ProjectToppleBot

A Scalable Robot Architecture for Space Applications

Kilian Olen
Controls and Manufacturing
Embry-Riddle Aeronautical University
Daytona Beach, Florida
olenk@my.erau.edu

William Baker
Electrical and Systems
Embry-Riddle Aeronautical University
Daytona Beach, Florida
bakerw7@my.erau.edu

Gabriel Rodriguez
Software and Communication
Embry-Riddle Aeronautical University
Daytona Beach, Florida
rodrig43@my.erau.edu

Erin Bader
Mechanical Design
Embry-Riddle Aeronautical University
Daytona Beach, Florida
badere@my.erau.edu

Submitted to the Engineering Physics Review Board in partial fulfillment of the requirements for EP 496: Space Systems Design I

at the

EMBRY-RIDDLE AERONAUTICAL UNIVERSITY

Kilian Olen

Principal Investigator / Project Lead…………………………………………………………………..

September 9, 2024

Accepted by………………………………………………………………………………………...

Aroh Barjatya
Professor of Engineering Physics
Director, Space and Atmospheric Instrumentation Lab

EMBRY-RIDDLE
Aeronautical University

Abstract

This paper presents the proposal for TopppleBot, a scalable reaction wheel cube rover with modular components that alters its intended functionality. By developing a standardized workflow for cube rover design, we provide a versatile robotic system that combats overreliance on custom solutions for space exploration. From exploring small celestial bodies to assembling structures on distant planets, ToppleBot possesses strong potential for space applications, all while maintaining a consistent internal architecture. Furthermore, with its simplified design, ToppleBot requires only three motors for complete control, allowing room for redundancy should the mission require. Additionally, Project ToppleBot provides the capability for additional modules to be explored by future senior design projects.

In this paper we explore the prior works in this field, including project CUBEBAS, from which our work stems off of. We then highlight the novel contributions project ToppleBot provides, particularly its standardized design approach, before breaking down the mission requirements we aim to achieve. Individual team roles and a proposed timeline of key deliverables are also presented. Ultimately, ToppleBot aims to advance the field of space robotics by providing a flexible, reliable, and easily adaptable platform for future space missions and exploration.

Table of Contents

1. Introduction

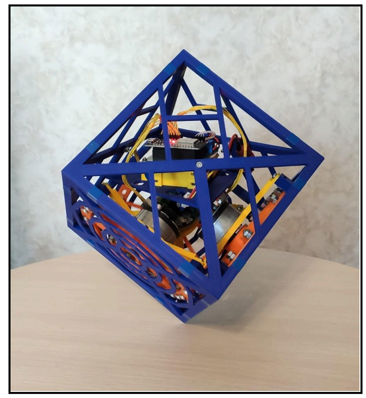


Fig. 1. ReM-RC Balancing Cube.

The CUBEBAS project [1] successfully replicated ReM-RC's open-source design (Fig. 1) [2], demonstrating active stability at both the edges and corners of a cubic structure. Self-balancing cubes present an intriguing control challenge similar to that of the inverted pendulum; however, their application has predominantly remained confined to academic settings. Nevertheless, reaction wheels, such as those implemented in ETH Zurich's Cubli [3], have a well-documented history in attitude control for space missions, evidenced by NASA's Cassini [4] and Hubble spacecraft [5]. Reaction wheels have similarly become a standard for attitude control in CubeSat missions, where stringent space constraints necessitate careful selection of onboard instruments.

Outside of attitude control systems, reaction wheels also see use in the field of mobile robotics. Common applications include improving the dynamic stability of bipedal robots [6], similar to how one shifts one's weight while walking. Quadrupeds, on the other hand, typically utilize such systems to balance across narrow terrains [7], where tight positioning reduces static stability. Although already a suitable solution for stabilization, there has been an increased interest in utilizing reaction wheels solely for locomotion. Cubli previously demonstrated the ability to navigate across flat terrains; however, JPL further explored its potential for space exploration with the concept rover, Hedgehog (Fig. 2.) [8].

Designed to explore small celestial bodies, Hedgehog shares many of the same characteristics as Cubli, with a notable addition of large protrusions acting as shock absorbers on each corner. Compared to wheeled rovers, which consist of multiple moving components, Hedgehog required minimal actuation, with just three motors enabling complete control over each axis of rotation. Coupling its high energy movements with use in microgravity environments, the Hedgehog rover showed potential for improved mobility over traditional systems, which was verified in a series of tests onboard a Zero-G aircraft [9]. Additionally, due to its simple design, the Hedgehog rover could be scaled to accommodate any array of instruments with minimal modification to the existing architecture.

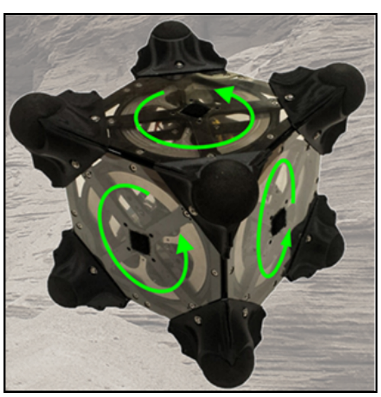


Fig. 2. (Left) JPL Hedgehog Concept Rover. (Right) MIT M-Bots.

While such a solution may not be ideal for navigation in higher gravity environments, researchers at MIT CSAIL are exploring a similar architecture for self-assembling modular robots, aptly named M-Bots (Fig. 2.) [10]. Instead of protrusions, their design incorporates strategically placed magnets to facilitate proper alignment and maintain connections during high-intensity movements. With the structure of each robot being cubical, they possess a rotational symmetry that allows for any two faces to join together regardless of orientation, further emphasizing its promise over alternative modular systems.

Hedgehog demonstrates promising potential as an extraplanetary rover, while the M-Bots present a potential solution for in situ manufacturing. Although they possess two entirely separate functions, their internal mechanisms remain much the same. If one were to design modular components for these systems, one could utilize the same base design for both functions. With the high costs of transporting custom solutions in space, a standardized design capable of performing multiple tasks would not only minimize costs, but also simplify repair procedures.

Given these existing systems, there remains an unexplored potential in reaction wheel cube robots. We propose a scalable workflow to develop a base system on which additional components can be integrated depending on the desired application. We recognize that the variable nature of such components will impact how the system interacts with the environment. We intend to develop an efficient control algorithm for the base system, comparable in performance to Cubli (Fig. 3), from which we can expand to take into account any components that impact its performance.

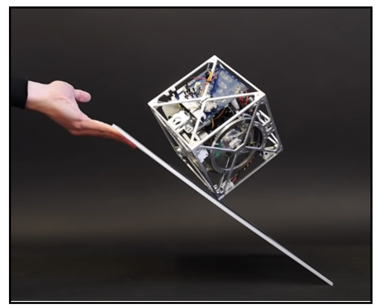


Fig. 3. ETHz Cubli Balancing on a manually adjusted incline.

The following section details the system architecture and control strategies developed by the CUBEBAS team, providing the groundwork for our current mission. We then outline the minimum and comprehensive requirements we aim to achieve, followed by key milestones in the development timeline. Finally, we finish off with individual team responsibilities, which is reinforced with a Letter of Intent from each team member. Through proper planning and effective communication, Project ToppleBot aims to provide a baseline solution from which others can expand upon, promoting a standardized procedure for cube robot design.

2. CUBEBAS Review

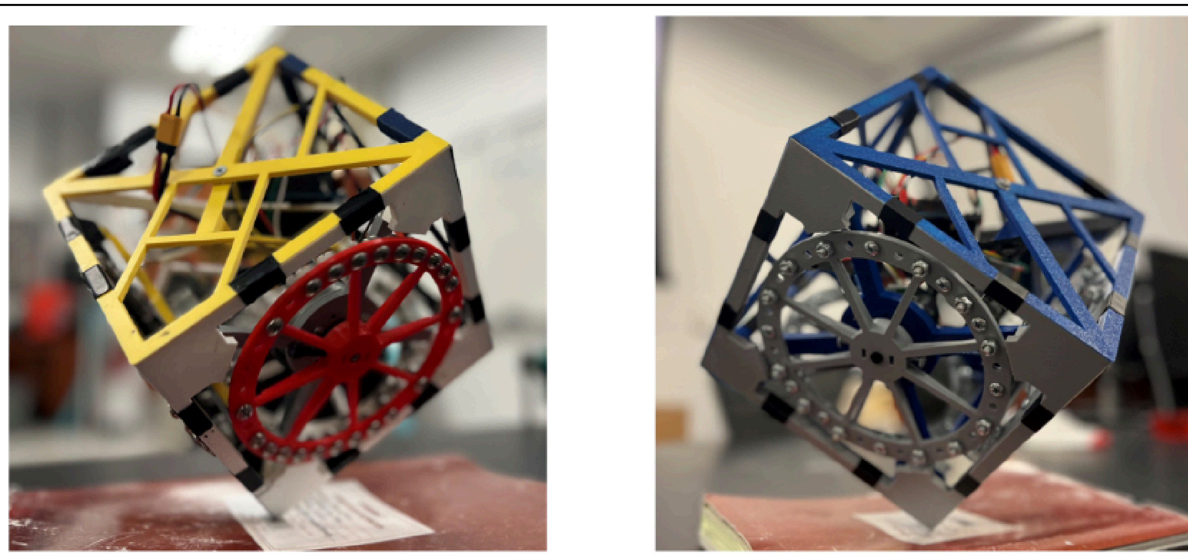
2.1. Structural Review

The CUBEBAS team modeled their self-balancing cube off of an open-sourced project by Rem-RC, whose project can be found on YouTube. The CUBEBAS team had several requirements for the structure. This included securing all the components, maintaining the cube's center of mass, and ensuring the controller mount secures the MPU 6050 flush to the mount. The two cubes, referred to as Big Dude and Little Dude, were 3D printed out of Polylactic acid (PLA). [1]

The first iteration of Little Dude relied heavily on the open-sourced files provided by Rem-RC for its structure; this led to complications. The original design consisted of two sides, motor and non-motor. The motor sides contained the wheel and motor, and the non-motor side contained mounts for the battery, sensors, and controller. In the second iteration of Little Dude, the motor sides were altered from their original circular pattern to an open wall with reinforced corners. This was done to prevent the wheel from snagging during rotation. Other alterations to Little Dude included adjustments to fit different nuts and bolts and address tolerancing issues.

Big Dude is the second version of the cube whose size was increased to be 33% bigger than Little Dude [1]. This was done to allow for more space inside the cube for electrical components. New motor holes were also added to the newer and bigger model in the CUBEBAS project. One of the final changes made between Little Dude and Big Dude is a decrease in bolts, which were used as weights, on the wheels. This was done to limit the moment of inertia in the bigger system.

Overall, Project CUBEBAS was able to structure two functioning cubes that were based on the original open-sourced files from Rem-RC. After a few iterations of the structure, they eventually ended on a PLA 3D-printed outer casing that had empty sides to allow for free movement of the wheels, and mounting for the electronic components in the center of the cube.



(a) Little Dude Balancing (b) Big Dude Balancing

Fig. 4. Image of Both CUBEBAS Cubes Balancing

2.2. Electrical Review

For the electrical review of the CUBEBAS, many hours of work went into the circuitry. The picture below shows their overall block layout for the CUBEBAS logic with the power supply and all the systems it feeds. Additionally, this shows the feedback and communication trails between all the different systems for their robot to work properly.

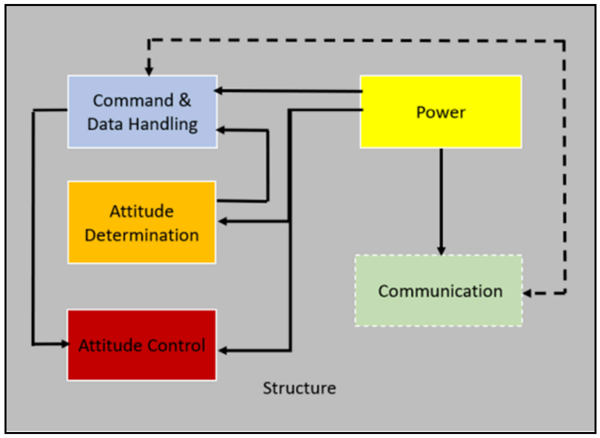


Fig. 5. CUBEBAS Structure Schematic

The parts that they used for the final design were well selected, but some of the original parts had unforeseen issues that caused them some difficulty that they successfully overcame. Thanks to them we know more about what parts are needed and what parts should be adjusted to better suit the needs of our Topplebot. Their final decision for a microcontroller was an ESP32-DevKitC 32E. This controller worked very well for them since it provided all the pinouts needed and was able to run their desired coding software. Additionally, they used MPU–6050 sensors for their attitude control. This chip was chosen since it nicely worked with their chosen microcontroller as well as it being widely used which made the integration into the system much easier. Finally, for their motors, they chose Three Nidec 24H BLDC. These motors worked for them, but they mentioned that these motors were not the optimal choice due to circuitry difficulties.

They designed a PCB capable of accommodating the microcontroller, all sensor connections, battery supply, motor driver, and voltage regulator. Additionally, they produced a detailed computer-generated layout illustrating the connections between the components. Creating a precise and well-structured PCB layout is a crucial step for our project to ensure that the circuitry is well-organized and secure. The image below displays the computer schematic with all the component connections.

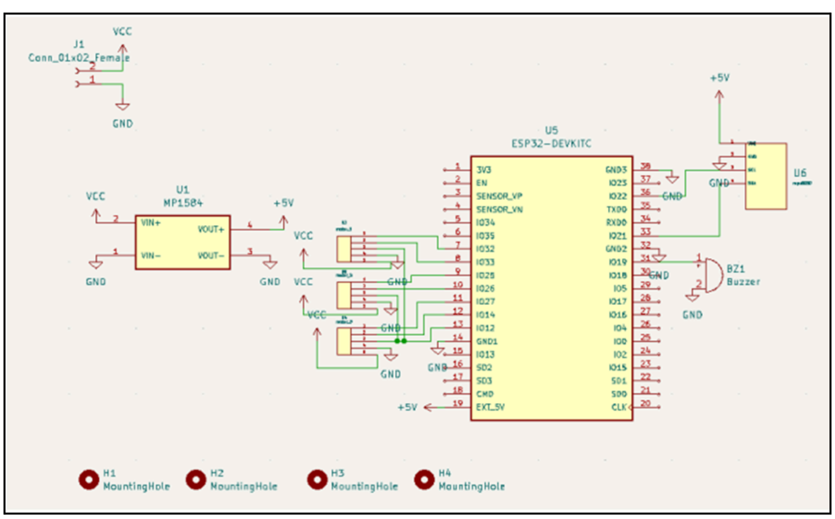


Fig. 6. CUBEBAS PCB Design

2.3. Software Review

The CUBEBAS software had small deviations from the code Rem-RC provided for its balancing cube to account for the differences in setup, but they fundamentally worked in the same way. Within the three software files, there was configuration settings for the ESP32 microcontroller, a file with different functions, and a main file that called everything in the desired order and frequency. Due to the limited computing power of the microcontroller, the code is quite simple, and comes down to a central if statement to deploy the controller. When the cube is started, it determines which balancing point it is on in the following if statement [2]:

```
if (balancing_point == 1) {
  angleX -= offsets.X1;
  angleY -= offsets.Y1;
  if (abs(angleX) > 8 || abs(angleY) > 8) vertical = false;
} else if (balancing_point == 2) {
  angleX -= offsets.X2;
  angleY -= offsets.Y2;
  if (abs(angleY) > 5) vertical = false;
} else if (balancing_point == 3) {
  angleX -= offsets.X3;
  angleY -= offsets.Y3;
  if (abs(angleY) > 5) vertical = false;
} else if (balancing_point == 4) {
  angleX -= offsets.X4;
  angleY -= offsets.Y4;
  if (abs(angleX) > 5) vertical = false;
}
```

Fig. 7. CUBEBAS Balancing Point Determination

Once it has determined the balancing point, one of the 4, it sets the control goals to the balancing point (the offsets) and maintains that position. The position is maintained by setting the desired speed of the motors using pulse width modulation to keep a desired average power. This approach is simple and very efficient, allowing for high frequency control. With more computing power, however, the software can be edited to encompass a larger operating envelope, allowing for balancing on all 12 edges and 8 corners, and even a toppling behavior for locomotion.

In terms of maintaining stability, CUBEBAS utilized a PID controller to drive the reaction wheels. While a common solution for stabilizing systems, PID controllers require manual tuning in order to achieve results with minimal error. This manual tuning also alters the reactiveness of some systems. By adjusting the gains of the PID controller, a system may react sharply with some overshoot error, while different gains could lead to a less responsive system that does not overshoot. In the following section, the framework for tackling ToppleBot's proposed goals will be discussed further.

3. Project Management

3.1. Mission Requirements

For completion within the next 8 months, the ToppleBot team has determined minimum requirements and comprehensive requirements for success. The following table details the requirements briefly.

MINIMUM REQUIREMENTS	
{1}	Construct two unique momentum-wheel cubes with enhanced motors and computing, different noticeably from the Big Dude and Little Dude designed by the CUBEBAS design team.
{2}	Design a workflow for momentum-wheel cubes based on cube dimensions, allowing for future teams to design cubes of arbitrary size.
{3}	Develop a graphical user interface to visually display cube orientation and sensor data.
{4}	Tune and implement a controller to perform controlled rotation of the cube while balancing.
COMPREHENSIVE REQUIREMENTS	
{5}	Tune and implement controllers to: - Initiate a “kip-up” with overshoot to rotate the cube. - Initiate a “kip-up” without overshoot for edge balancing. - Transition from edge to corner balancing. - Maintain corner balancing if the plane that the cube is being balanced on is raised to an incline.
{6}	Develop a Locomotion controller with odometry tracking.

Fig. 8. ToppleBot Mission Requirements

3.2. Projected Timeline

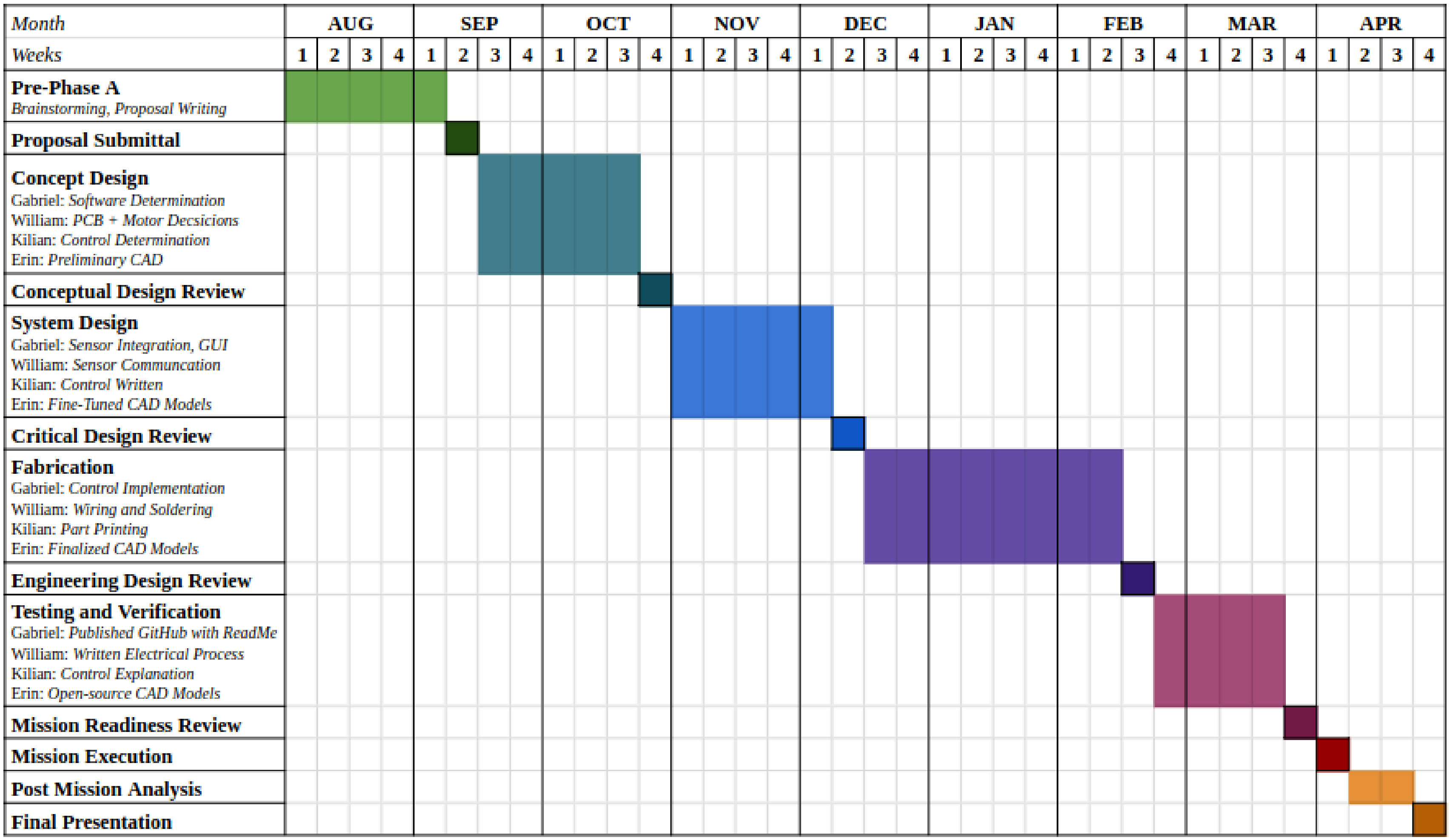


Fig. 9. Key milestones in the ToppleBot development timeline.

4. ToppleBot Technical Breakdown

4.1. Structure and Modeling

The structure of the CUBEBAS project will have to undergo several alterations in order to fulfill our project goals and requirements. Some alterations to our inherited “Little Dude” would consist of a size increase, frame alterations, frame stabilization, as well as design improvements to physical components. Overall, ToppleBot’s design takes into account the shortcomings faced by project CUBEBAS and develops an improved model.

As seen in the previous iterations of the CUBEBAS project, “Big Dude” had undergone a significant size increase in comparison to its predecessor “Little Dude”, due to a struggle to fit the components flush to the siding. This size increase will allow for more room to mount additional components, such as controllers and braking systems.

The original frame was not only quite unstable but also lacked refinement. Our goal is to stabilize the frame and fulfill some manufacturing issues within the previous project, such as:

- Unstable frame assembly, where the sides were adhered using tape.
- Wheel loosening, where the wheel would not stay on the axis.
- Off-center center of mass
- Printing tolerance issues, where the tolerances between 3D printers caused structural inconsistencies.

To improve these issues, we plan to adhere the sides of the frame together using some sort of interlocking form that will not alter the edge of the frame for balancing. This means that we can do a male/female method or some method of precise fastening; changing the material of the frame could also increase fastening options. This will hopefully stabilize the frame for more intense movement. The wheel will also have to be fastened to the axle on the motor to prevent loosening. This will be done using some form of nut and bolt. To center the mass of the entire project, counterweights can be applied to the opposing edges of the motors, and the electrical components would remain in the center of the structure. Finally, to avoid printing inconsistencies, we will utilize the same printer throughout the entire project. This will hopefully mitigate the chance of differing tolerances.

Several aspects of the original CUBEBAS design were quite preliminary, like the wheel and counterweight design. To improve these vital components, the wheel would be remodeled to be sturdier and better suited to fit the frame and motion requirements. The counterweights on the wheel would also be designed to provide a more accurate dispersion of weight and better fulfill the desired inertia.

4.2. Electrical Systems

For all the circuit and design to be done properly many parts need to work together to function as intended. These parts include the batteries, motors, motor drivers, sensors, and microcontrollers. All these parts need to be combined and wired together to not break once the cube starts to tumble and move. Additionally, it must be well connected since the circuitry must be able to function in any orientation and not hang out of the cube.

The approach to the electrical circuitry and power systems is in progress now and being developed. The electrical circuit that we have currently completed is finding all of the parts that will be used for assembly. We are going to use an ESP 32 to be the brains of the robot. This will allow us to use micro ROS (Robot Operating System). With a board that can run micro ROS the power consumption will be more rigorous. Furthermore, there will be sensors and motor drivers that will need to be controlled that will have to play into the circuitry. There will also be a wireless communication system so that the ToppleBot can communicate back to a control system and the control system can communicate to the ToppleBot to control its movement.

For the theoretical circuit design, there will be many parts that have to be integrated to work properly. The power supply must be able to supply enough voltage and current to each of the parts to ensure proper function. Since the cube is going to need to be tightly compact the power source must be a battery. With the power supply being a battery a voltage regulator will be needed to keep the voltage at a constant level to protect the circuit. Moreover, the motors will need to be chosen carefully since it will be crucial to keep the power requirement low but still have enough power. A brushless motor is needed since it produces less heat and is much more power efficient to extend the life of the battery.

The list provided shows the current parts planned to create the first ToppleBot. The parts below are mostly found from Digikey and many different options are available once more specific requirements for each part are found. The motors, motor drivers, and battery are the most susceptible to change since they are based on the model, weight, and size of the Topplebot which are not yet finalized.

Component	Possible Part	Cost
Single Board Computer	ESP 32	
IMU	Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055	$34.95
Motor Driver	IM523L6AXKMA1	$10.98
Motor	FIT0441	$19.90
Power Source (Battery)	PS-1250 F1v	$19.31

Fig. 9. Potential Electrical Component Breakdown

4.3. Software Design

4.3.1. Software Selection

In order to allow for an increase in capabilities, the Topple Bot will feature an increase in computing complexity from the CUBEBAS, and thus, the software framework must also be improved. For this reason, the software stack for the Topple Bot will be built off of ROS2 (Robot Operating System 2). This open-sourced software will allow the team to have a comprehensive home for all of the systems aboard the Topple Bot. ROS2 is widely used across different robotics applications due to its available tools and libraries that allow for development across various domains and applications. Written with Python and C++, It adds a layer of abstraction above the sensor, motor, and control integration, allowing for all of these systems to "talk" to each other at high frequencies for real-time deployment [13]. In the following section, the methods for tackling the requirements will be detailed.

With the extensive tools and libraries that ROS2 provides, there is also increased complexity, especially when compared to the Rem-RC workflow. Thus, in order to keep everything organized and the ability of version control, a GitHub repository will be made. In this repository will be the ROS workspace, where the different packages (groups of nodes) and nodes are organized. With the use of the remote repository, the team can edit the workspace off of the raspberry pi and then update the code within the pi to minimize the time it has to be on.

As mentioned previously, ROS2 has various tools and libraries that are useful for visualizing the robot. To have a display of the robot that updates as its position changes, RViz, the 3D visualization tool within ROS2, will be utilized [14]. In order to use this tool, a URDF file that represents the physics and appearance of the robot is necessary, as well as sensor data integration that is converted to transfer frames from its initial pose, known as the odometry (odom) frame. Thus, a URDF file will need to be created and nodes that interpret sensor data from the IMU and create transfer frames will need to be written. To facilitate a wireless display, the team will use Foxglove, which allows the ROS2 topics to be transferred through Wifi, and RViz to be displayed on an offboard computer [15].

4.3.2. Control Implementation

To control the ToppleBot, the team will be utilizing ROS2 action nodes. ROS2 provides the ability to create "action" nodes that can be called by other "manager" nodes, and send updates on their current status [16]. Thus, ROS2 actions will be created for both of the kip-up behaviors, corner balancing, edge balancing, and eventually roll-over for locomotion. With this framework, the team can call and keep track of robot actions dynamically and easily through the manager nodes.

Prior to implementing a control algorithm, the governing system equations will be identified, and a workflow will be implemented to parametrize each variable component of the system. While a PID controller will be utilized for initial testing, alternative control algorithms will also be explored, such as the LQR controller used by CUBLI. Should everything proceed without issue, there may even be an opportunity to train a neural network for navigation in an unknown environment.

5. References

[1] D. Harrington, D. Myers, and C. Trimble, “Project CUBEBAS: Cube Balancing and Stabilization,” May 02, 2023.

[2] Remigijus, remrc/Self-Balancing-Cube. (Sep. 08, 2024). C++. Accessed: Sep. 09, 2024. [Online]. Available: https://github.com/remrc/Self-Balancing-Cube

[3] M. Gajamohan, M. Merz, I. Thommen, and R. D’Andrea, “The Cubli: A cube that can jump up and balance,” in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 2012, pp. 3722–3727. doi: 10.1109/IROS.2012.6385896.

[4] C. Henry, “An Introduction to the Design of the Cassini Spacecraft,” Space Science Reviews, vol. 104, pp. 129–153, Jul. 2002, doi: 10.1023/A:1023696808894.

[5] Gainor, C., “Not yet imagined: a study of Hubble Space Telescope operations.” 2020.

[6] T. L. Brown and J. P. Schmiedeler, “Energetic effects of reaction wheel actuation on underactuated biped robot walking,” in 2014 IEEE International Conference on Robotics and Automation (ICRA), May 2014, pp. 2576–2581. doi: 10.1109/ICRA.2014.6907228.

[7] C.-Y. Lee, S. Yang, B. Bokser, and Z. Manchester, “Enhanced Balance for Legged Robots Using Reaction Wheels,” in 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom: IEEE, May 2023, pp. 9980–9987. Doi: 10.1109/ICRA48891.2023.10160833.

[8] R. G. Reid, L. Roveda, I. A. D. Nesnas, and M. Pavone, “Contact Dynamics of Internally-Actuated Platforms for the Exploration of Small Solar System Bodies”.

[9] NASA Jet Propulsion Laboratory, How to Explore the Surface of a Comet or Asteroid, (Sep. 03, 2015). Accessed: Sep. 09, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=bDmoqjNQAu8

[10] J. W. Romanishin, K. Gilpin, S. Claici, and D. Rus, “3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions,” in 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA: IEEE, May 2015, pp. 1925–1932. doi: 10.1109/ICRA.2015.7139450.

[11] F. Perez, "Swiss robots: Cubli, a cube that can jump up, balance, and walk across your desk," *Robohub*, Nov. 21, 2013. [Online]. Available: https://robohub.org/swiss-robots-cubli-a-cube-that-can-jump-up-balance-and-walk-across-your-desk/.

[12] A. Gajamohan, M. Merz, I. Thommen, and R. D'Andrea, "The Cubli: A cube that can jump up and balance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vilamoura, Portugal, Oct. 2012, pp. 3722-3727. [Online]. Available: https://ethz.ch/content/dam/ethz/special-interest/mavt/dynamic-systems-n-control/idsc-dam/Research_DAndrea/Cubli/Cubli_IROS2012.pdf.

[13] Open Source Robotics Foundation, "ROS2: Robot Operating System," [Online]. Available: https://www.ros.org.

[14] Open Source Robotics Foundation, "RViz: 3D Visualization Tool for ROS," [Online]. Available: https://www.ros.org.

[15] Foxglove Inc., "Foxglove Studio: ROS2 Visualization through Wi-Fi," [Online]. Available: https://foxglove.dev.

[16] R. Connolly, "Dynamic Control of Robots using ROS2 Actions," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1234-1241, 2021.

6. Letters of Intent

Gabriel Rodriguez
rodrig43@my.erau.edu
Embry-Riddle Aeronautical University
Daytona Beach, Florida, 32114

September 9, 2024

Dear Engineering Physics Review Committee,

I acknowledge that I am a co-investigator to the ToppleBot project under the Engineering Physics Senior Design course at Embry-Riddle Aeronautical University. The project aims to build on last year's CUBEBAS initiative to design and create scalable reaction wheel cube rovers. I intend to perform in accordance with course expectations, and to fully meet my responsibilities with diligence and professionalism.

I will design and implement the vehicle's software subsystem, ensuring that all subsystems are integrated and represented in a user-friendly manner. To build a fully functional system, I will work closely with the other co-investigators. More specifically, I will be coding the derived control algorithms, implementing the CAD models into the visualization environment, and enabling the onboard instrumentation to communicate effectively and efficiently.

I acknowledge that I will be responsible for the creation and integration of the software suite for the ToppleBot, in accordance with the project requirements. Given the successful completion of the project, extensive documentation will be provided, such that future teams can continue the development of the ToppleBot.

Sincerely,
Gabriel Rodriguez
Gabrial Rodriguez
Physical Sciences Department
Embry-Riddle Aeronautical University

Kilian Olen
Olenk@my.erau.edu
Embry-Riddle Aeronautical University
Daytona Beach, Florida, 32114

September 9, 2024

Dear Engineering Physics Review Committee,

I acknowledge my role as a co-investigator for the ToppleBot project as part of the Engineering Physics Senior Design course at Embry-Riddle Aeronautical University. This initiative builds upon last year's CUBEBAS project, focusing on the design and creation of scalable reaction wheel cube rovers. I am committed to fulfilling my responsibilities with diligence and professionalism, in alignment with the course expectations.

My primary responsibilities will include developing the system representation for the ToppleBot and designing the control algorithm necessary to achieve the project objectives. Additionally, I will be involved in physical manufacturing and 3D printing as required. I will collaborate closely with my co-investigators, ensuring that the development of control systems aligns with Gabriel's responsibilities while coordinating my manufacturing efforts with Erin's physical designs.

I understand that I will be accountable for both the control systems and manufacturing procedures for the ToppleBot. Upon successful completion of the project, I will ensure thorough documentation is provided to facilitate future teams in further exploring this topic.

Sincerely,
Kilian Olen
Kilian Olen
Physical Sciences Department
Embry-Riddle Aeronautical University

William Baker
Bakerw7@my.erau.edu
Embry-Riddle Aeronautical University
Daytona Beach, Florida, 32114

September 9, 2024

Dear Engineering Physics Review Committee,

I acknowledge that I am a co-investigator in the ToppleBot project under the Engineering Physics Senior Design course at Embry-Riddle Aeronautical University. This project strives to improve the CUBEBAS that was built last year by a different group. I intend to complete all course expectations and complete everything that I am tasked to do for this project. I will do my best to complete all tasks in a correct and timely manner.

My job is to design and create the electrical systems in ToppleBot. I must ensure that the ToppleBot has the proper power, components, and sensors and that all the parts are wired together correctly and securely. I will design and make a PCB for all parts to be properly attached and organized.

I acknowledge that I will be responsible for creating and building the electrical system for the ToppleBot. I will be doing my best to complete all tasks and requirements. With the hopeful success of the ToppleBot future teams will be able to continue due to all the resources acquired throughout the process being available to whomever needs.

Sincerely,
William Baker
William Baker

Physical Sciences Department
Embry-Riddle Aeronautical University

Erin Bader
Badere@my.erau.edu
Embry-Riddle Aeronautical University
Daytona Beach, Florida, 32114

September 9, 2024

Dear Engineering Physics Review Committee,

I acknowledge that I am a co-investigator to the ToppleBot project under the Engineering Physics Senior Design course at Embry-Riddle Aeronautical University. The project aims to build on last year's CUBEBAS initiative to design and create scalable reaction wheel cube rovers. I intend to perform in accordance with course expectations, and to fully meet my responsibilities with diligence and professionalism.

I will be designing and modeling the structural components of the ToppleBot. I will work with the other co-inventors to create a fully functioning system. To do this I will create CAD models that take in account stability, function, and cost. I will be in charge of altering the models throughout trials to reach our team's project requirements.

I acknowledge that I will be responsible for designing and constructing the structure of the ToppleBot to meet project requirements. Given the successful completion of the project, extensive documentation will be provided, such that future teams can continue the development of the ToppleBot.

Sincerely,
Erin Bader
Erin Bader
Physical Sciences Department
Embry-Riddle Aeronautical University