulum-driven robot is Rotundus [13]. As the weight of the bob increases, the torque available to drive the robot also increases. However, a heavier bob results in a heavier robot. One significant limitation of this design is its challenge in ascending steep slopes. While a well-designed spherical robot can typically ascend slopes of around 30 degrees, surpassing this incline may necessitate impractical design techniques. Despite these limitations, the pendulum drive is a low-power, easily implementable design allowing for a sealed shell. Rotundus can achieve speeds of 6 mph, navigate through various terrains, and even float, supporting a payload of 1.81 kg.

A drawback of the pendulum-based design is its non-holonomic movement, introducing a turning radius associated with its motion. Like other barycenter offset designs, the center of mass cannot extend beyond the shell boundaries. Additionally, the design balance becomes crucial when considering the size and mass of internal elements, as larger sizes and masses increase both output torque and energy requirements.

Roball, designed as a child's toy, utilizes a pendulum-based mechanism with an added tilt feature for turning. This robot is intended for operation in unconstrained environments with minimal cost and complexity. Onboard sensors enable autonomous navigation, and all elements are situated on the robot's plateau (equator). Steering is achieved using a counterweight, where, in this model, the counterweight is a battery positioned at the bottom of the shell. The shell moves around the stationary counterweight, propelling the robot forward. In contrast to previous examples, where internal masses move within the shell, Roball's design involves the shell moving around a fixed internal weight. [24-27]



Figure 6 A commercialized pendulum-driven robot, Rotundus

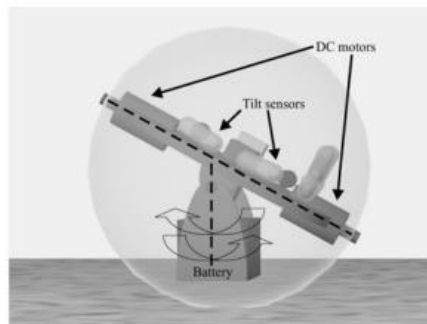


Figure 7 Rear view of Roball's steering mechanism

V. Double Pendulum:

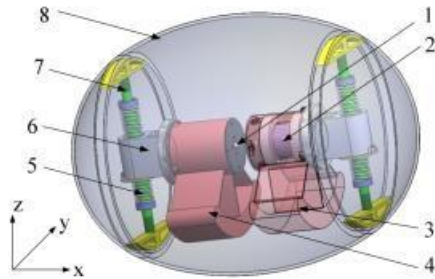


Figure 8 Mechanical structure of dual pendulum robot designed by Zhao: (1) Motor A; (2) Motor B; (3) Ballast B; (4) Ballast A; (5) Spring; (6) Linear Bearing; (7) Guide; (8) Outer Shell

A recent innovative concept discussed in contemporary literature involves a drive system featuring two internal pendulums. B. Zhao proposes such a system within an elliptical shell, enabling the robot to execute turning motions in place. The design prioritizes proof of concept and path planning over optimizing parameters like speed and maximum incline. Springs are incorporated to dampen impacts on internal mechanisms when navigating challenging terrains or encountering large obstacles. The literature presents the theoretical

framework, verified through simulation, and showcases a proof-of-concept physical prototype.

This robot achieves in-place turning using the "Stick-Slip" principle. In the initial stage (stick), each pendulum is slowly rotated to a horizontal position in opposite directions, causing an equilibrium shift, but not enough to overcome static friction. In the subsequent stage (slip), the pendulums are rapidly forced back to their original vertical orientation, overcoming static friction and allowing the robot to slip and turn in place.

Another variation of the dual pendulum design is observed in Kisbot II. Its predecessor, Kisbot I, employed a single pendulum design with adaptive legs for stair climbing. Kisbot II eliminates the need for the stick-slip principle by enabling the internal mechanics to rotate about an axis perpendicular to the pendulum movement. This additional degree of freedom allows Kisbot II to turn from left to right and front to back, eliminating reliance on the stick-slip principle.

Some teams have introduced an alternative approach for barycenter offset designs by concentrating the majority of materials at the center of the robot. This reduces the energy needed to spin the sphere. Shafts connect the center mass to the outer shell, and weights are designed to traverse these shafts. By moving the weights along the shafts, the center of mass is altered, initiating the ball's rolling motion. Contrary to the pendulum design, this shifting mass approach is holonomic, capable of moving in any direction regardless of orientation. However, the controls are more complex, requiring real-time orientation and distance data for all masses. Additionally, the slow movement of internal weights and potential challenges in downhill rolling are drawbacks of this design. [24]

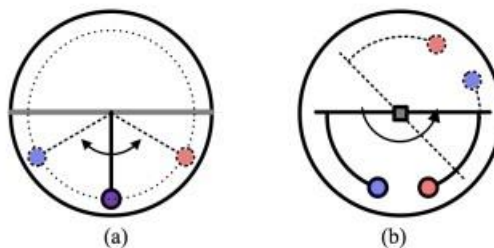


Figure 9 Axes of rotation of pendulums for Kisbot II. (a) Side view (b) Front view

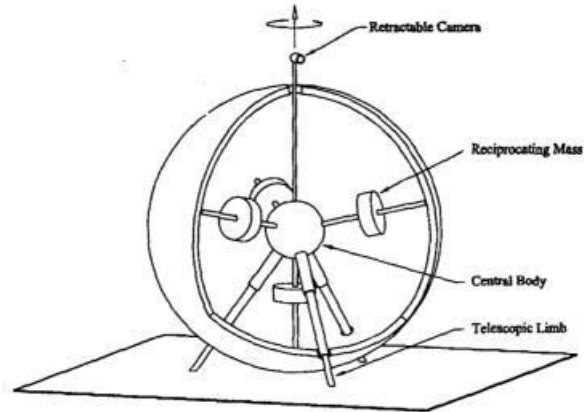


Figure 10 Picture of Sphere robot and its system of shifting masses

VI. Notable Enhancements:

Innovative enhancements to spherical robots are as varied as the propulsion methods employed. These designs integrate an array of features such as sensors, telescopic cameras, adaptable legs, and even jumping mechanisms. Some designs go a step further, showcasing the ability to undergo complete transformations.

N. Chadill introduces a reconfigurable robot capable of transforming from a sphere into a dual-hemisphere platform equipped with three legs and omnidirectional wheels. The underlying concept revolves around the robot compacting into a sphere for transport and deployment, transforming into a leg-wheeled robot afterward. While the current design doesn't actively roll in its spherical configuration, the concept holds potential for application in existing spherical robot designs.



Figure 11 Image of spherical robot after reconfiguration

Another reconfigurable spherical robot with deployable legs is Kisbot I, the precursor to Kisbot II. Kisbot I features two drive modes: a pendulum-driven mode and a wheel-driven mode. In its pendulum mode, the legs retract, allowing the robot to maneuver accordingly. When transitioning to its wheeled mode, the legs extend, providing stability for driving like a conventional wheeled vehicle. These deployable legs, residing in independent hemispheres capable of rotating independently, offer the flexibility for the robot to stabilize itself in unconventional terrains.

An alternative approach to enabling a spherical robot to overcome large obstacles is to equip it with the capability to jump. In a paper by L. Bing, a spherical robot propelled by dual pendulums is retrofitted with a jumping mechanism. This mechanism empowers the robot to jump at a desired angle and direction upon reconfiguration. The ability to jump is particularly valuable for such robots, compensating for their inherent challenge in climbing steep slopes. The jumping process involves storing energy in a spring, which, when released, propels a mass attached to it upwards. The conservation of momentum results in the entire robot ascending with the mass. [24] [29]

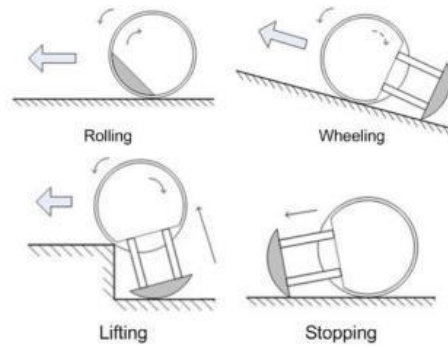


Figure 12 Movement types of Kisbot I

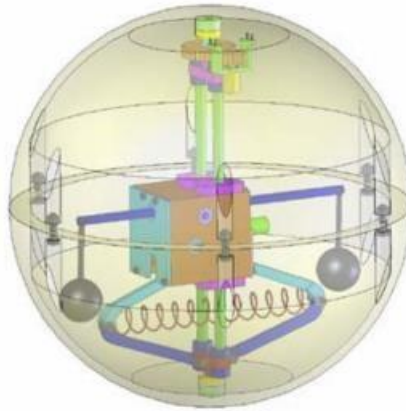


Figure 13 Image of hopping robot and its jumping mechanism

2. **Shell Transformation:**

While not as prevalent as barycenter offset designs, the concept of shell transformation presents a unique approach to propelling spherical robots. This relatively recent idea

introduces intriguing concepts associated with the propulsion mechanism. Instead of relying on a complex internal mechatronics system to drive the sphere, these robots utilize the transformation of their outer body. This transformation can occur through the deformation of the enclosing shell or the influence of environmental elements such as wind or water acting on the body itself. Depending on the specific design, this category of robots may exhibit greater versatility than a barycenter offset system. However, it's essential to note that this concept is still in its early stages compared to the previously discussed designs, making it a potential avenue for future research. Therefore, these innovative ideas warrant further investigation. [30]

Types of Shell Deformation:

I. Pressurized Air Bladders:

M. Artusi introduces a basic deformable spherical rover with an outer shell comprising four sections of dielectric elastomer actuators. The application of an electric field allows the sequential transformation of these sections, enabling the robot to roll. K. Wait proposes a more advanced concept using pressurized air bladders, similar to a soccer ball's structure. Each pentagonal section of the outer sphere represents an elastomer bladder that can inflate and deflate, determining the direction of movement. This system allows for holonomic movements by inflating multiple bladders in various combinations.

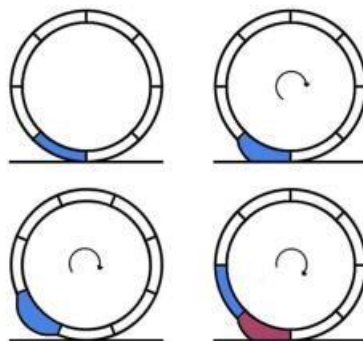


Figure 14 Soccer ball type robot movement

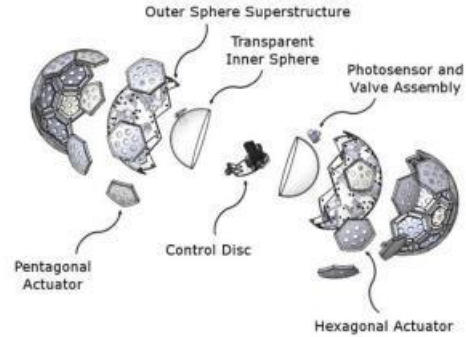
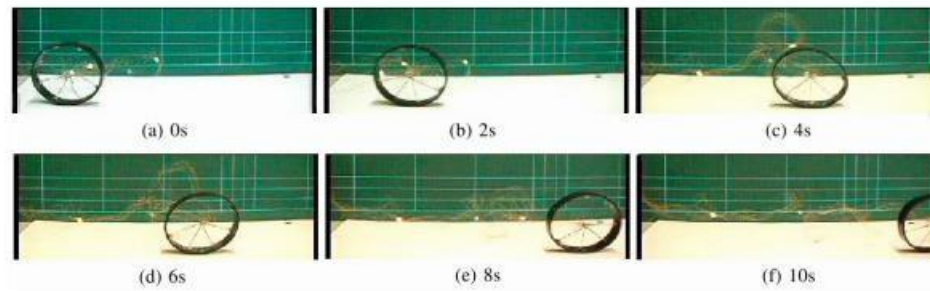


Figure 15 Breakaway view of the soccer ball robot

An interesting outcome of this research is the non-computational control method, contrasting with earlier designs. Inside the soccer ball shell resides another spherical robot, closely resembling the hamster ball and IDU design. A high-power LED is positioned at the rear of the control disc within the inner shell. The control disc, manipulated via radio control, moves and directs the LED onto the photo diodes of the bladders, selectively inflating them to propel the robot. [31]

II. Shape Memory Alloys:

Others have suggested altering the outer shell to enable the robot to jump. T. Yamanaka presents a robot with "Super-ball-like" properties, leveraging the unique bouncing characteristics of Superballs through spin and elasticity. By manipulating the outer shell and incorporating a rotor into the internal structure, the robot can execute controlled hops. Sugiyama introduces a locomotion method entirely based on the manipulation of the outer shell. The robot utilizes shape memory alloy (SMA) coils that extend/retract when voltage is applied, allowing it to flatten like a pancake and quickly spring back to its original form, facilitating jumping. Proper control also enables smooth locomotion.



Locomotion by deformation of SMA coils

3. Conservation of Angular Momentum:

Barycenter offset designs are widely favored for their simplicity and ease of control, with shell transformation being the next commonly employed design. However, a significant drawback of barycenter offset designs is their torque limitation, as the center of mass cannot extend beyond the sphere.

In the past two decades, researchers have explored the integration of control moment gyroscopes (CMGs) into spherical robots. This involves rapidly spinning a large flywheel and rotating it around an axis to utilize the laws of conservation of angular momentum for sphere control. This method links the output torque of the internal mechanism to the angular velocity of the CMGs. The use of CMGs represents the latest approach to achieving an output torque surpassing that of a barycenter offset system. Various designs with flywheels have been implemented, each presenting different levels of success and challenges.

An intriguing aspect of CMG usage is that these systems generate reaction forces in all three spatial dimensions. If a CMG is spinning around the X-axis and is rotated about the Y-axis, a torque about the Z-axis (precession) is induced. While this feature holds potential for generating torque in the intended direction, it also introduces control complexities. Depending on the robot's design, the precession torque can either be harnessed to control or enhance the robot's angular momentum, or if not considered properly, it may lead to undesired steering. Therefore, while a gyro-based or gyro-augmented spherical robot may exceed a barycenter offset robot in terms of torque, additional design challenges must be addressed. [32-34]

Types of COAM based Spherical Robot: I.

Balancing:

An early experiment involved Gyrover, a disc-shaped object that maintained balance on its edge. Internal gyroscopes were employed to stabilize the robot, and the precession torque effects were utilized for steering. While innovative in design, its practicality for commercial applications might be limited. The literature suggests its suitability for high speeds and rough terrains, with the ability to perform in-place turns, offering a certain degree of holonomy. However, challenges may arise in correcting its orientation if it topples over, and achieving precise movements may be demanding unless the embedded electronics and mechanics are meticulously designed.

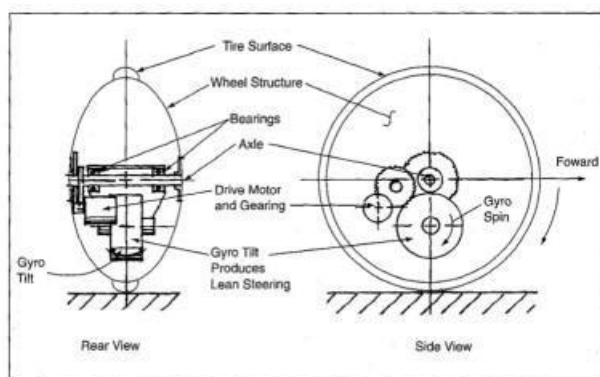


Figure 16 Mechanical breakdown of Gyrover

II. Uni-Dimensional COAM:

In this configuration, a variable-speed rotor is employed as a bob. However, this specific design deviates from the conventional pendulum-type drive system using the CMG bob in a distinct manner. Adjusting the speed of the spinning CMG induces shell rotation. Moreover, manipulating the CMG's up-and-down movement, akin to a bob, when it is already spinning rapidly generates a precession torque, potentially achieving higher torques compared to a standard pendulum-based design. This robot employs COAM in a one-dimensional fashion but possesses the capability to reorient that dimension for controlling its movement.

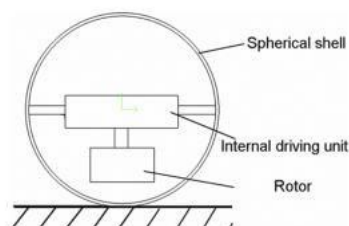


Figure 17 Schematic of rotor-based bob presented by S. Guanghai

V. Joshi introduces a robot controlled by two pairs of diametrically opposed CMGs, each pair managed by a single motor controller. As the angular velocity of a pair increases, the shell rotates in the opposite direction to uphold the system's total angular momentum. With a second pair within the ball, the ball gains a second degree of freedom, allowing it to move

in a true holonomic manner. The state space calculations for this robot type are more intricate than a simple barycenter offset design, but genuine holonomy can be achieved. This is classified as a Uni-Directional COAM robot, involving two systems that apply the principles of COAM, with each system affecting only one spatial plane in its COAM dynamics.

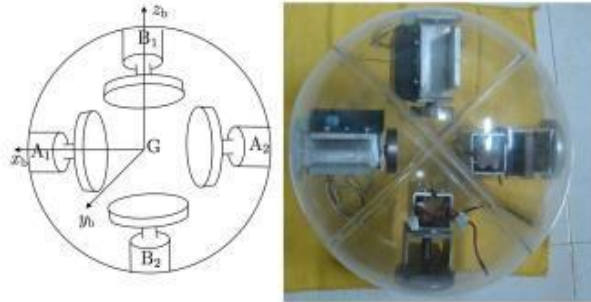


Figure 18 V. Joshi's diametrically opposed rotor pair design

III. Tri-Dimensional COAM:

The BHQ-5, the fifth robot in the BHQ series, employs a pendulum-type drive system with the integration of a control moment gyroscope (CMG) to enhance its stability. By placing the CMG in the position where a conventional pendulum system would have a bob, this robot can rotate itself based on the CMG's rotations. Moreover, adjusting the CMG's orientation and movement can increase the robot's angular momentum, providing more torque than a standard pendulum and bob setup. To simplify the concept for explanatory purposes, envision this as a pendulum-type robot with a bob of variable mass. Similar to a pendulum-type robot, the bob in this design can steer or propel the robot, but with varying power levels. The BHQ-5 harnesses the precession torque of a CMG, incorporating all three spatial dimensions into its Conservation of Angular Momentum (COAM) dynamics.

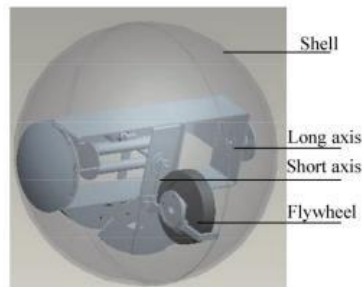


Figure 19 BHQ-5, a pendulum type robot with a CMG in place of a bob

Spherical robots find diverse applications, each requiring specific control methods. Table 1 provides a concise taxonomy of these robots. Active drive designs commonly rely on three fundamental principles: barycenter offset, outer-shell deformation, or conservation of angular momentum. Among these, barycenter offset designs are often the simplest and more easily controllable. They can generally be analyzed using a single model. However, their power is constrained as the center of gravity cannot be moved outside the shell.

Common barycenter offset designs include single-wheel, car, universal wheel, and pendulum models, which have abundant research data available.

Designs employing conservation of angular momentum typically generate torque by manipulating a single axis or three axes of a control moment gyroscope (CMG). These designs can utilize the counter-rotational force generated during CMG spin (single-axis) or the precession torque created by rotating a spinning CMG orthogonally to its axis of spin (triple-axis). The torque magnitude in single-axis designs is controlled by the acceleration of the spinning CMG, while in triple-axis designs, precession torque is controlled by the product of angular velocity and rotational acceleration. Integrating CMGs into spherical robots offers a solution to power constraints imposed by barycenter-offset designs but presents its own set of challenges.

Shell transformation designs represent a relatively new concept that may be more challenging to design but can be controlled with a calculation-less approach. Some groups

are merging these three concepts, creating robots with advantages over those based on a single principle. Noteworthy examples include hybrids, such as a pendulum type with a CMG bob.

After reviewing the information about various types of spherical robots, it can be concluded that the pendulum-driven approach offers a reliable and practical solution for this type of innovation. Several factors contribute to the viability of the pendulum-driven design:

a) Simplicity and Ease of Control:

- Pendulum-driven designs, such as those based on barycenter offset, are generally less complex than some alternative methods.
- These designs are easier to control, and their behavior can be analyzed with a single model, simplifying the development and implementation process.

b) Holonomic Movement:

- Pendulum-driven robots, especially those incorporating control moment gyroscopes (CMGs), can achieve true holonomic movement.
- The ability to move in any direction without reorienting the robot provides a significant advantage in terms of maneuverability.

c) Conservation of Angular Momentum:

- Incorporating CMGs into the pendulum-driven design allows for effective conservation of angular momentum.
- By manipulating the precession torque of CMGs, these robots can achieve higher torques compared to traditional pendulum-based designs.

d) Reliability and Stability:

- Pendulum-driven designs offer stability due to their inherent properties of maintaining equilibrium.
- The predictable behavior of pendulum systems contributes to the reliability of these robots in various environments.

e) Versatility:

- Pendulum-driven designs can be adapted for different applications, including navigation in rough terrain, climbing, and even jumping in some cases.
- The versatility of these designs makes them suitable for a wide range of scenarios.

f) Integration with Other Principles:

- Some innovative designs combine pendulum principles with other concepts, such as CMGs or shell transformation, to enhance performance and overcome limitations.

While each design approach has its merits, the pendulum-driven spherical robots, particularly those incorporating advanced technologies like CMGs, stand out as a reliable and practical choice for achieving controlled and versatile movement in various applications. The simplicity of the design, coupled with the ability to integrate additional features, makes it a promising avenue for further exploration and development in the field of spherical robotics.

MATERIAL SELECTION PROCEDURE FOR OUTDOOR AND SUBMERSIBLE ROBOT:

1. Define Environmental Requirements:

Identify the environmental conditions the robot will encounter, including exposure to sunlight, temperature variations, moisture, and submersion in water.

2. Assess Mechanical Properties:

Evaluate mechanical properties such as strength, stiffness, and impact resistance to ensure the selected materials can withstand outdoor use and potential impacts during operation.

3. Consider Corrosion Resistance:

Prioritize materials with high corrosion resistance to withstand exposure to water and moisture, minimizing the risk of degradation over time.

4. Evaluate Waterproofing Capabilities:

Choose materials and coatings that provide effective waterproofing to protect internal components from water ingress during submersion.

5. Assess UV Stability:

Select materials with UV stability to prevent degradation and discoloration when exposed to sunlight for extended periods.

6. Ensure Buoyancy and Density:

Consider the buoyancy and density of materials to ensure the robot maintains proper buoyancy and stability when submerged in water.

7. Check Electrical Insulation Properties:

Ensure selected materials provide adequate electrical insulation to protect sensitive electronic components from water and moisture.

8. Explore Material Compatibility:

Verify compatibility between selected materials and manufacturing processes such as 3D printing, injection molding, or machining to facilitate fabrication of robot components.

9. Conduct Material Testing:

Perform material testing, including mechanical, corrosion, and waterproofing tests, to validate material suitability for outdoor and submersible applications.

10. Select Materials and Components:

Based on the assessment and testing results, choose materials and components that best meet the requirements for outdoor durability and submersion resistance.

11. Integrate and Fabricate:

Integrate selected materials into the design of the robot's structure, chassis, and enclosures, ensuring proper sealing and waterproofing measures are implemented.

12. Prototype and Test:

Fabricate prototypes of robot components using selected materials and conduct thorough testing to evaluate performance under simulated outdoor and submersible conditions.

13. Iterate and Optimize:

Iterate on the design and material selection process based on testing feedback, making adjustments to improve durability, waterproofing, and overall performance.

Document Material Selection:

Document the material selection process, including rationale for chosen materials, testing results, and any design iterations, for future reference and replication.

CHAPTER 3: METHODOLOGY

Following are the steps which we go through in order to carry out our project:

- I. Finalizing the material
- II. Optimum design of ball
- III. Optimum design for turning mechanism
- IV. Buoyancy Calculation
- V. FEM Analysis of sphere and shaft
- VI. Motor's calculation and selection
- VII. Servo motor and bearing calculations and selection
- VIII. Control system with Arduino
- IX. Data Acquisition
- X. Installation of cameras
- XI. Battery Selection
- XII. Waterproofing
- XIII. Balancing of spherical robot

Finalizing the material

From the above discussed section in literature review, we have selected PETG as our primary material for 3D printing our spherical robot and its other constituents.

Various design options were considered to achieve the project's objectives. The Barycenter Offset mechanism was selected for this robot. The designs, categorized according to their intended purposes, are outlined as follows:

Designs of the robot:

- **Design with Barycenter offset:** The term "barycenter offset" in physics or engineering is the center of mass of an object is the point at which the entire mass of the object can be considered to be concentrated. If this point is offset from the geometric center, it means the center of mass is not at the center of the object.

For a spherical body, having an offset center of mass can impact its motion, especially if the body is rolling. The offset center of mass introduces a torque that can cause the spherical body to start rolling. This is analogous to the way a cylinder or a wheel with an offset mass distribution can initiate rolling motion when placed on a surface.

In summary, having an offset center of mass in a spherical body can contribute to its ability to initiate and sustain rolling motion, which can be advantageous in various applications, including robotics or mechanical systems.

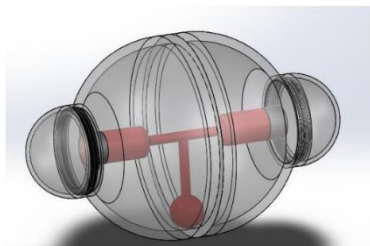


Figure 20 BCO

Inertia Mass design:

In an inertial mass design for moving a spherical body, internal masses are strategically placed within the sphere to leverage the principles of inertia. These masses, which may be adjustable or fixed components, contribute to the overall stability and motion of the spherical body. When an external force is applied, the inertial masses resist changes in their state of motion, generating angular momentum. By controlling the distribution or mobility of these masses, the direction and speed of the sphere's movement can be purposefully manipulated. This design allows for dynamic and controlled motion, with the potential for sophisticated feedback mechanisms to enhance stability and responsiveness.

In this configuration, a mass is hung from the axle to impart inertia to the motor's body. Essentially, the mass grips or engages with the motors responsible for rotating the ball's shell. As the motors need the torque required for rotating a load, the suspended mass fulfills that clutching function. The system's design is illustrated below.

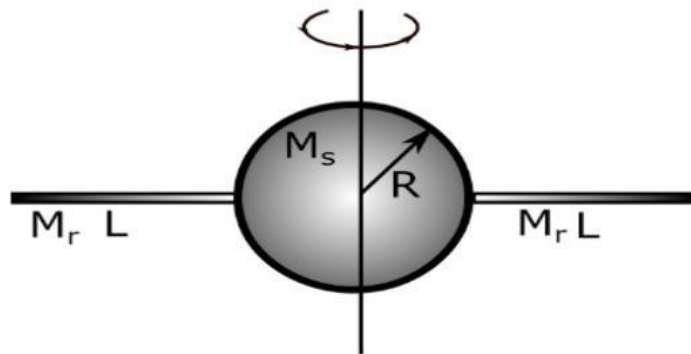
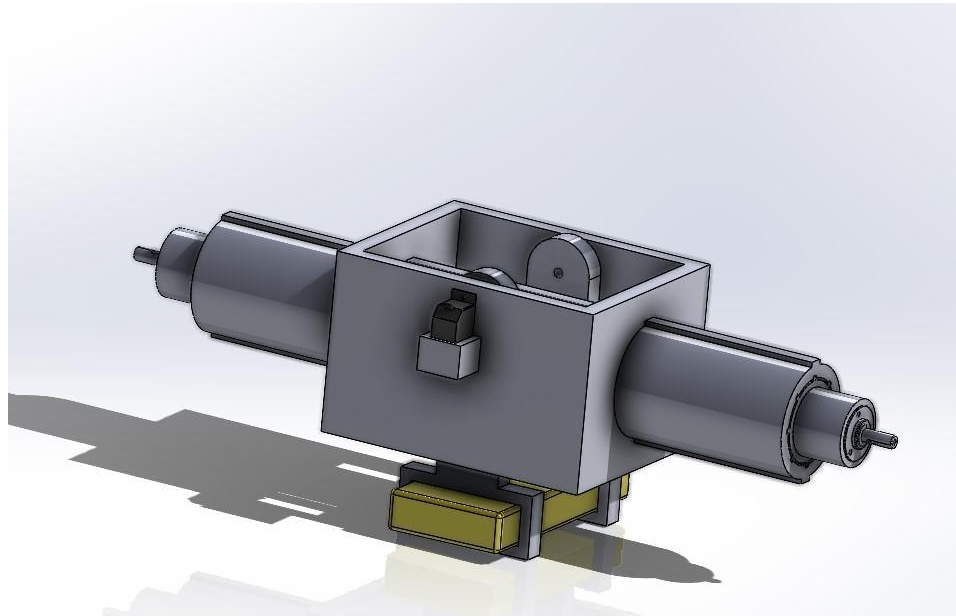


Figure 21 Mass Inertia

Optimum designs for turning mechanism

For this section, we have selected the pendulum-based turning mechanism, for which the reasons are quoted below:

- Pendulum design: In this design, a pendulum will hang in the center of the ball with a fixed shaft. The mass can be rotated to the desired angle using a servo motor, enabling the ball to make turns.



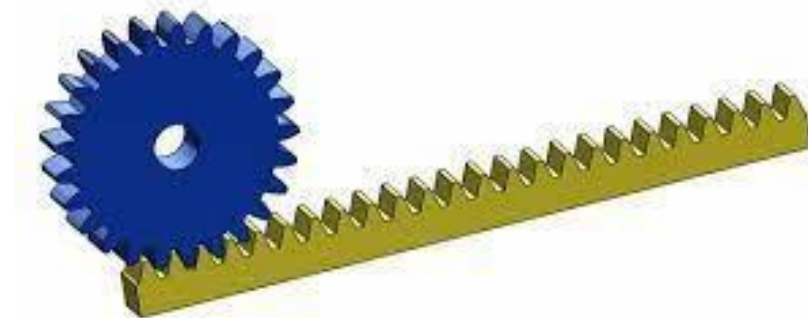
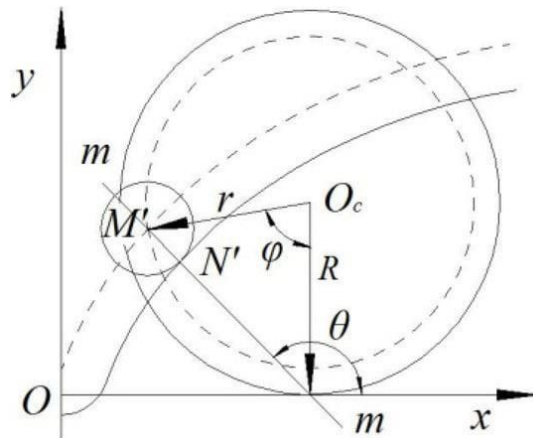
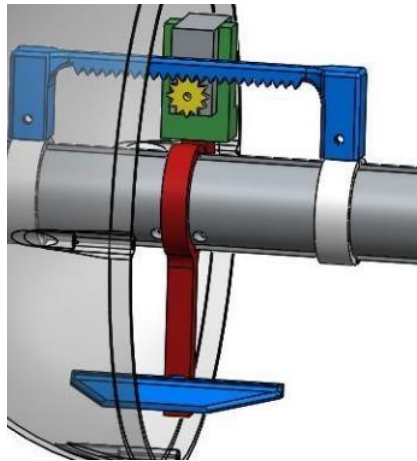
Internal Assembly Diagram

Advantages of Pendulum Design for Turning a Spherical Body:

1. Smooth Turning: The pendulum design facilitates smooth turning by allowing controlled angular movement of the mass, contributing to stability.
2. Versatility: It provides versatility in adjusting the turning angle by controlling the movement of the suspended mass.
3. Dynamic Response: The dynamic response of a pendulum system allows for quick and agile turns, making it suitable for dynamic environments.

Pinion and rack design:

In this design, a pinion and rack system is employed to shift the sliding mass to the right or left side of the ball, facilitating the achievement of turns.



Advantages of Rack and Pinion Design for Turning a Spherical Body:

1. Precise Control: The rack and pinion system allow for precise control of the turning motion, enabling accurate adjustments.
2. Mechanical Simplicity: It is a mechanically straightforward system, making it easy to implement and maintain.
3. Efficiency: The direct engagement of the gears results in efficient power transfer, minimizing energy loss during the turning process.

BUOYANCY CALCULATIONS

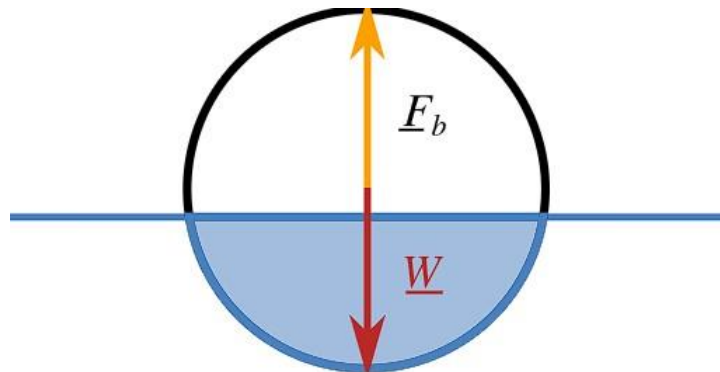


Figure 22 Buoyancy diagram

Submersion calculations were crucial in determining the buoyancy and stability of the spherical 3D printed robot when submerged in a liquid environment. The buoyancy force acting on the robot was calculated using Archimedes' principle, which states that the buoyant force on an object submerged in a fluid is equal to the weight of the fluid displaced by the object.

Next, the density of the fluid in which the robot would be submerged was considered. For instance, if the robot was designed to operate in water, the density of water (approximately 1000 kg/m^3) was used in the calculations.

The following formula was used to calculate the buoyancy:

$$\text{mass required for submergence up to } h = \frac{1}{3} \times \pi \times h^2 \times (3 \times r - h) \times \text{Density of water}$$

The value of the radius was assumed based on manufacturing options available, ($r = 150$) mm. While the value of (h) was taken according to a submersion of 75% of the radius, which comes out as 112.5 mm.

The fact that we have considered our robot to be 25% submerged in water is based on our own thinking, as this spherical robot is amphibian as per our design, so we didn't want this spherical robot to be completely or partially submerged. That's why we have done

our calculations according to our own decision, and the submersion height can be increased based on what one wants to achieve. This project is solely for the purpose of detecting the objects on the surfaces, and not under the surfaces.

The mass ($m = 4.5\text{kg}$) is the value used later on in calculations as it is our targeted mass for the required submergence.

FEM ANALYSIS

Two important conditions were studied to determine the thickness to be used for the sphere. The first was the stress induced as the sphere stands on a rigid floor: this contact stress would be contributing of the spherical robot as it rolls. The second condition is the sphere being able to withstand the impact against a wall when it is travelling at its maximum speed. The application libraries of COMSOL along with tutorials available on the COMSOL website were used to prepare for this FEA application.

For the impact case, initially, the impact condition was studied by using a transient study, which would have been indeed a more fine-tuned approach. The application titled “impact between two soft rings” was used as reference which analyzed the impact of two rings against each other. This was a transient study which used the solid mechanics module of COMSOL and utilized the “contact pair” conditions available therein. We edited this application into a ring colliding against a rigid wall, which gave good results. But when the same conditions were replicated for a 2d axisymmetric study for our spherical shell, the “contact pair” setting (which basically prevents the boundaries of two distinct parts from colliding against each other) refused to work wherein the spherical shell crossed the rigid

wall boundaries. A 3D model was also made where the contact pair setting returned to functionality, however the computation cost of a transient study for a 3d model was far too much, something which our laptops could not perform.

As a result, another approach was used, a simpler stationery study based on hertzian contact theory was used for the two conditions.

Shigley's Machine Design was consulted wherein the following equation for hertzian contact between two spheres was included.

Equation 1

$$a = \sqrt[3]{\frac{3F}{8} \frac{(1 - \nu_1^2)/E_1 + (1 - \nu_2^2)/E_2}{1/d_1 + 1/d_2}}$$

Equation 2

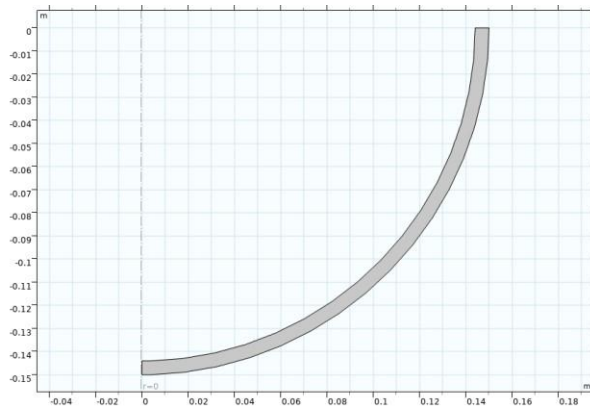
$$P_{\max} = \frac{3F}{2\pi a^2}$$

Where a is the radius of the circular contact area and Pmax is the maximum pressure developed.

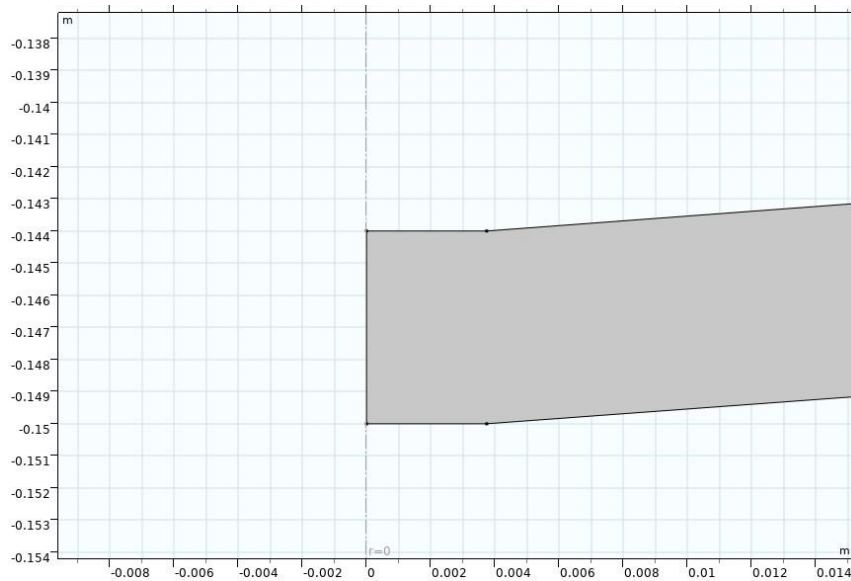
Since in our case the two contacting bodies were a sphere and a rigid flat, editions were made to the formula wherein the elastic modulus and the radius of one of the spheres reaches infinity simplifying the formula to:

$$a = \sqrt[3]{\frac{3F}{8} \frac{(1 - \nu_1^2)/E_1}{1/d_1}}$$

With this value of a, an axisymmetric 2d geometry of the following form was made.



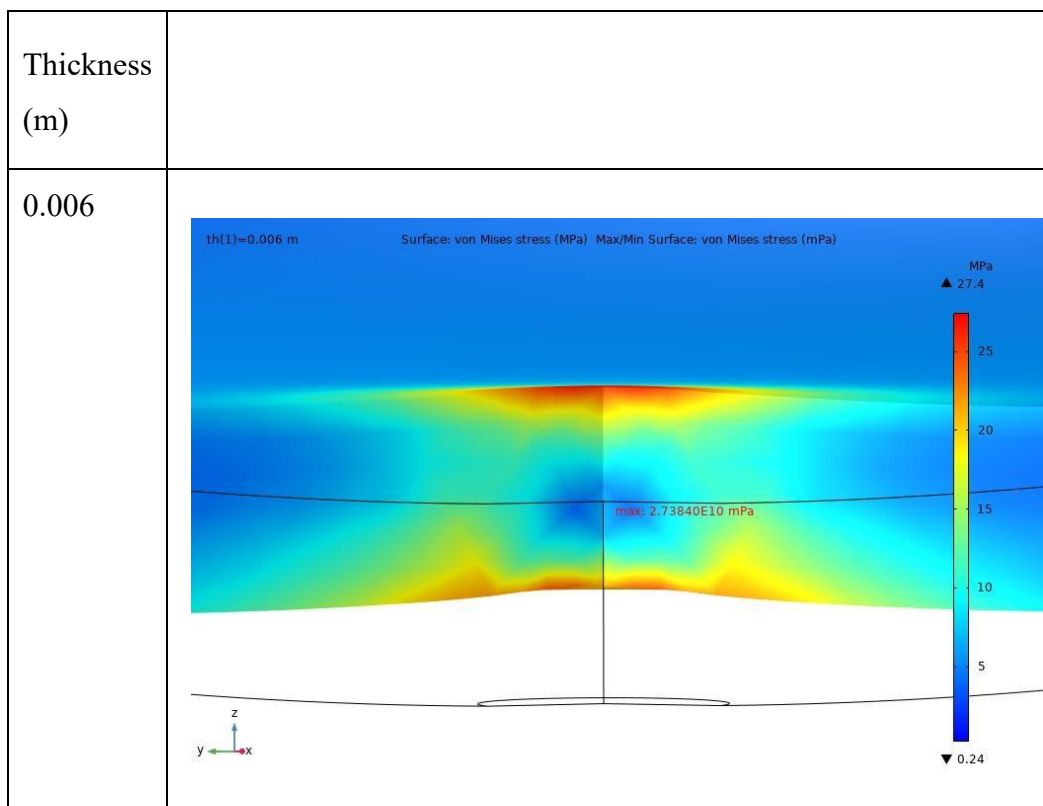
The axisymmetric 2D geometry used for the analysis

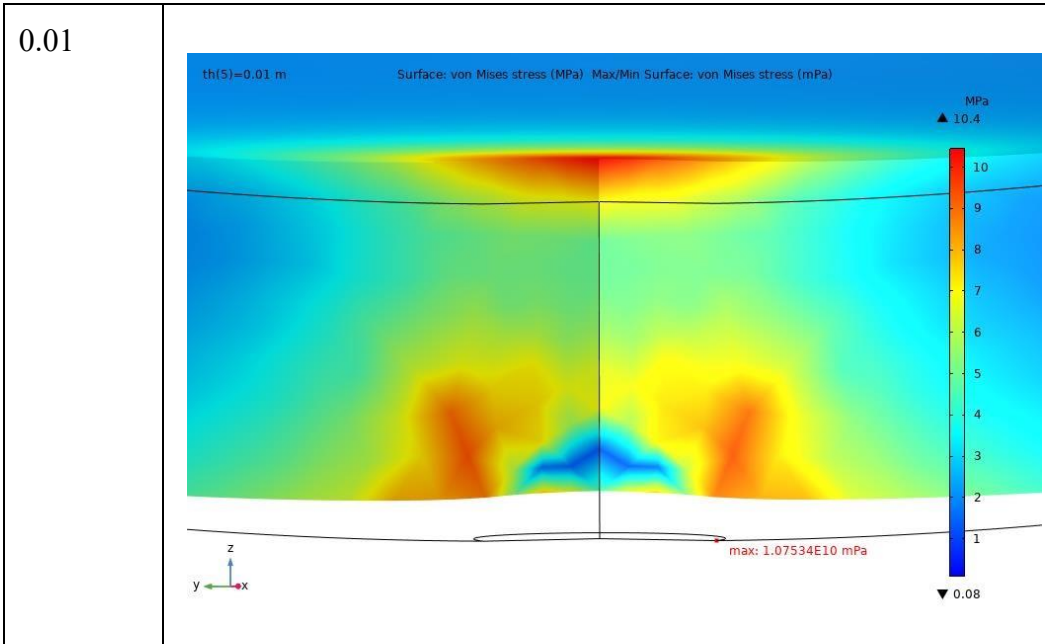
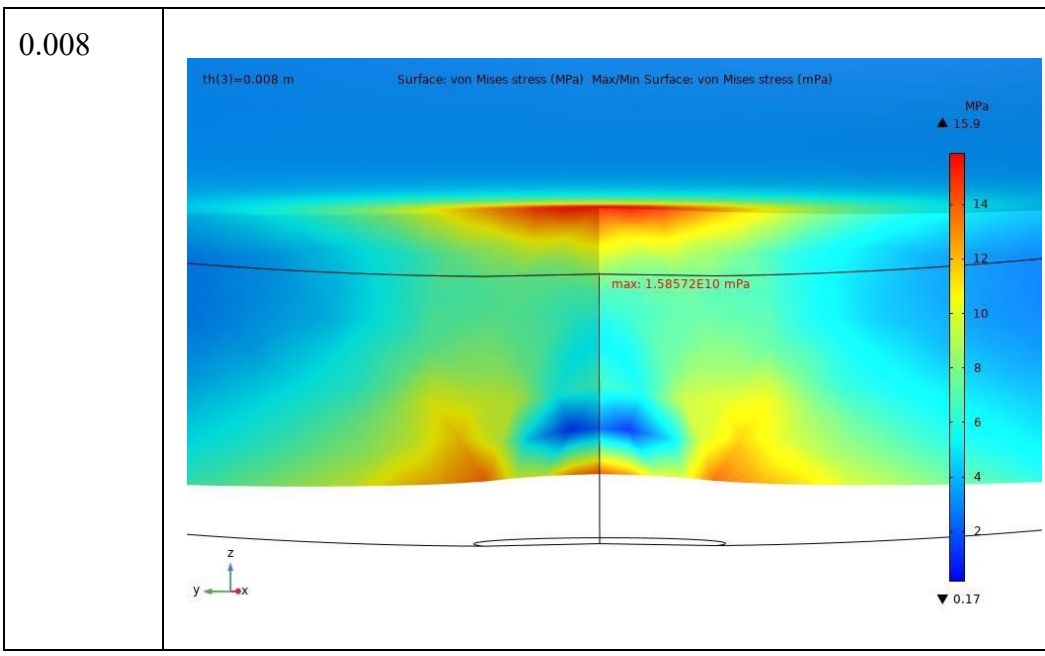


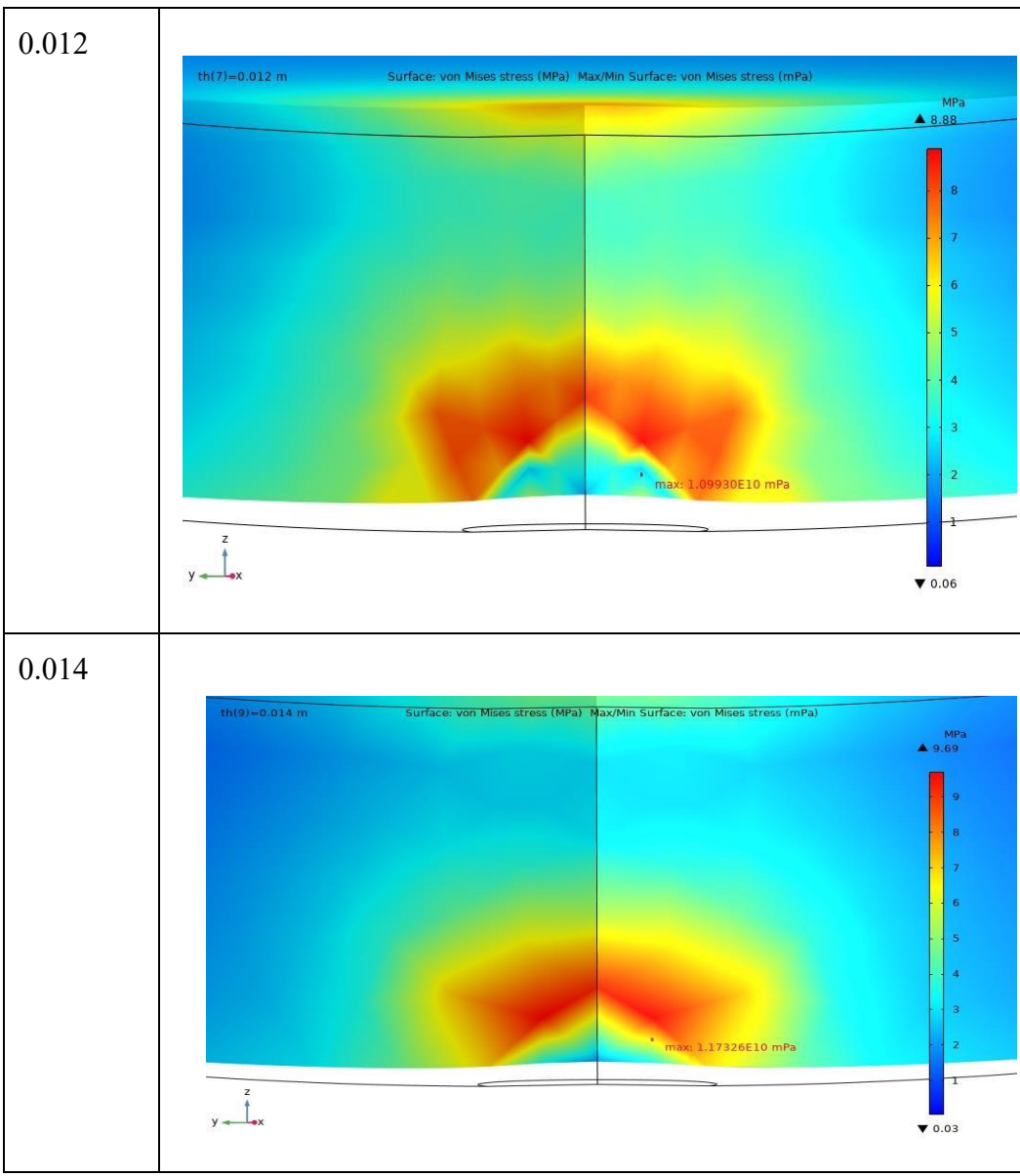
A close-up of the contact surface.

The value of P_{max} was calculated according to the impact force calculated with an estimated collision time of 0.1 second, a mass of 7kg and a max speed of 3.7m. Of course, using exact collision time would have been preferable however, that would have required prototyping and actual drop testing, something which is not yet available at this point. This pressure was applied as a boundary condition on the contact surface and a parametric study involving the thickness of the spherical shell was carried out. The results were post

processed to show von Mises stress (equivalent stress). The following 3d plots were achieved.







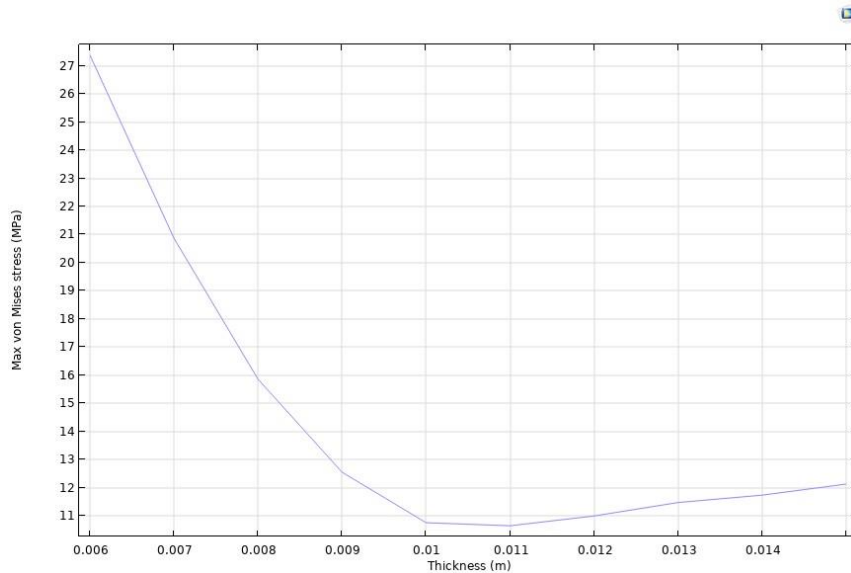
FEA ANALYSIS

Thickness vs Von Mises 1

Shell Thickness (m)	Maximum von Mesis Stress (MPa)
0.006	27.3839811
0.007	20.8706912
0.008	15.8572424

0.009	12.5499499
0.01	10.7534017
0.011	10.6441367
0.012	10.9930263
0.013	11.4710976
0.014	11.7325743
0.015	12.1268374

Overall, the maximum von Mises stress in the volume due to impact as a function of the shell can be represented in the graph below

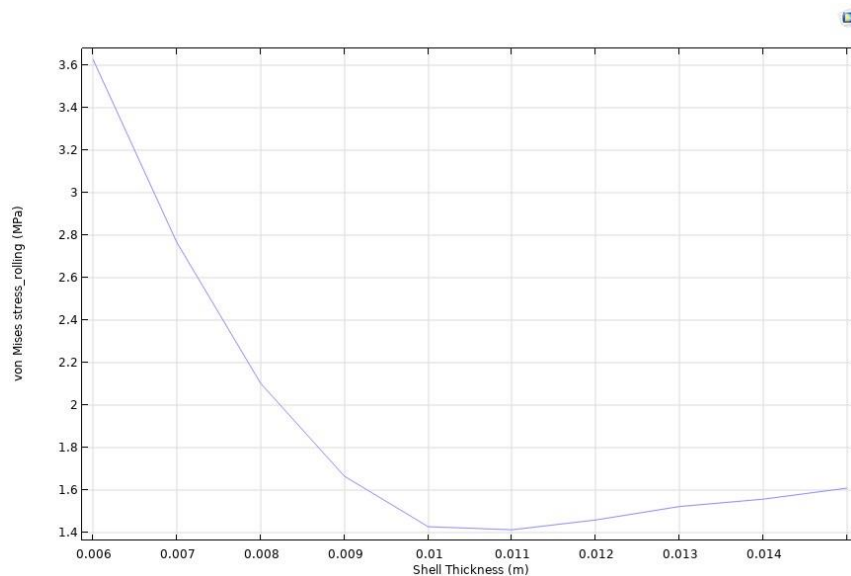


Sphere Shell Thickness vs Max von Mises Stress graph

Before this analysis we designed the sphere as having 6mm thickness but with this graph in mind, it is quite evident that massive stress reductions can be obtained by increasing the thickness up to 0.01 m after which the stress improvements diminish.

Additionally, PETG, one of the principal materials in consideration, has a variable yield stress that is quoted in several material libraries online as being anywhere from 20 MPa all the way to 100 MPa. Hence considering the worst-case scenario of the yield stress being 20 MPa, our shell will yield if a thickness of 6mm is used. A 10mm thickness on the other hand would not only keep our shell safe but also give us a healthy safety factor of 2.

The same analysis was then repeated with the loads being primarily the weight of the shell and all the components it contains (7kg).

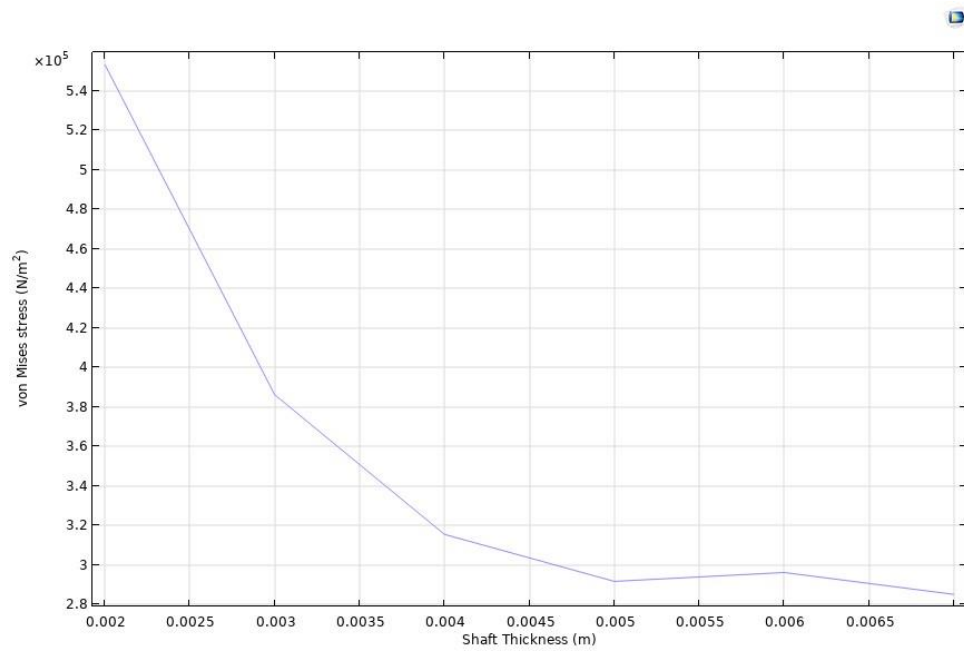


The trend of the results obtained is very similar to that observed in the impact stress study, wherein stress reductions are significant until 0.01 m, after which the improvements appear to diminish completely. Indeed, at first glance, the values of the stresses seem insignificant, peaking at only 3.6 MPa. However, it must be remembered that this stress is repetitive. Every time the sphere revolves, this stress is applied and removed, causing what is called rolling fatigue. The fatigue strength is often far lower than the yield stress, so regardless of whether this seems like a small value, it is important to keep this contact stress in mind.

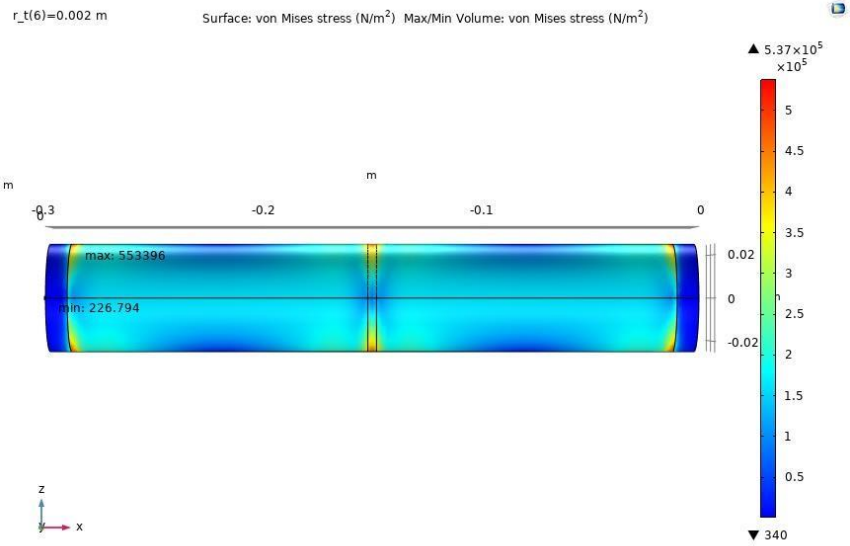
Considering how both the impact stress and the fatigue stress reduce with increasing thickness, we will opt for higher thickness values for the spherical shell wherever possible. However, increasing the shell thickness leads to another undesired consequence: added weight. This is especially important for us, as our spherical robot is intended for amphibious applications and must be light enough to float in water. Therefore, we explored other avenues to reduce the weight.

One such avenue was the shaft design. Initially, the shaft was dimensioned with a diameter of 50 mm and a thickness of 7 mm. The shaft is designed to be stationary and is responsible for bearing the weights of the internal components, estimated to be around 3 kg. The edges of the shaft were fixed, and a total force of 3 kg centered around the midpoint, along with the weight of the shaft itself, was applied.

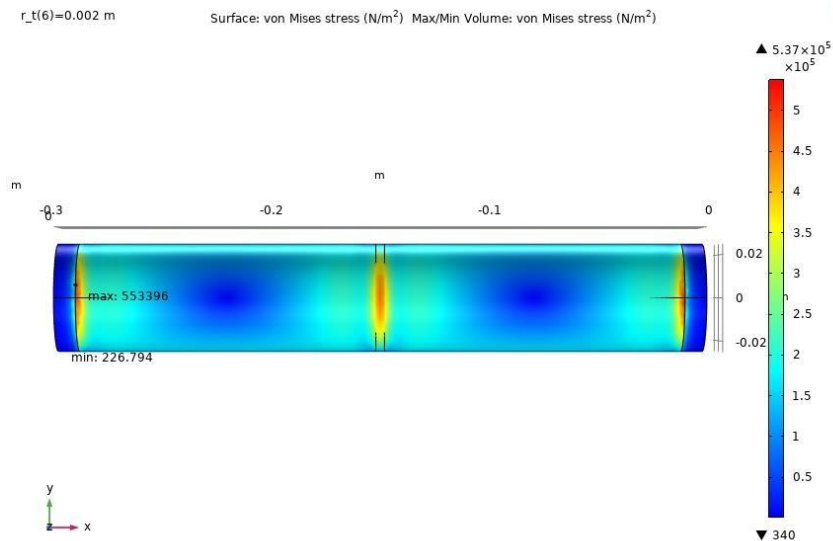
The parametric study of the shaft led to interesting results, wherein the initial stress was very small, nothing that could cause failure. But an even better insight found was that a thickness of 4 mm would provide almost the same strength as the thickness of 7 mm. Hence, we will decrease the thickness of this shaft to maintain the weight of the robot such that it can still float.



Shaft Thickness vs Max von Mises Stress graph



Side View of the Shaft under loading Condition



Top View of the Shaft under loading Condition

MOTOR CALCULATIONS

Equation 3

$$Power = v(t) \left[a(t) \left(m + \frac{J}{R} \right) + \mu N \right]$$

The dynamics of our spherical 3D printed robot were modeled using a governing ordinary differential equation (ODE) that captures its motion in various scenarios. This equation encompasses three key terms, each representing distinct aspects of the robot's behavior and energy requirements.

1. Linear Acceleration Term: The first term in the ODE accounts for linear acceleration, quantifying the energy necessary to overcome the inertia of the robot

as it accelerates along a straight path. In other words, this term describes how the robot's velocity changes over time when it moves in a linear direction. It takes into consideration factors such as the mass of the robot and the external forces acting upon it to determine the rate of change of linear velocity.

2. **Rotational Acceleration Term:** The second term in the equation addresses rotational acceleration, representing the energy required to overcome the rotational inertia of the robot as it changes its orientation or spins about its axis. This term reflects how the robot's angular velocity evolves over time, accounting for its rotational dynamics. Factors such as the moment of inertia and torque influence the rate of change of angular velocity captured by this term.
3. **Friction Term:** The final term in the ODE accounts for friction, which must be overcome by the robot when it is stationary or in motion. Friction arises due to the interaction between the robot and its environment, hindering its movement to varying degrees. Whether the robot is rolling along a surface or remaining stationary, frictional forces oppose its motion and contribute to the overall dynamics captured by the equation.

By combining these three terms, the ODE provides a comprehensive framework for understanding and predicting the motion of our spherical 3D printed robot under different conditions. It allows us to analyze how the robot's velocity and orientation evolve over time in response to external forces and constraints, facilitating the optimization of its design and control strategies for enhanced performance

As can be seen the ODE is dependent on the acceleration function. Now in order to reach the nature of this function one must realize that in real life as experienced in cars, while running, cycling that there is always a top speed. The speed at which the acceleration drops to zero. It is also experienced practically that the acceleration peaks in the rest, right after

rest. So surely it is not a linear function, it has to be some sort of a parabola, or exponentially decaying function.

Naturally one would also assume that the function of acceleration is dependent upon time and rightfully so. However, according to Iyer et al. “acceleration is not strictly a function of time or distance at a traffic intersection”. In reality it is a dependent on much more than time: the terrain, bad drivers etc. are all contributing factors. But that dependence can be negated in the aims of achieving a simple model dependent on time alone.

According to Iyer et al. [2] the acceleration of a vehicle at an intersection such as a traffic spot can be modelled by the following relation. We chose to follow the same relation for our spherical robot:

Equation 4

$$\lambda = \frac{\ln\left(\frac{a_{max}}{a_{min}}\right)}{X}$$

Equation 5

$$a(x) = a_{max} e^{-\lambda x}$$

It initially appears that this function is not dependent on time but rather x , the distance travelled. However the distance travelled x is also a function of time and as such, the function is indeed dependent on time along with the constants a_{max} and a_{min} which are the max and minimum values of the vehicle while x is the interchange distance across which the vehicle has to accelerate from rest to its maximum speed. Max and min values of a and x were assumed based on similar terrestrial vehicles mentioned in the same research paper.

The polar moment of inertia of a hollow spherical shell was calculated using the following formula

Equation 6

$$j_{shell} = \frac{2 * s_mass}{5} \times \frac{0.125^5 - (0.125 - t)^5}{0.125^3 - (0.125 - t)^3}$$

Power required by the robot was related to torque using the following equation:

Equation 7

$$Power = Torque \times \omega$$

MODELLING THE ROBOTS' DYNAMICS IN MATLAB SIMULINK

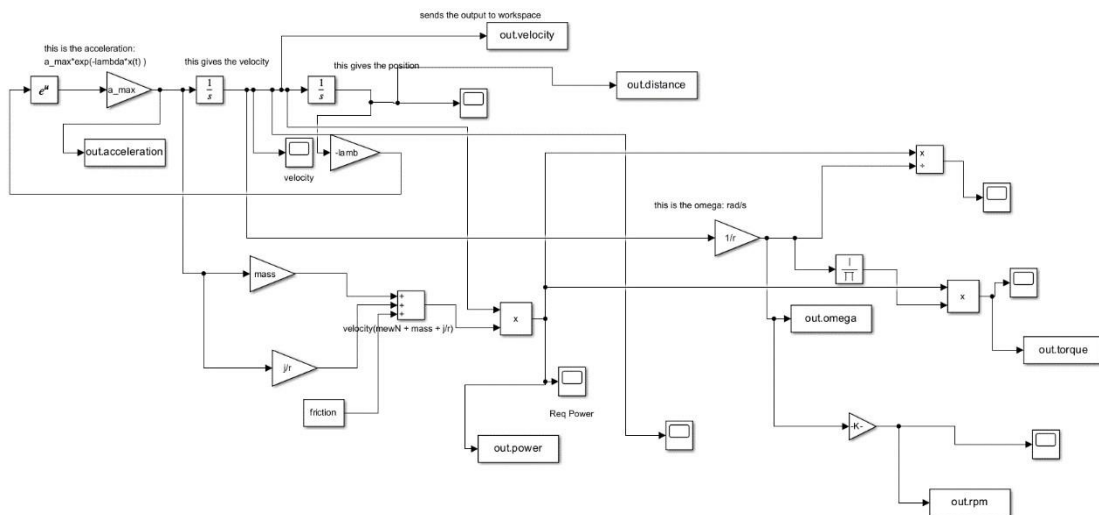


Figure 23 Simulink Model

In our quest to optimize the performance of the spherical 3D printed robot, we turned to Simulink to simulate and analyze the dynamic behavior of the system. This involved modeling the governing equations discussed earlier within the Simulink environment to gain crucial insights into the torque, velocity, and acceleration profiles of the robot.

By encapsulating the differential equations governing the robot's dynamics into Simulink blocks and connecting them appropriately, we created a dynamic simulation model. This model allowed us to input various parameters such as mass, moment of inertia, external forces, and friction coefficients and observe their effects on the robot's motion.

1. **Torque Profile:** The simulation provided a torque profile that outlines the torque exerted on the robot's propulsion system over time. This profile is essential for understanding the torque requirements for driving the robot and ensuring that the selected motors can deliver the necessary torque for propulsion.
2. **Velocity Profile:** Additionally, the simulation yielded a velocity profile illustrating how the robot's velocity evolves over time under different operating conditions. This profile offers valuable insights into the robot's speed capabilities and helps assess its performance across various scenarios.
3. **Acceleration Profile:** The simulation also generated an acceleration profile, elucidating how the robot's acceleration changes over time. This profile is crucial for evaluating the robot's maneuverability and agility, guiding decisions related to control algorithms and motion planning.

By analyzing these profiles, we determined the maximum torque and required idle speeds to operate our robot at the desired speeds and accelerations.

In essence, Simulink proved to be a powerful tool for simulating the dynamics of our spherical 3D printed robot, enabling us to derive actionable insights and make informed decisions regarding motor selection and system optimization.

```

%code in ex with some changes
a_max=1; %maximum acceleration required
a_min=0.5; %minimum acceleration, both used in acc profile eq
tn=0.006; %thickness of the sphere, i think
X=5.128; %in metre=17ft % refer to research paper, but from what i remember it is the intersection length
r=0.125; %radius of the sphere in m
h=0.75*r; %the fraction of radius submerged in water. in this case 75%
dens_wa=997.77 %kg per m^3;
dens_material=123; % in kg/m^3
mass=1/3*pi*h^2*(3*r - h)*dens_wa %gives the mass of water causing buoyant force
lamb=log(a_max/a_min)/X; %just a const used in calc profile
s_mass=4/3*pi*(r^3-(r-tn)^3)*dens_material %mass of the sphere alone, function of radius, thickness and material density.
j=(2*s_mass/5) * ((0.125)^5-(0.125-tn)^5)/(((0.125)^3-(0.125-tn)^3)) %moment of inertia of sph, it
%radius of sphere
j_2=2/3*s_mass*r^2 %this is another formula of moment of inertia of thin sphere just for checking
friction= 0.1 * mass * 9.81 %frictional force
%plot((out.omega./2*pi)*60, out.torque.*10.1971621)
plot((out.omega./2*pi)*60, out.torque/2)
hold on
xlabel('RPM');
ylabel('Torque (Nm)');
title('Torque vs Rpm curve for one Motor');
hold off
plot((out.omega./2*pi)*60, out.power/2)
xlabel('RPM');
ylabel('Power (watts)');

```

Figure 24 MATLAB Code

The provided code snippet serves a dual purpose: it supplies crucial constants and model parameters to the Simulink model while also performing buoyancy calculations. The comments within the code explain the function of each defined parameter. Additionally, the variable "mass" represents the mass required to submerge the sphere by 75% in water.

MODEL RESULTS

Maximum velocity of robot: 3.7 m/s

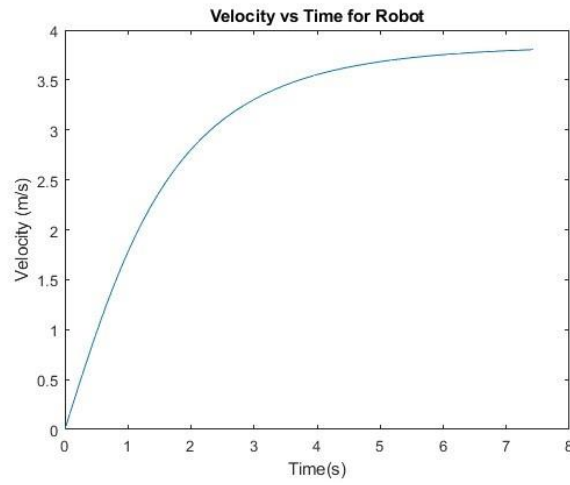


Figure 25 Velocity time Curve

Maximum acceleration: 1m/s²

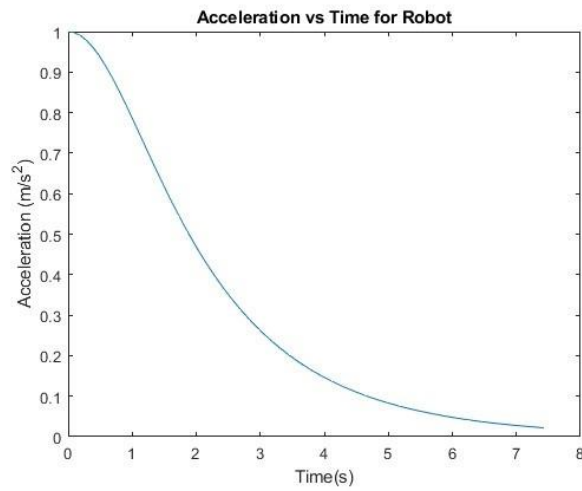


Figure 26 Acceleration time Curve

Maximum Torque: 0.67N/m

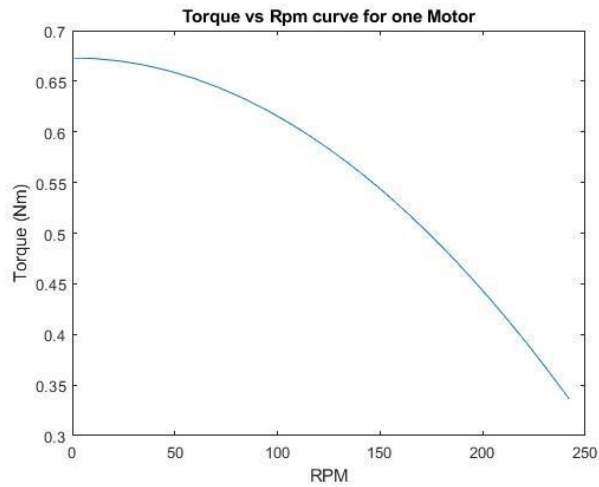


Figure 27 Torque RPM Curve

Servo motor and bearing calculation and selection

Servo motor: The servo motor is a crucial component of our project as it facilitates the robot's navigation by swinging the attached mass, which includes the pendulum and the battery. The selected servo has a rating of 25 kgcm, whereas our calculated requirement is 18 kgcm.

Load rating: $\text{mass} \times \text{height} = 2\text{kg} \times 9\text{cm} = 18\text{kgcm}$



Bearing calculation and selection

6805 bearing was selected for keeping motors in contact with the spheres and its mount to ensure proper torque transfer from motor's shaft to sphere and causing it to maneuver in forward and backwards direction. Similarly, 608zz bearing was selected for camera mounting and we used the same method for its calculation with its respective data acquired through analysis. Following is the MATLAB code which we utilized to compute appropriate bearing for function.

```

bearing(1).m  x  +
1  %bearing selection
2
3  %initial values guessed
4  X=0.56; Y=1.5; %as per nsk catalogue
5  V = 1; %as we selected inner sliding ring bearing
6  weight= 5.3*9.81;
7  Fr=weight/4.44822; %Fr= radial load = mass*g and divided by 4.44822 to get force into lbf
8  Fa= 0; %Fa= axial load(in our case it approx. equals to zero)
9  Fe= (X*V*Fr)+(Y*Fa);
10 ka=1; %precision gearing
11 kr=1; %90 reliability
12
13 l=8000; %life in kh as per shigleys manual (intermittent service)
14 rpm=400; %motor rpm provided
15 sec=60; %sec in a minute
16 L= 1*rpm*sec;
17 Lo=10^6;
18 a=3; %for ball bearing
19 C10=ka*Fe*((L/(kr*Lo))^1/a);
20
21 % from NSK catalogue, we selected 6806 series bearing for a motor with dia
22 % 30mm
23 Co=770; Cr=1010;
24 e=Fa/Co;
25 %from the initial guess we took values based on combined loading, now
26 %taking only radial loading into account
27
28 X_=1;
29 Fe_ = X_*V*Fr;
30 C10_ = ka*Fe_*((L/(kr*Lo))^1/a);
31
32 %if C10_ is greater than C10 of the NSK catalogue; then our assumed bearing is
33 %correct

```

CONTROL SYSTEM WITH ARDUINO

Arduino Uno

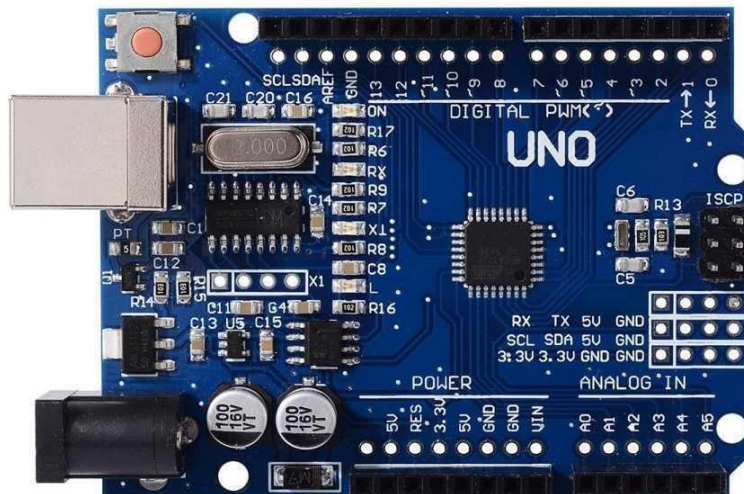


Figure 28 Arduino Board

Integrating Arduino Uno as the central controller for the Smart Rolling Robot's propulsion and stabilization systems is a strategic choice tailored to the specific needs and functionalities of our design. Arduino Uno's versatility and robustness make it the perfect

match for orchestrating the complex interplay between the two brushless DC motors driving the robot's shell RPM and the servo responsible for adjusting the pendulum to offset the robot's center of mass and facilitate turning maneuvers.

With Arduino Uno at the helm, developers have access to a wealth of resources, from its extensive library of motor control algorithms to its vast community of enthusiasts and experts. This allows for the seamless implementation of precise RPM control for the shell's brushless DC motors, enabling the robot to achieve optimal speed and agility across diverse terrain, both on land and in water.

Moreover, Arduino Uno's intuitive programming interface empowers developers to craft sophisticated control strategies for the servo-driven pendulum system. By dynamically adjusting the pendulum's position, the Smart Rolling Robot can effectively shift its center of mass, enhancing stability and maneuverability during turns and ensuring smooth navigation through challenging environments.

The synergy between Arduino Uno and the Smart Rolling Robot's propulsion and stabilization systems not only streamlines development but also enhances performance and reliability. Arduino Uno's compatibility with a wide range of sensors and communication modules further expands the robot's capabilities, enabling advanced functionalities such as obstacle detection, environmental sensing, and wireless communication.

In essence, Arduino Uno serves as the backbone of our robot's control architecture, driving innovation and enabling the realization of our vision for a versatile and agile disaster response solution. With Arduino Uno's support, the Smart Rolling Robot stands ready to revolutionize flood surveys and disaster response operations, offering rapid deployment, precise control, and unmatched reliability in the face of adversity.

```
BALL_DRIVER.ino
67
68
69
70 if (bluetooth == 'B')
71 {
72   backward();
73   myservo.write(pos);
74 }
75
76
77 if (bluetooth == 'L')
78 {
79   forward();
80   myservo.write(pos1);
81 }
82
83 if (bluetooth == 'R')
84 {
85   forward();
86   myservo.write(pos2);
87 }
88 if (bluetooth == 'S')
89 {
90   stop_robot();
91   myservo.write(pos);
92 }
93
94 if (bluetooth == 'U')
95 {
96   lpwm = rpwm + 1;
97   delay(1000);
98   EEPROM.write(30, lpwm);
99   Serial.print("X");
100  Serial.print(lpwm);
101 }
102 if (bluetooth == 'u')
```

```
BALL_DRIVER.ino
34 }
35
36 int lpwm = 255;
37 int rpwm = 255;
38 void loop() {
39   // put your main code here, to run repeatedly:
40   lpwm = EEPROM.read(30);
41   rpwm = EEPROM.read(40);
42   pos1 = EEPROM.read(40);
43   pos2 = EEPROM.read(60);
44
45   analogWrite(enA, lpwm);
46   analogWrite(enB, rpwm);
47
48   bluetooth = Serial.read();
49
50   Serial.println(bluetooth);
51
52   delay(100);
53
54   if (bluetooth == 'F')
55   {
56     forward();
57     myservo.write(pos);
58   }
59   if (bluetooth == 'L')
60   {
61     left();
62   }
63
64   if (bluetooth == 'R')
65   {
66     right();
67   }
68
69 }
```

BALL_DRIVER.ino

```
1  #include <EEPROM.h>
2  char bluetooth = 0;
3  #define enA 6// SPEED CONT
4  #define in1 9
5  #define in2 10
6  // Motor B connections
7  #define enB 4// SPEED CONT
8  #define in3 5
9  #define in4 7
10
11 #include <Servo.h>
12
13 Servo myservo; // create servo object to control a servo
14 // twelve servo objects can be created on most boards
15
16 int pos = 90; // variable to store the servo position
17 int pos1 = 65;
18 int pos2 = 110;
19
20 void setup() {
21     Serial.begin(9600);
22     // put your setup code here, to run once:
23     pinMode(enA, OUTPUT);
24     pinMode(enB, OUTPUT);
25     pinMode(in1, OUTPUT);
26     pinMode(in2, OUTPUT);
27     pinMode(in3, OUTPUT);
28     pinMode(in4, OUTPUT);
29     myservo.attach(12);
30
31     myservo.write(pos);
32     analogWrite(enA, 0);
33     analogWrite(enB, 0);
34 }
35
36 int lpwm = 255;
```

```
BALL_DRIVER.ino
100 }
101 }
102 if (bluetooth == 'u')
103 {
104     lpwm = lpwm - 1;
105     delay(1000);
106     EEPROM.write(30, lpwm);
107     Serial.print("X!");
108     Serial.print(lpwm);
109 }
110 if (bluetooth == 'D')
111 {
112     rpwm = rpwm + 1;
113     delay(1000);
114     EEPROM.write(20, rpwm);
115     Serial.print("Y!");
116     Serial.print(rpwm);
117     Serial.println();
118 }
119 if (bluetooth == 'd')
120 {
121     rpwm = rpwm - 1;
122     delay(1000);
123     EEPROM.write(20, rpwm);
124     Serial.print("y!");
125     Serial.print(rpwm);
126     Serial.println();
127 }
128 ///////////////////////////////////////////////////////////////////
129 if (bluetooth == 'M')
130 {
131     pos1 = pos1 + 1;
132     delay(1000);
133     EEPROM.write(40, pos1);
134     Serial.print("M!");
135     Serial.print(pos1);
136 }
137 }
138 if (bluetooth == 'm')
139 {
140     pos1 = pos1 - 1;
141     delay(1000);
142     EEPROM.write(40, pos1);
143     Serial.print("m!");
144     Serial.print(pos1);
145     Serial.println();
146 }
147 }
148 if (bluetooth == 'N')
149 {
150     pos2 = pos2 + 1;
151     delay(1000);
152     EEPROM.write(60, pos2);
153     Serial.print("N!");
154     Serial.print(pos2);
155     Serial.println();
156 }
157 if (bluetooth == 'n')
158 {
159     pos2 = pos2 - 1;
160     delay(1000);
161     EEPROM.write(60, pos2);
162     Serial.print("n!");
163     Serial.print(pos2);
164     Serial.println();
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
```

```
BALL_DRIVER.ino
167 }
168
169 void forward() {
170   digitalWrite(in1, HIGH);
171   digitalWrite(in2, LOW);
172   digitalWrite(in3, HIGH);
173   digitalWrite(in4, LOW);
174 }
175
176 // Function to go backward
177 void backward() {
178   digitalWrite(in1, LOW);
179   digitalWrite(in2, HIGH);
180   digitalWrite(in3, LOW);
181   digitalWrite(in4, HIGH);
182 }
183
184
185 // Function to go Right
186 void right() {
187   digitalWrite(in1, HIGH);
188   digitalWrite(in2, LOW);
189   digitalWrite(in3, LOW);
190   digitalWrite(in4, HIGH);
191 }
192
193 }
194
195 // Function to go left
196 void left() {
197   digitalWrite(in1, LOW);
198   digitalWrite(in2, HIGH);
199   digitalWrite(in3, HIGH);
200   digitalWrite(in4, LOW);
201 }
202
203
204 // Function to stop
205 void stop_robot() {
206   digitalWrite(in1, LOW);
207   digitalWrite(in2, LOW);
208   digitalWrite(in3, LOW);
209   digitalWrite(in4, LOW);
210 }
211
212
213
214
215
216 // Function to go backward
217 void backward() {
218   digitalWrite(in1, LOW);
219   digitalWrite(in2, HIGH);
220   digitalWrite(in3, LOW);
221   digitalWrite(in4, HIGH);
222 }
223
224
225 // Function to go Right
226 void right() {
227   digitalWrite(in1, HIGH);
228   digitalWrite(in2, LOW);
229   digitalWrite(in3, LOW);
230   digitalWrite(in4, HIGH);
231 }
232
233 }
234
235 // Function to go Left
236 void left() {
237   digitalWrite(in1, LOW);
238   digitalWrite(in2, HIGH);
239   digitalWrite(in3, HIGH);
240   digitalWrite(in4, LOW);
241 }
242
243
244 // Function to stop
245 void stop_robot() {
246   digitalWrite(in1, LOW);
247   digitalWrite(in2, LOW);
248   digitalWrite(in3, LOW);
249   digitalWrite(in4, LOW);
250 }
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Internal Circuitry Code in Arduino
MANUFACTURING PROCESS: BUILDING THE
SMART ROLLING ROBOT



Figure 29 3D Printer

Manufacturing the Smart Rolling Robot involves several key steps. First, we'll prepare the CAD file for 3D printing, ensuring it's optimized for the process. This means making sure the design is suitable for printing and will result in strong, functional parts.

Next, we'll choose the right materials for the job. We need ones that are strong, durable, and resistant to water since the robot will be used in flood-prone areas. Once we've selected the materials, we'll move on to the printing process.

Using 3D printing techniques like Fused Deposition Modeling (FDM), we'll build the robot's parts layer by layer. After printing, we'll perform finishing touches such as sanding and painting to ensure both aesthetics and functionality.

The next step involves the assembly phase, where we'll integrate all the printed components, along with motors, electronics, and sensors. Testing becomes paramount during this stage to ensure the proper functioning of all components.

Throughout the process, we'll maintain a focus on scalability and iteration, aiming to swiftly implement any necessary changes or enhancements.

Ultimately, our goal is to develop a dependable and efficient solution tailored for disaster response scenarios.

DATA ACQUISITION

TRANSFERRING VIDEO FEED WITH ESP32 CONTROLLERS:



Figure 30 ESP32-CAM

The integration of ESP32 controllers represents a significant advancement in the Smart Rolling Robot's capabilities, enabling seamless transmission of live video feed from the onboard camera to remote operators or monitoring stations. This innovative solution leverages the ESP32's powerful features, including dual-core processing, Wi-Fi connectivity, and low-power consumption, to facilitate real-time video streaming in diverse and challenging environments.

1. Onboard Camera Integration:

The Smart Rolling Robot is equipped with a high-definition camera strategically positioned to capture a comprehensive view of its surroundings. The camera feed serves as a critical source of visual information for remote operators, enabling them to assess the situation and make informed decisions in real-time.

2. ESP32 Controller Configuration:

ESP32 controllers are configured to establish a robust Wi-Fi connection between the Smart Rolling Robot and remote monitoring stations. The controllers leverage their

advanced networking capabilities to ensure stable and reliable data transmission, even in bandwidth-constrained or congested environments.

3. Real-Time Video Streaming:

Using the ESP32 controllers, the Smart Rolling Robot streams live video feed over the Wi-Fi network to designated monitoring stations or mobile devices. The video stream provides operators with a real-time perspective of the robot's surroundings, facilitating remote navigation, obstacle detection, and situational awareness.

4. Low-Latency Communication:

The ESP32 controllers facilitate low-latency communication between the Smart Rolling Robot and remote operators, ensuring minimal delay between video capture and display. This near-instantaneous feedback enables operators to respond promptly to changing conditions and execute precise maneuvers as needed.

5. Adaptive Bitrate Streaming:

To optimize bandwidth utilization and ensure consistent video quality, the ESP32 controllers support adaptive bitrate streaming. This dynamic encoding mechanism automatically adjusts the video bitrate based on network conditions, delivering smooth and uninterrupted video playback across varying connection speeds.

6. Secure Data Transmission:

ESP32 controllers implement robust encryption protocols to secure data transmission between the Smart Rolling Robot and remote monitoring stations. Advanced encryption algorithms safeguard the integrity and confidentiality of the video feed, protecting sensitive information from unauthorized access or tampering.

7. Integration with Monitoring Software:

The video feed streamed by the Smart Rolling Robot's ESP32 controllers is seamlessly integrated into monitoring software interfaces, providing operators with intuitive controls and visual feedback. Customizable dashboards and overlays enhance situational awareness, enabling operators to effectively navigate and interact with the robot in realtime.

Conclusion:

The incorporation of ESP32 controllers for video feed transmission represents a pivotal advancement in the Smart Rolling Robot's capabilities, enhancing its effectiveness and versatility in disaster response scenarios. By leveraging the ESP32's robust networking capabilities, the robot can provide remote operators with vital visual information, enabling rapid decision-making and response coordination in critical situations.

For its usage, we utilized two 3.7-volt battery cells to power the ESP32 controllers and the camera. Through the terminal, we provided power input to the ESP32 camera power ports and connected to Wi-Fi, enabling real-time video feed access via an IP address assigned by the Arduino Uno. The code uploaded into the ESP32 via the ESP32 CAM MB development board ensures the proper functioning of the system.

```
CameraWebServer_copy_20230515100256.ino  app_httpd.cpp  camera_index.h  camera_pins.h
1  #include "esp_camera.h"
2  #include "mdns.h"
3
4  //
5  // WARNING!! PSRAM IC required for VGA resolution and High JPEG quality
6  // Ensure ESP32 sensor module or other board with PSRAM is selected
7  // Partial images will be transmitted if image exceeds buffer size
8  //
9  // You must select partition scheme from the board menu that has at least 1MB APP space.
10 // Face Recognition is DISABLED for ESP32 and ESP32-S2, because it takes up from 15
11 // seconds to process single frame. Face Detection is ENABLED if PSRAM is enabled as well
12
13 // =====
14 // Select camera model
15 // =====
16 // #define CAMERA_MODEL_SHOVOR_KIT // Has PSRAM
17 // #define CAMERA_MODEL_ESP_EYE // Has PSRAM
18 // #define CAMERA_MODEL_ESP32S1_EYE // Has PSRAM
19 // #define CAMERA_MODEL_NOSTACK_PSRAM // Has PSRAM
20 // #define CAMERA_MODEL_NOSTACK_S2_PSRAM // PSRAM version B has PSRAM
21 // #define CAMERA_MODEL_NOSTACK_HDMI // Has PSRAM
22 // #define CAMERA_MODEL_NOSTACK_ESP32CAM // No PSRAM
23 // #define CAMERA_MODEL_NOSTACK_UNITCAM // No PSRAM
24 // #define CAMERA_MODEL_AI_THINKER // Has PSRAM
25 // #define CAMERA_MODEL_TIGER_ES8258 // No PSRAM
26 // #define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
27 // ** Expressif internal boards **
28 // #define CAMERA_MODEL_ESP32_CAM_BOARD
29 // #define CAMERA_MODEL_ESP32S2_CAM_BOARD
30 // #define CAMERA_MODEL_ESP32S3_CAM_LCD
31
32 #include "camera_pins.h"
33
34 // =====
35 // Enter your WiFi credentials
36 // =====
```

```

CameraWebServer_copy_20230515160256.ino  app_http.cpp  camera_index.h  camera_pins.h
34 // -----
35 // Enter your wifi credentials
36 // -----
37 const char* ssid = "TP-Link_6230";
38 const char* password = "59950677";
39
40 void startCameraServer();
41 void setUpLedFlash(int pin);
42
43 void setup() {
44   Serial.begin(115200);
45   Serial.setDebugOutput(true);
46   Serial.println();
47
48   Camera config { config;
49   config.ledc_channel = LEDC_CHANNEL_0;
50   config.ledc_timer = LEDC_TIMER_0;
51   config.pin_05 = V2_GPIO_NUM;
52   config.pin_d1 = V3_GPIO_NUM;
53   config.pin_d2 = V8_GPIO_NUM;
54   config.pin_05 = V9_GPIO_NUM;
55   config.pin_d4 = V6_GPIO_NUM;
56   config.pin_05 = V7_GPIO_NUM;
57   config.pin_06 = V8_GPIO_NUM;
58   config.pin_07 = V9_GPIO_NUM;
59   config.pin_xclk = XCLK_GPIO_NUM;
60   config.pin_pclk = PCLK_GPIO_NUM;
61   config.pin_vsync = VSYNC_GPIO_NUM;
62   config.pin_href = HREF_GPIO_NUM;
63   config.pin_sccb_sda = S10C_GPIO_NUM;
64   config.pin_sccb_scl = S10C_GPIO_NUM;
65   config.pin_pwdn = PWDN_GPIO_NUM;
66   config.pin_reset = RESET_GPIO_NUM;
67   config.sclk_freq_hz = 2000000;
68   config.frame_size = FRAMESIZE_UXGA;
69   config.pixel_format = PIXFORMAT_JPEG; // for streaming

```

```

CameraWebServer_copy_20230515160256.ino  app_http.cpp  camera_index.h  camera_pins.h
69   config.frame_size = FRAMESIZE_UXGA;
70   config.pixel_format = PIXFORMAT_JPEG; // for streaming
71   //config.grab_mode = CAMERA_GRAB_WHEN_READY; // for face detection/recognition
72   config.grab_mode = CAMERA_GRAB_WHEN_READY;
73   config.fb_location = CAMERA_FB_IN_PSRAM;
74   config.jpeg_quality = 12;
75   config.fb_count = 1;
76
77   // If PSRAM is present, init with UXA resolution and higher JPEG quality
78   // for larger pre-allocated frame buffer.
79   if(config.pixel_format == PIXFORMAT_JPEG){
80     if(psramFound()){
81       config.jpeg_quality = 10;
82       config.fb_count = 2;
83       config.grab_mode = CAMERA_GRAB_LATEST;
84     } else {
85       // Limit the frame size when PSRAM is not available
86       config.frame_size = FRAMESIZE_SVGA;
87       config.fb_location = CAMERA_FB_IN_DRAM;
88     }
89   } else {
90     // Best option for face detection/recognition
91     config.frame_size = FRAMESIZE_240X240;
92     #if CONFIG_ID_TARGET_ESP32S3
93       config.fb_count = 2;
94     #endif
95   }
96
97   #if defined(CAMERA_MODEL_ESP_32C)
98     pinMode(13, INPUT_PULLUP);
99     pinMode(14, INPUT_PULLUP);
100   #endif
101
102   // camera init
103   esp_err_t err = esp_camera_init(&config);

```

```

CameraWebServer_copy_20230515160256.ino  app_http.cpp  camera_index.h  camera_pins.h
101 // camera init
102 esp_err_t err = esp_camera_init(&config);
103 if (err != ESP_OK) {
104   Serial.println("camera init failed with error 0x%x", err);
105   return;
106 }
107
108 sensor_t * s = esp_camera_sensor_get();
109 // Initial sensors are flipped vertically and colors are a bit saturated
110 if (s->id.PID == OV3660_PID) {
111   s->set_vflip(s, 1); // Flip it back
112   s->set_brightness(s, 1); // up the brightness just a bit
113   s->set_saturation(s, -2); // lower the saturation
114 }
115 // drop down frame size for higher initial frame rate
116 if(config.pixel_format == PIXFORMAT_JPEG){
117   s->set_framesize(s, FRAMESIZE_QVGA);
118 }
119
120 #if defined(CAMERA_MODEL_M5STACK_HIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
121   s->set_vflip(s, 1);
122   s->set_hmirror(s, 1);
123 #endif
124
125 #if defined(CAMERA_MODEL_ESP32S3_EYE)
126   s->set_vflip(s, 1);
127 #endif
128
129 // Setup LED Flash if LED pin is defined in camera_pins.h
130 #if defined(LED_GPIO_NUM)
131   setUpLedFlash(LED_GPIO_NUM);
132 #endif
133
134 WiFi.begin(ssid, password);
135 WiFi.setSleep(false);

```

```
CameraWebServer copy_20230515160256.ino  esp8266:199 camera.ino.h camera_pins.h
120 #if defined(CAMERA_MODEL_POSTACK_WIDE) || defined(CAMERA_MODEL_POSTACK_ESP32CAM)
121   s->set_vflip(s, 1);
122   s->set_hmirror(s, 1);
123 #endif
124
125 #if defined(CAMERA_MODEL_ESP3253_EYE)
126   s->set_vflip(s, 1);
127 #endif
128
129 // Setup LED Flash if LED pin is defined in camera_pins.h
130 #if defined(LED_GPIO_NUM)
131   setupLedFlash(LED_GPIO_NUM);
132 #endif
133
134 WiFi.begin(ssid, password);
135 WiFi.setSleep(false);
136
137 while (WiFi.status() != WL_CONNECTED) {
138   delay(500);
139   Serial.print(".");
140 }
141 Serial.println("");
142 Serial.println("WiFi connected");
143
144 startCameraServer();
145
146 Serial.print("Camera Ready! Use 'http://");
147 Serial.print(WiFi.localIP());
148 Serial.println("' to connect");
149
150 }
151
152 void loop() {
153   // Do nothing, everything is done in another task by the web server
154   delay(10000);
155 }
```

ESP 32 Code in Arduino

Battery Selection

Powering Motors with Lithium Polymer Batteries:

In the heart of the Smart Rolling Robot lies a crucial element: lithium polymer (LiPo) batteries. These powerhouses ensure the robot operates reliably and continuously, vital for

its demanding tasks. Let's delve into why LiPo batteries are the perfect choice for this critical role.

Advantages of LiPo Batteries:

High Energy Density: LiPo batteries pack a punch in a compact, lightweight package, providing ample power to keep the robot going strong.

Lightweight Design: Keeping weight down is key for a nimble robot like ours. LiPo batteries help keep the Smart Rolling Robot light on its wheels, enhancing its agility and maneuverability.

High Discharge Rates: When it's time to spring into action, LiPo batteries deliver the goods. Their high discharge rates mean the robot can accelerate quickly and respond dynamically to changing conditions.

Fast Recharge Times: Time is of the essence in disaster situations. LiPo batteries boast fast recharge times, ensuring the robot can quickly get back into action after a brief pit stop.

Compatibility with Motor Controllers: Working seamlessly with our motor controllers, LiPo batteries ensure smooth and reliable power delivery to the motors, keeping the robot running smoothly.

Safety Considerations: While LiPo batteries offer impressive performance, safety is paramount. Proper handling and precautions are essential to ensure the robot operates safely and reliably.

With LiPo batteries at its core, the Smart Rolling Robot is equipped to tackle the toughest challenges in flood surveys and disaster response. Their combination of lightweight design, high performance, and fast recharge times make them the perfect power source for our agile and versatile robot, ensuring it can operate efficiently and confidently in dynamic environments.

Installation of cameras:

The installation of cameras poses a significant challenge due to the rolling motion of the robot's body. It is imperative to establish a stationary point for camera support, preventing the camera from rolling along with the body. To address this issue, a specialized joint is required to maintain the camera's stationary position relative to the moving ball. A bearing is employed to suspend the cameras with substantial weight, allowing the weight to slide over the surface of the rolling ball. This configuration ensures that the camera remains stationary despite the rolling motion of the body.

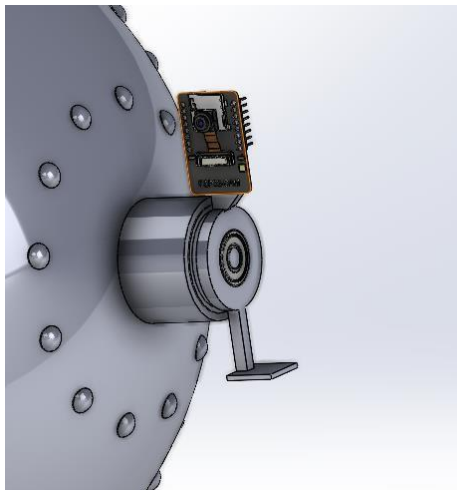


Figure 31 Camera Mount

Waterproofing:

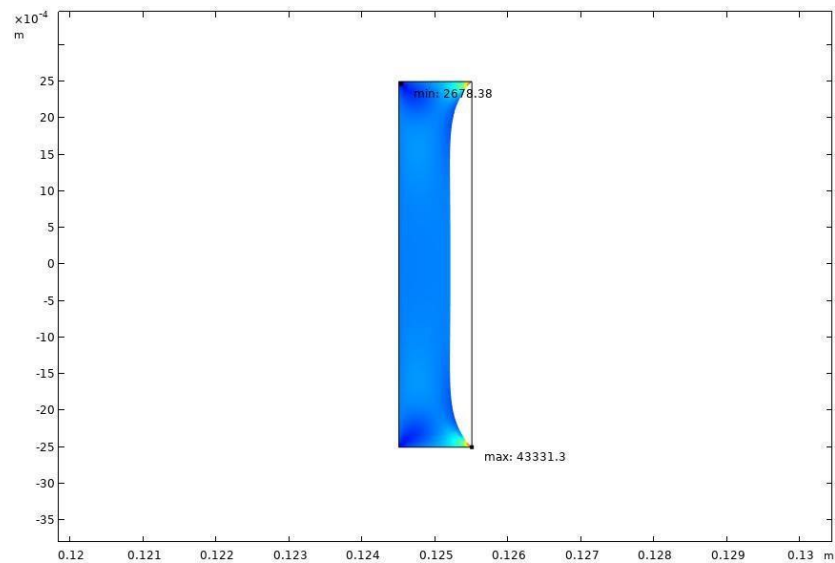
To waterproof the ball, it is essential to seal any possible joints where there is a risk of water leakage. To achieve this, a 3d printed seal of PETG material is applied in between the closing lid of the spheres to ensure waterproofing and an additional measure of waterproofing is taken and that is silicone glue is applied to seal all potential joints, even those with a minimal chance of water leakage.

We also performed FEA analysis and checked whether our designed product is capable of enduring pressure and stress while keeping the spherical robot waterproof and internal components intact.

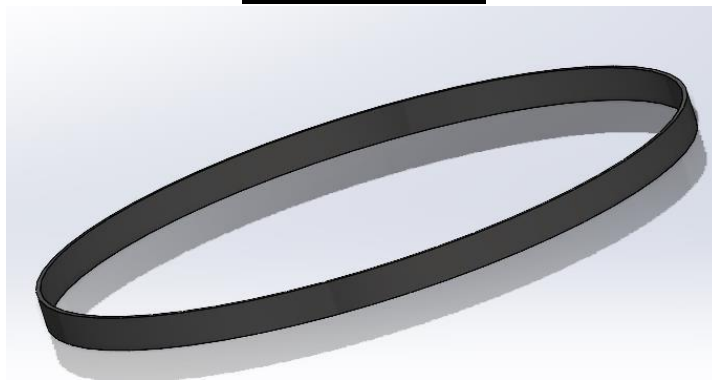
The force exerted on the ring is calculated using the equation $F = \frac{1}{2}\rho Av^2$.

This force is determined to be 0.36352 N at a speed of 5 km/h, indicating the pressure the sealing ring must withstand during operation.

The deformation at that force applied is also very less which is good indication of using this sealing ring in our spherical robot.



Sealing Ring FEA



Sealing Ring

Balancing of spherical robot:

The spherical robot is self-balancing due to its spherical nature and the reason being, the concentrated mass is applied on its center of gravity and that aids the system to remain balanced on land and on water.

CHAPTER 4: RESULTS AND DISCUSSION INITIAL DESIGN

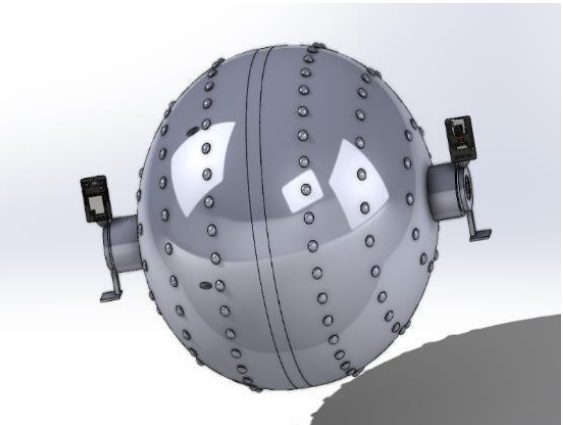
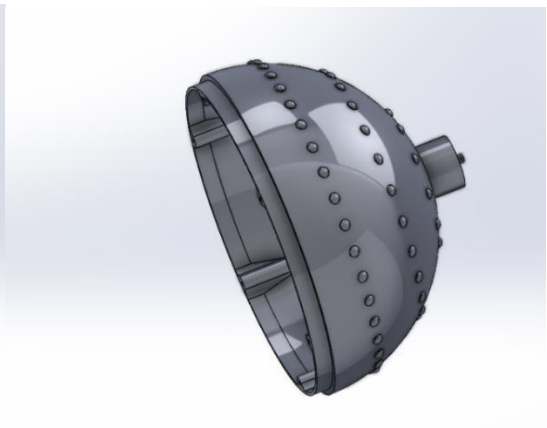
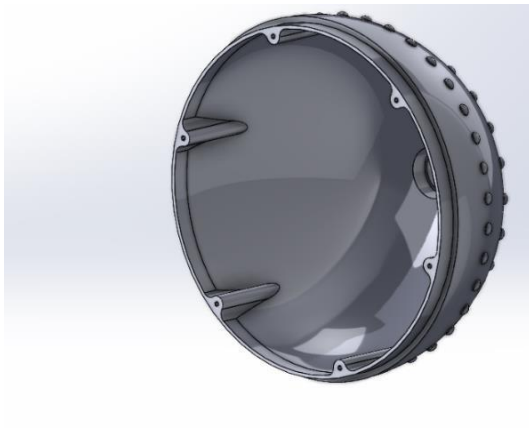
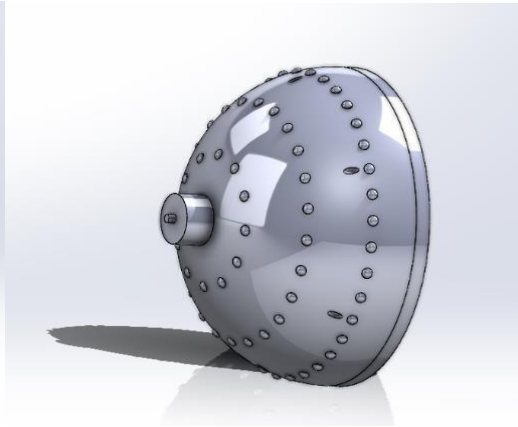
The Components of Spherical Robot

Following is the breakdown of the components that constitutes our project:

- 1- **Spheres:** The spheres are the main part of our project. Spheres are the outer structure that contains the internal driving unit and propagates itself in forward, backward direction and can steer left and right according to the desire of operator. The small dome shaped tread patterns are for better maneuverability in water and muddy areas. Additionally, both the halves of the spheres include threaded holes, these will be used for mating of the two halves of the shells.

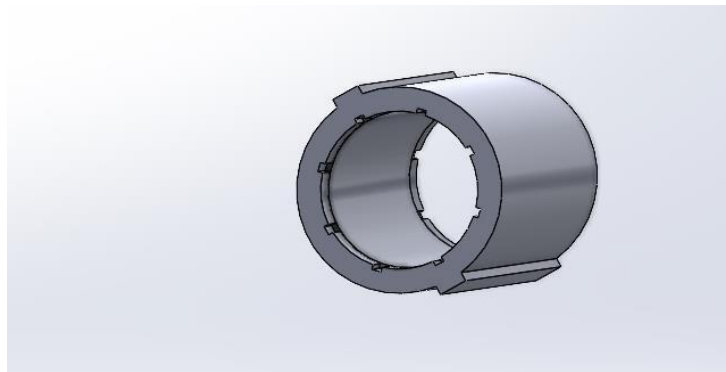
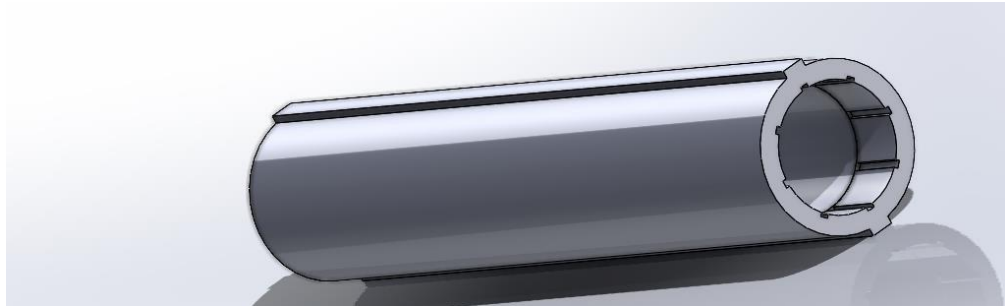
Initially a diameter of 300mm was chosen since the weight required for partial submersion was optimal, allowed us enough head room to add extra equipment inside the sphere shell. Additionally, the camera was at a decent enough height with this diameter as well. The thickness was initially chosen at 6mm. However, as mentioned before the FEA analysis done revealed that theoretically this thickness was not enough in cases of impact and contact stresses.

Initial Dimensions have not been mentioned

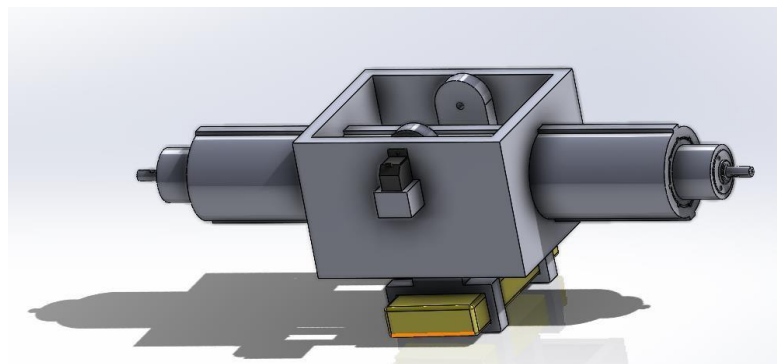
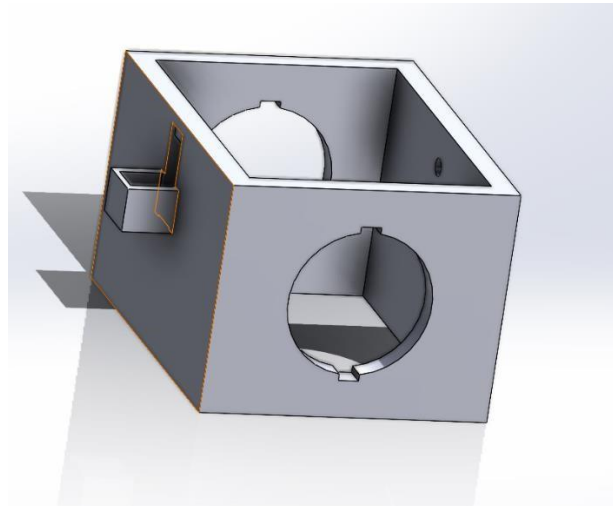


2- **Shaft:** The shaft carries the main load and by its assistance the main assembly doesn't rotate by the motion of the motor. There is a lock in which the motor shaft fits, and it rotates the spheres which is our main goal. Secondly it also houses the motor mounting in it and has key holes pattern engraved onto it to prevent slippage. Thirdly, the shaft itself has keyway pattern onto its top and bottom in order to prevent slippage when fitted across the internal box.

Initially the shaft diameter was chosen at 50mm which would allow us sufficient headroom to easily mount the camera modules and also allow us to fit a powerful enough DC motor for robot propulsion. The shaft thickness was initially chosen as 7mm, however as preceding FEA analysis showed this offered a lot more strength than required and it was decided to choose a lower diameter.



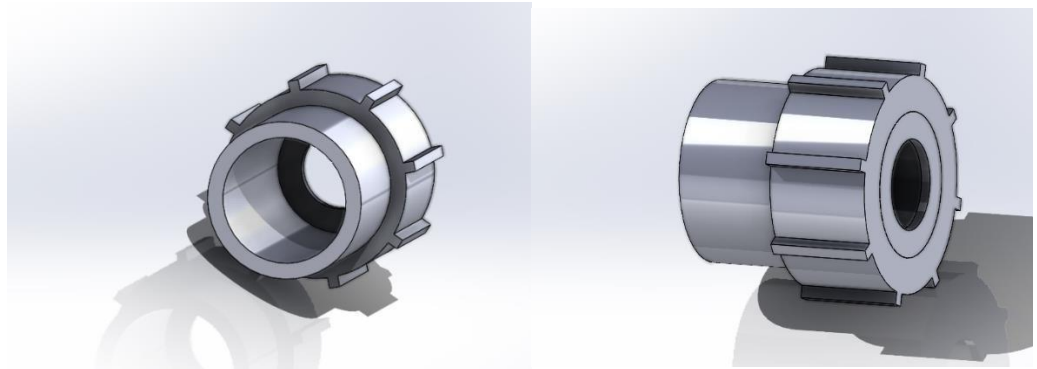
- 3- **Internal Box:** Internal box holds our main assembly, such as controller, pendulum, servo, battery, shaft and maintains the balance of the robot's internal driving mechanism



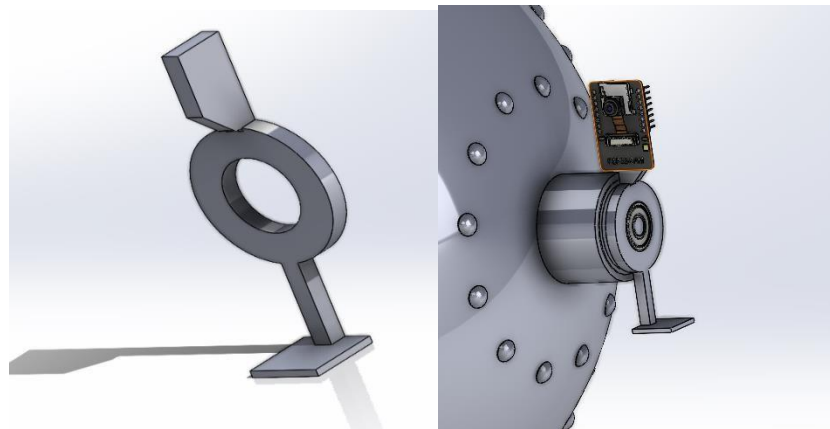
The internal box was 130*100 (width*breadth) mm in dimensions and 80 mm in length. The thickness was 8 mm. The FEA analysis results were showing satisfactory results of this box to bear load as it was the main compartment of the ball carrying all the necessary components.

- 4- **Motor housing:** DC motor is fitted in the motor housing with keyways pattern engraved onto it and mounted inside the shaft to prevent slippage and keep the

assembly in contact with the whole interval driving system to provide the acquired torques to the spheres only. It has hole at its end to let the wires from motor to go to the controller board.



- 5- **Camera Mount:** The camera mount is very simple and is fitted on the 6082z bearing onto most either side of the spheres. There is a stand for camera and for battery to held straight by the mass of battery



- 6- **Servo motor:** Servo motor is very crucial part of our project as it helps the robot to navigate left or right by swinging the mass attached to it (i.e. pendulum and battery). The servo selected rating is 25kgcm and while that of our calculated is 18kgcm.

Load rating: $\text{mass} \times \text{height} = 2\text{kg} \times 9\text{cm} = 18\text{kgcm}$

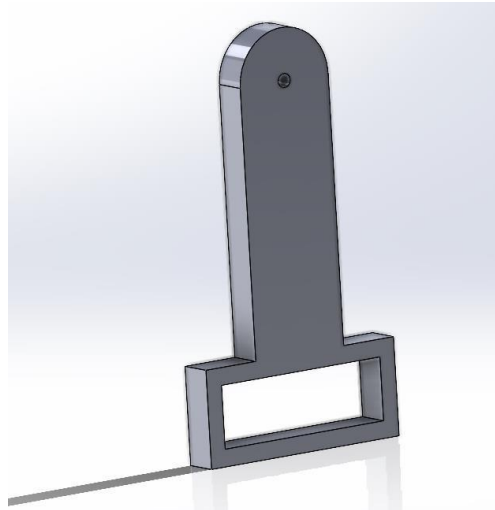


- 7- **Bolts and screws:** M12 nuts and bolts will be utilized as they are much reliable in strength and of smaller size to fasten the spheres with each other.



- 8- **Pendulum:** The pendulum is of very simple design and as for the mass, we will attach the battery which will act as counter mass and will help the spherical robot

to tilt to either left or right direction according to the operator via the swing of servo motor.



- 9- **6806 Bearing:** This bearing is mounted onto the motor and under the hood of the sphere in order to transmit the torque by the motor's shaft to the sphere and prevent the motor body to rotate in order to keep the internal driving system constant.



Design Selection:

For a spherical body, we have chosen a design having an offset center of mass can impact its motion, especially if the body is rolling. The offset center of mass (Bary Center Offset) introduces a torque that can cause the spherical body to start rolling. This is analogous to the way a cylinder or a wheel with an offset mass distribution can initiate rolling motion when placed on a surface.

In summary, having an offset center of mass in a spherical body can contribute to its ability to sustain rolling motion, which can be advantageous in various applications, including robotics or mechanical systems.

A mechanism where the center of gravity of a ball can be adjusted forward or backward to control its rolling motion by the torque applied by the motor. This adjustment is achieved by using DC geared motors mounted on opposite faces of the ball, with their shafts facing inside of the ball. The shafts of both motors are connected to sphere that rotates in tandem with the rotation of the motor shafts.

This design suggests a dynamic system where the movement of the pendulum, driven by the motors, affects the distribution of weight or center of gravity within the ball. By the torque applied from motors to the sphere body itself from inside (locking mechanism of motor shaft with the sphere inside cavity), the ball can be manipulated to roll either forward or backward, offering a method of directional control.

Turning mechanism selection:

We have drafted a design in solid works and in this design, a pendulum will hang in the center of the ball with a fixed shaft. The mass can be rotated to the desired angle using a servo motor, enabling the ball to make turns.

Material selection:

In material selection, we have to 3d print our spherical robot components and following is a comparison of available material along with its conclusion of finalizing our project's material selection which sets in accordance with our project's requirements:

Comparison of 3D Printing Materials for Outdoor and Submersible Applications:

1. ABS (Acrylonitrile Butadiene Styrene):

- **Mechanical Properties:** ABS offers good strength and impact resistance, making it suitable for outdoor use where durability is essential.
- **Corrosion Resistance:** While ABS is generally resistant to moisture and mild chemicals, it may degrade over time when exposed to prolonged water submersion.
- **Waterproofing:** ABS can be post-processed with sealants or coatings to improve waterproofing, but may require additional measures to prevent water ingress.
- **UV Stability:** ABS is susceptible to UV degradation over time, so outdoor exposure should be limited or mitigated with UV-resistant coatings or paints.
- **Buoyancy/Density:** ABS is moderately dense, so careful design considerations are needed to ensure buoyancy and stability when submerged in water.

2. ASA (Acrylonitrile Styrene Acrylate):

- **Mechanical Properties:** ASA offers similar strength and impact resistance to ABS, with improved weatherability and UV stability.
- **Corrosion Resistance:** ASA exhibits excellent resistance to moisture and weathering, making it highly suitable for outdoor applications.
- **Waterproofing:** ASA inherently resists water absorption, providing superior waterproofing without the need for additional sealing or coating.
- **UV Stability:** ASA is highly UV-stable, maintaining its color and mechanical properties even after prolonged exposure to sunlight.
- **Buoyancy/Density:** ASA has a density similar to ABS, requiring similar considerations for buoyancy and stability in water.

3. PETG (Polyethylene Terephthalate Glycol):

- **Mechanical Properties:** PETG offers good strength and impact resistance, comparable to ABS, with enhanced flexibility and transparency.
- **Corrosion Resistance:** PETG exhibits excellent resistance to moisture and chemicals, making it suitable for outdoor and submersible applications.
- **Waterproofing:** PETG is inherently water-resistant and can maintain its properties when submerged in water without additional waterproofing measures.
- **UV Stability:** PETG has good UV stability, although prolonged exposure to sunlight may cause slight degradation over time.
- **Buoyancy/Density:** PETG has a density similar to ABS and ASA, requiring similar considerations for buoyancy and stability in water.

4. PLA (Polylactic Acid):

- **Mechanical Properties:** PLA offers moderate strength and stiffness, suitable for lightweight applications but less durable than ABS, ASA, or PETG.
- **Corrosion Resistance:** PLA is susceptible to degradation in moist environments and may not be suitable for prolonged outdoor use or submersion in water.
- **Waterproofing:** PLA has poor water resistance and may absorb moisture, leading to swelling and degradation over time when exposed to water.
- **UV Stability:** PLA has limited UV stability and may degrade when exposed to sunlight, necessitating UV-resistant coatings for outdoor applications.
- **Buoyancy/Density:** PLA has a density similar to ABS, ASA, and PETG, but its poor water resistance may affect buoyancy and stability in water.

Conclusion:

Based on the comparison, ABS, ASA, and PETG emerge as more suitable materials for outdoor and submersible applications compared to PLA due to their superior durability, weatherability, and resistance to moisture and UV degradation. This was further validated by Dr Usman Bhutta's master students research, who specifically conducted drop testing of spherical balls using PETG and PLA and found that PETG was indeed the superior material. Considering these factors, market availability and pricing we have chosen PETG as our material of choice for 3d printing the most critical components of our project. The stress bearing components such as the spherical shell, along with the shaft and its associated parts were decided to be printed using PETG. The internal components that would not bear as much stress such as the pendulum etc. were printed using the cheaper PLA material.

This is also the material we used in our CAD model and the material we performed stress testing on using finite element analysis.

MOTOR SELECTION

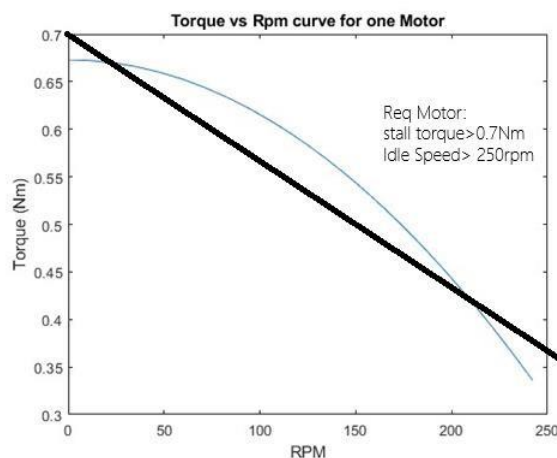
Procedure for Selecting DC Motors

1. **Define Requirements:** Begin by articulating the precise performance requirements of the robot, encompassing parameters such as velocity, acceleration, and torque specifications.
2. **Dynamic Simulation:** Employ Simulink to create a detailed model of the robot's dynamics, generating torque, velocity, and acceleration profiles across a spectrum of operating conditions.
3. **Specification Identification:** Analyze the generated profiles to discern the requisite peak torque and speed parameters necessary for appropriate motor selection.
4. **Criteria Establishment:** Establish definitive selection criteria, encompassing pivotal factors like stall torque, rated speed, and dimensional compatibility.
5. **Motor Exploration:** Engage in an exhaustive exploration of available DC motors that align with the established criteria, leveraging resources such as datasheets, online catalogs, and specialized motor selection tools.
6. **Option Evaluation:** Conduct a thorough assessment of viable motor options, evaluating their characteristics through the lens of torque-speed curves, efficiency metrics, and compatibility with the robot's system architecture.
7. **Comparative Analysis:** Undertake a comprehensive cost-benefit analysis to systematically compare the performance attributes and associated costs of diverse motor alternatives.
8. **Selection and Procurement:** Elect the most fitting motor option based on the culmination of evaluation criteria and acquisition it from a reputable supplier or manufacturer.
9. **Integration and Testing:** Integrate the selected motor seamlessly into the robot's design framework and proceed to conduct rigorous testing to validate its performance across varied operational scenarios.

10. Optimization Iterations: Implement iterative optimization measures, refining control parameters and system configurations as warranted to enhance the motor's performance efficacy.
11. Documentation: Methodically document the entirety of the motor selection process and resultant outcomes for archival purposes, facilitating future reference and replication endeavors.

Following this we selected Robomaster's M2006 P36 (DC) as our primary driving motors. The most important parameter to consider when choosing motors is the torque vs rpm curve, which becomes crucial. The chosen motors must satisfy this curve. Obviously, we cannot expect a motor's operating curve to fall directly on the one that we require but it should envelop our torque vs rpm curve. Operating curves are also seldom available for motors, but most motors can safely be assumed to have a linear operation relation dependent on their rated stall torque and idle speed.

Once again going back to our torque vs rpm curve mentioned in chapter 2 figure 27 and editing it, we want a motor somewhat in the following operating conditions (stall torque > 0.7Nm and idle speed > 250rpm):

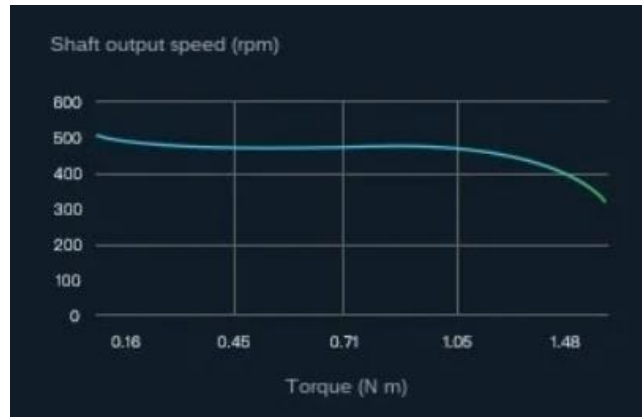


Considering our requirements, we selected Robomaster's M2006 P36 (DC) as our primary driving motors. Since it exceeded both the idle speed and the stall torque requirements. Additionally, it also fit well into our design parameters. Our shaft being about 40mm in

internal diameter, the motor had to be smaller than this value. This specific motor fits that condition.

Robo Master-M2006 power system parameters	Rated voltage	24V
	Cut-in speed continuous	500rpm
	maximum torque 1N · m	1N · m
	maximum speed operating	416rpm
	environment temperature	0-55°C
RoboMaster M2006 P36 Brushless DC Motor Motor retarding device parameters	Weight	90g
	Outer diameter	24.4 mm
	Total length	64.8mm
	Reduction ratio	36:1
	Output shaft	Type D with tapping hole

Tech Specs of the RoboMaster M2006.

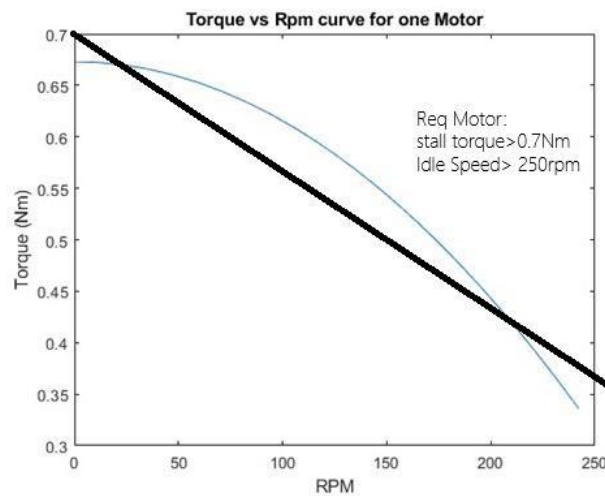


CHAPTER 5: REVISED DESIGN

MOTOR SELECTION

The previous motors chosen were brushless DC motors from an international company that were to be bought and imported from abroad. They cost an accumulative amount of 80-90000 pkr alone. (Brushless DC motors enhance the robot's functionality by providing higher efficiency, precise control, higher power-to-weight ratio, lower maintenance requirements, and quieter operation compared to traditional brushed motors). With the original budget stripped off, we now had to switch to the simple brushed local DC motors that matched our torque and rpm requirements as closely as possible. Again, the graph made before was consulted once again and the closest motor that satisfied both the cost and was close to the specification constraints was the JGA25-371 motor.

The data sheet attached below, highlighted in red shows the motor we are using. The no load rpm and stall torque are both lower than required, but it is the best we can do right now



JGA25-371 Data sheet 2

Model: JGA25-371		Data Sheet										
Voltage		No Load		Load				Stall		Reducer		Weight
Workable	Rated	Speed	Current	Speed	Current	Torque	Output	Torque	Current	Ratio	Size	Unit
Range	Volt/V	rpm	mA	rpm	mA	kg.cm	W	kg.cm	A	1:00	mm	g
6-24V	12	977	46	781	300	0.11	1.25	0.55	1	4.4	15	99
6-24V	12	463	46	370	300	0.23	1.25	1.1	1	9.28	17	99
6-24V	12	201	46	168	300	0.53	1.25	2.65	1	21.3	19	99
6-24V	12	126	46	100	300	0.85	1.25	4.2	1	34	21	99
6-24V	12	95	46	76	300	1.1	1.25	5.5	1	45	21	99
6-24V	12	55	46	44	300	1.95	1.25	9.7	1	78	23	99
6-24V	12	41	46	32	300	2.5	1.25	12.5	1	103	23	99
6-24V	12	25	46	20	300	4.2	1.25	21	1	171	25	99
6-24V	12	19	46	15	300	5.6	1.25	28	1	226	25	99
6-24V	12	11	46	8.8	300	9.45	1.25	47	1	378	27	99
6-24V	12	8.6	46	6.8	300	12	1.25	60	1	500	27	99

3D PRINTING

The materials selected i.e. PETG and PLA were used to print the robot. The stress bearing components such as the spherical shell, along with the shaft and its associated parts were decided to be printed using PETG. The internal components that would not bear as much stress such as the pendulum etc. were printed using the cheaper PLA material. But, in order to print the parts tolerances were to be added to the initial CAD design, as the constituting material contracts as it solidifies. The extent of this contraction determines how much of an upscale (check if this word “tolerance” is the exact word that is used for 3d material contraction) needs to be added. We found that (insert the source where we got this data from) for PETG an upscale of 1.005 while for PLA an upscale of 1.003 was to be used to cater for the contraction of the material. For interference fit a tolerance of 0, while for loose fit a tolerance of 1 -1.5 mm used.



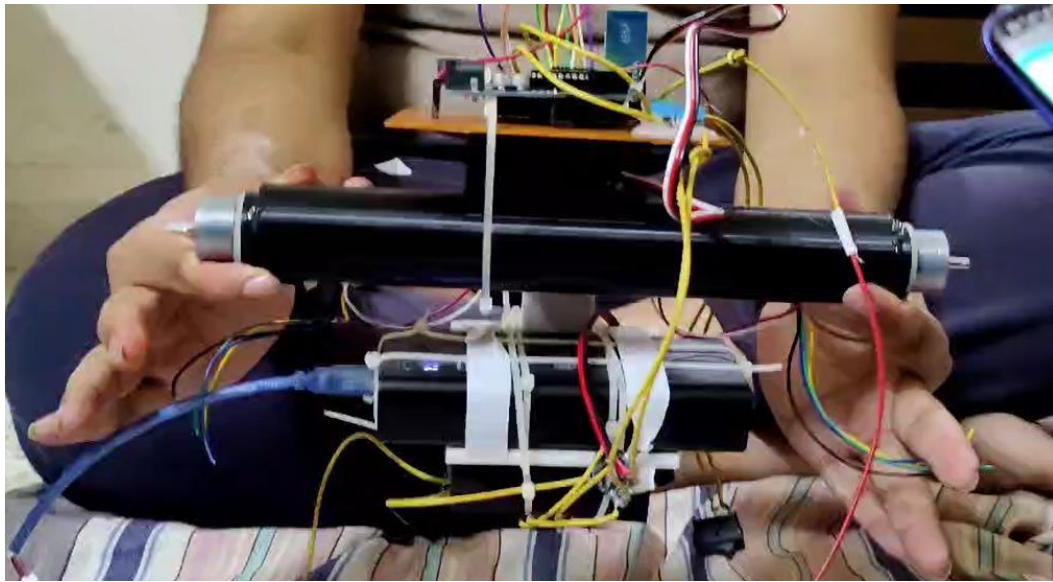
Spherical Shell



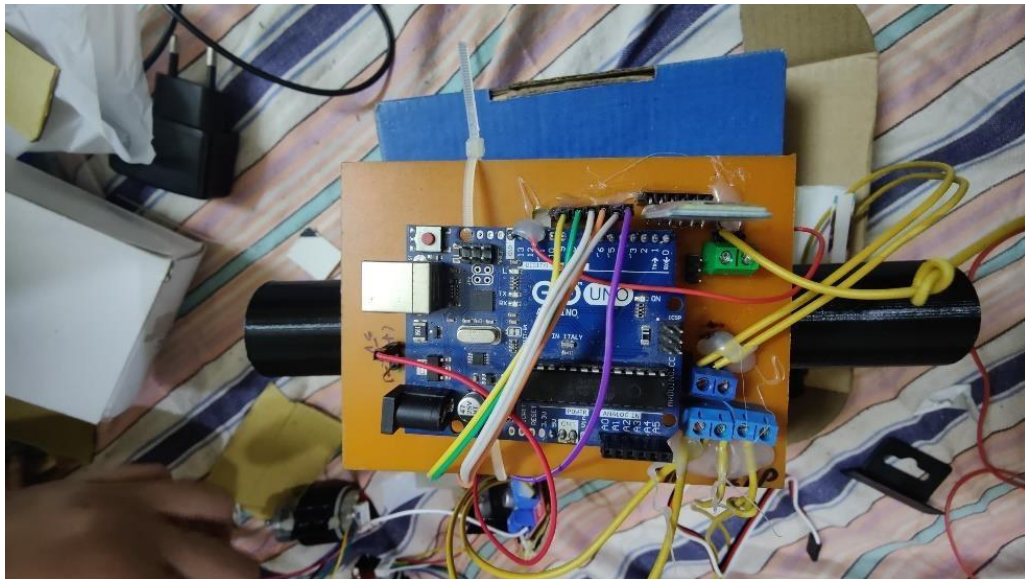
Camera Mount



Motor Housing



Internal Circuitry



Arduino Schematic



Finished Product on Land



Finished Product on Water

Tolerance - Shrinkage in Motor Shaft Cavity

One of the critical challenges faced during the manufacturing process was the occurrence of shrinkage in the cavity intended for the tight fitting of the motor shaft. The shrinkage rendered the initial part incompatible, necessitating the production of another component with altered dimensions to accommodate the motor shaft properly.

Implications:

- The shrinkage problem compromised the intended precision of the motor shaft fit, potentially hindering the robot's maneuverability and functionality.
- Inefficiencies in the manufacturing process arose due to the need to print another part, leading to increased time and resource consumption.

Solution:

To address the shrinkage issue effectively, the manufacturing team reevaluated the design specifications and incorporated adjustments to the dimensions of the cavity region in the sphere. By altering the dimensions, the new part was fabricated to fit snugly around the motor shaft, ensuring optimal functionality and maneuverability of the spherical robot.

Interference - Mismatch with Servo Motor Size

Another significant challenge encountered pertained to interference resulting from a discrepancy between the designed shaft housing for the servo motor and the actual size of the servo motor obtained. The designed shaft, based on dimensions available online, proved incompatible with the larger-than-expected size of the servo motor.

Implications:

- The mismatch between the designed shaft and the actual size of the servo motor posed a significant obstacle in the assembly process, halting progress and necessitating corrective action.
- The interference issue highlighted the importance of accurate component sizing and validation in the design phase to mitigate manufacturing discrepancies.

Solution:

To rectify the interference problem, the manufacturing team undertook a redesign process to adjust the shaft dimensions according to the actual size of the servo motor. By leveraging precise measurements and prototyping, a revised shaft housing was fabricated to accommodate the larger servo motor, ensuring seamless integration and functionality within the spherical robot.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

In the realm of disaster relief applications, the development of innovative robotic solutions plays a pivotal role in enhancing the efficiency and effectiveness of rescue operations. This thesis endeavors to contribute to this domain by proposing a novel spherical robot designed for scenarios where conventional methods, such as drones, may prove insufficient, particularly in detecting individuals trapped under devastated buildings in flooded areas.

The chosen design paradigm for the spherical robot is based on a combination of barycenter offset and pendulum principles. This hybrid approach integrates the strengths of both designs, aiming to achieve optimal performance in navigating complex terrains and providing enhanced mobility for search and rescue missions. The core mechanism involves motors strategically positioned at each end of a shaft, securely fitted within a lock. As the motor shaft initiates rotation, it imparts controlled motion to the spherical robot, enabling it to navigate through challenging environments with precision.

One of the key features of this designed spherical robot is its ability to offer real-time image feeds to the operator. Equipped with advanced imaging technology, the robot serves as an invaluable tool for operators to assess the situation on the ground. This capability proves crucial in identifying potential emergency scenarios, especially when individuals are trapped under rubble or in areas inaccessible to traditional means of surveillance.

The incorporation of the barycenter offset and pendulum-based design not only facilitates seamless movement but also enhances the robot's adaptability to various terrains, including flooded and structurally compromised environments. The dynamic rotation achieved through the motorized shaft ensures that the robot can navigate through confined spaces, providing unprecedented access to areas where human intervention may be challenging.

In conclusion, the proposed spherical robot represents a significant advancement in the field of disaster relief robotics. By combining innovative design principles, including barycenter offset and pendulum-based motion, with cutting-edge imaging technology, this robot has the potential to revolutionize search and rescue operations. The real-time data

transmission to operators empowers them with crucial information, enabling swift decision-making in critical situations. As the sphere of robotics continues to evolve, this research stands at the forefront of leveraging robotic technology for the greater good of humanity, particularly in the face of natural disasters and emergencies.

REFERENCES

- [1] <https://ffc.gov.pk/annual-flood-reports/>

- [2] <https://disasterphilanthropy.org/disasters/2022-pakistan-floods/>
- [3] "Pakistan Floods: The Deluge of Disaster – Facts & Figures as of 15 September 2010 | ReliefWeb". Reliefweb.int. 15 September 2010.
- [4] "Floods worsen, 270 killed: officials". The Express Tribune. 13 September 2011. Retrieved 13 September 2011
- [5] "Pakistan floods 'leave 58 dead, 66,000 affected', says NDMA". The News International. AFP. 5 August 2013. Retrieved 5 August 2013.
- [6] dawn.com/2012/10/17/floods-killed-455-affected-five-million-ndma/
- [7] "India and Pakistan Strain as Flooding Kills Hundreds". The New York Times. 8 September 2014. Retrieved 9 September 2014
- [8] NDMA annual report 2016
- [9] NDMA Monsoon 2019 Daily Situation Report No. 070
- [10] NDMA Monsoon 2022 Daily Situation Report No 158
- [11] NDMA Monsoon 2023 Daily Situation Report No. 97 (30 Sep 2023) [12] P. O'Neill, THE ABC'S OF DISASTER RESPONSE, 2005.
- [13] C. Van Tilburg, First Report of Using Portable Unmanned Aircraft Systems (Drones) for Search and Rescue, Wilderness and Environmental Medicine. 28 (2) (2017) 116–118,
- [14] D. Pokhrel, N. R. Luitel, and S. Das, "Design and Development of a Spherical Robot (SpheRobot)," 2013.
- [15] Z. Guo, T. Li and M. Wang, "A Survey on Amphibious Robots," 2018 37th Chinese Control Conference (CCC), Wuhan, China, 2018, pp. 5299-5304, doi: 10.23919/ChiCC.2018.8483367.

- [16] V. Muralidharan and A. D. Mahindrakar, "Geometric controllability and stabilization of spherical robot dynamics," *IEEE Transactions on Automatic Control*, vol. 60, pp. 2762-2767, 2015.
- [17] Q. Zhan, T. Zhou, M. Chen, and S. Cai, "Dynamic trajectory planning of a spherical mobile robot," in *Robotics, Automation and Mechatronics, 2006 IEEE Conference on*, 2006, pp. 1-6.
- [18] Y. Ming, D. Zongquan, Y. Xinyi, and Y. Weizhen, "Introducing HIT spherical robot: Dynamic modeling and analysis based on decoupled subsystem," in *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, 2006, pp. 181-186.
- [19] J. Qingxuan, Z. Yili, S. Hanxu, C. Hongyu, and L. Hongyi, "Motion control of a novel spherical robot equipped with a flywheel," in *Information and Automation, 2009. ICIA'09. International Conference on*, 2009, pp. 893-898.
- [20] Q. Zhan, Y. Cai, and C. Yan, "Design, analysis and experiments of an omni-directional spherical robot," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 4921- 4926.
- [21] S. Mahboubi, M. M. S. Fakhrabadi, and A. Ghanbari, "Design and implementation of a novel spherical mobile robot," *Journal of Intelligent & Robotic Systems*, vol. 71, pp. 4364, 2013.
- [22] M. Roozegar, M. Mahjoob, M. Esfandyari, and M. S. Panahi, "XCS-based reinforcement learning algorithm for motion planning of a spherical mobile robot," *Applied Intelligence*, vol. 45, pp. 736- 746, 2016.
- [23] Chase R., Pandya A. "A review of active mechanical driving principles of spherical robots", *Robotics*, 2012, vol.1, no.1, pp.3–23.
- [24] Crossley V.A. "A literature review on the design of spherical rolling robots". Pittsburgh, Pa., 2006, 6pp.

- [25] Ylikorpi T., Suomela J. “Ball-shaped robots. Climbing and walking robots: Towards new applications”, H.Zhang (Ed.). Vienna: InTech, 2007. P.235–256.
3757–3762.
- [26] Karavaev Yu.L., Kilin A.A., “The dynamics and control of a spherical robot with an internal omniwheel platform”, Regul. Chaotic Dyn., 2015, vol.20, no.2, pp.134–152.
- [27] Ylikorpi T., “Mobility and Motion Modelling of Pendulum-Driven Ball Decoupled Models Robots: for Steering and Obstacle Crossing”, Doctoral Dissertations, School of Electrical Engineering, 2017, 251 p. ISBN 978-952-60-7616-4
- [28] Michaud, F.; Caron, S. Roball, the rolling robot. Auton. Robots 2002, 12, 211–222.
- [29] Wait, K.; Jackson, P.; Smoot, L. Self Locomotion of a Spherical Rolling Robot Using a Novel Deformable Pneumatic Method. In Proceedings of 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, May 2010; pp.
- [30] Yamanaka, T.; Nakaura, S.; Sampei, M. Hopping Motion Analysis of “Superball” Like Spherical Robot Based on Feedback Control. In Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, NV, USA, October 2003, Volume 4, pp. 3805–3810.
- [31] Sugiyama, Y.; Shiotsu, A.; Yamanaka, M.; Hirai, S. Circular/Spherical Robots for Crawling and Jumping. In Proceedings of 2005 IEEE International Conference on Robotics and Automation (ICRA 2005), Barcelona, Spain, April 2005; pp. 3595–3600
- [32] Au, K.; Xu, Y. Decoupled Dynamics and Stabilization of Single Wheel Robot. In Proceedings of 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’99), Kyongju, Korea, October 1999; Volume 1, pp. 197–203
- [33] Brown, H., Jr.; Xu, Y. A Single-Wheel, Gyroscopically Stabilized Robot. In Proceedings of 1996 IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, April 1996; Volume 4, pp. 3658–3663

- [34] Rotundus. Available online: <http://www.rotundus.se> (accessed on 22 May 2012)
- [35] <https://www.wired.com/2017/01/chinas-launching-drones-fight-back-earthquakes/>
- [36] <https://www.dw.com/en/will-robots-replace-dogs-in-earthquake-rescueoperations/a-64647765>
- [37] Tadokoro, Satoshi. Disaster Robotics: Results from the ImPACT Tough Robotics Challenge. Springer International Publishing, 2019
- [38] H. Kolbari, A. Ahmadi, M. Bahrami and F. Janati, "Impedance Estimation and Motion Control of a Pendulum-Driven Spherical Robot," 2018 6th RSI International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 2018, pp. 6-11, doi: 10.1109/ICRoM.2018.8657621