Cleanbot 3000

Joseph Barchanowicz, Daniel Vu, Mher Gharakhanian, Karl Aron Garcia, Jesus Bolaoen, Ivan Marron, Ori Dabach, Mario Valdez, Javier Reyes, Ramisa Aanan, Meyer Millman, Jose Flores, Jeffrey Broquist, Alejandro Cruz, Daniel Lopez, Rojelio Bejar, Cristian Ticas, Miguel Lucero, Adrian Gunatilake, Marcus Rentie
Advisor: Professor James Flynn
California State University Northridge
November 16, 2021

path to get maximum coverage of the UVC LEDs on the cleanroom's floor. The Marker Localization team uses fixed markers to find the Cleanbot's position inside the cleanroom. These markers allow the Clean

Abstract - The purpose of this paper is to present the research, current progress, and future plans of the Cleanbot 3000 Autonomous Sanitation Robot worked on by the Fall 2021 Senior Design Team. This project is under the advisement of Professor James Flynn. The goal of this project is to research, design, and assemble an autonomous robot that can traverse and sterilize the floors of cleanrooms at the NASA Jet Propulsion Laboratory (JPL). The robot must also adhere to JPL's specifications and cleanroom standards. The Cleanbot research and design team is divided into four different subteams which consists of Mapping Navigation, Marker Localization, Power & Mechanical Design, and Grant Research. Each subteam built upon work done by previous semesters and continued to expand research, perform computer simulations and live experiments to test algorithms, create software based models, and organize future goals.

1. Introduction

The Cleanbot 3000 team is made up of four subgroups: Mapping Navigation, Marker Localization, Power & Mechanical Design, and Grant Research. Each group has made progress by continuing the research that was done by the groups in prior semesters. Utilizing separate subgroups helps create a more streamline process in developing a prototype by separating each of the main goals and design portions of the project. The Mapping and Navigation team's job is to develop a path planning algorithm by setting a two-dimensional goal in Self Localization and Mapping (SLAM), which is used for autonomous navigation. The Mapping and Navigation team researches the best bot to know where the flight hardware and restricted areas are located. The Power and Mechanical Design team's job is to design the Cleanbot's power system as well as it's chassis design and mechanical components like wheels, motors, and cooling fans. The final group - which was added this semester - is the Grant Research team. Because of the Covid-19 pandemic, JPL has struggled to fund the Cleanbot's research and development efforts. The Grant Research team's job is to find any available grants that can provide the Cleanbot project with funding that can ensure the project keeps moving forward.

2. Mapping Navigation

2.1 Previous Work

In previous semesters, the Mapping and Navigation team manually moved a Sunfounder Pi-Car with a RPLIDAR A1 mounted on top by using simultaneous localization and mapping (SLAM) algorithm commands through Linux. Researching the navigation stack for the Robot Operating System (ROS), it revealed that since the Pi-Car was rear wheel drive and cannot be transformed into holonomic drive it was no longer useful as the Navigation team's developmental tool to design the robot.

During the Spring 2021 semester, the Mapping and Navigation team researched alternative holonomic robots to use as the developmental tool for the path planning algorithm, ultimately deciding on the TurtleBot3 (model: Burger). The TurtleBot3 is a ROS-based, programmable, holonomic robot that uses SLAM to create and map a room. The main components of the TurtleBot3 are Raspberry Pi, 360° Lidar, OpenCR 1.0, and twin Dynamixel motors [3]. Shown below in **Figure 2.1** is the TurtleBot3 Burger.

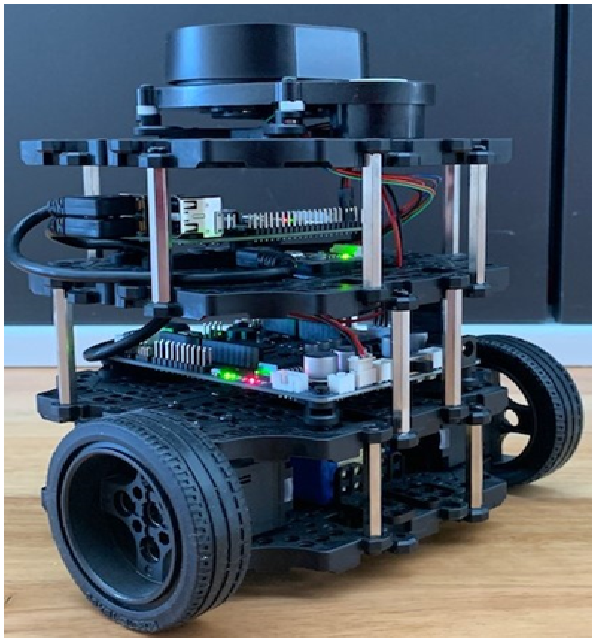

Figure 2.1: TurtleBot3 (model: Burger) assembled

2.2 Plan of Action

2.2.1 Path Planning Algorithm

The Mapping and Navigation team's goal is to implement a path planning algorithm that optimizes the total resources needed to traverse JPL clean rooms floor. One of the parameters set by JPL is that the Cleanbot must remain at least one meter from flight hardware. The plan to complete the task while remaining within the parameters is to break up the map into different segments, using the grid and linear position algorithm, to maximize the coverage by determining and prioritizing the best cleaning locations as shown in **Figure 2.2**. This also allows for multiple Cleanbot's to enter a clean room to clean a specific segment of the cleanroom floor.

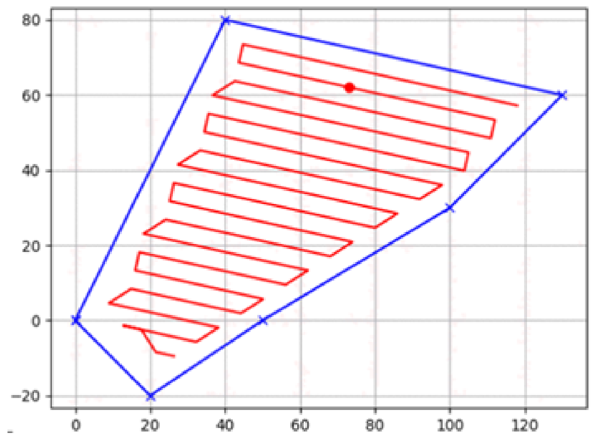

Figure 2.2: Linear Grid Path Planning Algorithm

2.2.2 Implementing Path Planning Algorithm Using RVIZ and Gazebo Simulations

The plan of action for the Mapping and Navigation team for the Fall 2021 semester was to work with ROS Visualization (RVIZ) and Gazebo simulator to run SLAM and navigation goals through a simulated world as well as the TurtleBot3. The Mapping and Navigation team continued to borrow a TurtleBot3 for the Fall 2021 semester. RVIZ is a 3D imaging software which will allow the user to get a 3D visualization of the state of the robot both simulated and real. The user can set a 2D Nav goal to set a destination for a robot to travel in either a simulated or real world. Gazebo is an open-source 3D robotics simulator which uses messages from ROS for the user to set 2D navigation goals to move a simulated robot through a simulated world. The Mapping and Navigation team used this software to simulate the navigation algorithm in a simulated JPL clean room.

2.3 Unix Setup

This semester the Mapping and Navigation team's new members gained a brief overview of using a Linux environment in order to use the software required for the Cleanbot. The team went through how to set up a ROS environment using the Ubuntu 20.04 LTS Linux distribution on their personal devices. The team installed ROS Noetic Ninjemys, Gazebo, and Python 3 ROS dependency packages in tandem with ROS visualisation (RVIZ) to run and test Turtlebot3 simulations. The team was able to use their ROS workspace to manipulate and run different launch files in Gazebo to have different maps for the Turtlebot3 to navigate in. Additionally, the team went over how to set up a Raspberry Pi 3 Model B+ to interface with the turtlebot directly.

2.4 ROS Gazebo/Simulator

2.4.1 Creating Worlds

This semester, the Mapping and Navigation team developed a map on the new ROS Noetic framework by using a Gazebo simulator. A Gazebo simulator is a built-in ROS simulated package that allows the developer to simulate real case scenarios. During this semester, it was used to simulate the TurtleBot3 to test for different specifications and scenarios set by JPL.

To create a simulated map, the user first uses a building editor within Gazebo to build a simulated room. The building editor allows the user to build the structure of the room by adding walls and multiple layers of the room, including stairs. Once the structure of the room is built, the map can be saved and exported into the Gazebo simulator. From there, the user can choose from a variety of objects which can be used to mimic different obstacles and flight hardware that is located in a JPL cleanroom.

2.4.2 Running SLAM

After a world was simulated in Gazebo, the Mapping and Navigation team placed a TurtleBot3 into the simulated world. The TurtleBot3 uses RVIZ to localize itself relative to the world by using odometry parameters to navigate the world. Using RVIZ, the Mapping and Navigation team ran SLAM simulations in which the simulation used its sensor data to construct and update a map of its environment in real time. Multiple tests were made where the TurtleBot3 autonomously traversed the floor of the simulated environment and constructed a 2D map of the space using SLAM. The results can be found in **Figure 2.3**. The SLAM data that is collected can be used for a future path planning algorithm. Using the data, the TurtleBot3 can also identify any obstacles it may encounter, such as flight hardware, as it traverses the JPL cleanrooms.

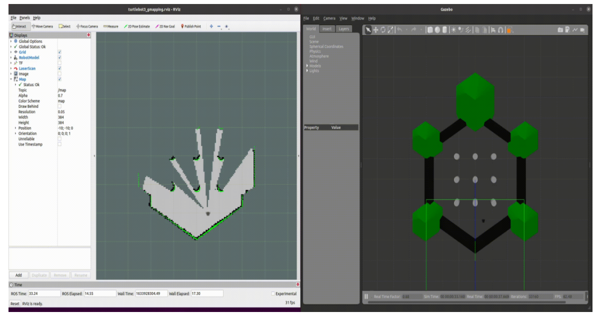

Figure 2.3: 2D mapping RVIZ (left), Gazebo simulated world (right)

2.4.3 2D Navigation Goal

One of the benefits of using RVIZ is the ability to give the TurtleBot3 a 2D navigation goal. The Mapping and Navigation team used a 2D navigation goal to see how the robot would behave in different scenarios. The 2D navigation goal uses the data collected from SLAM to localize the robot and give it a coordinate on the map to travel to. Once the robot has its destination it would choose the most efficient path to travel. This is shown in **Figure 2.4.**

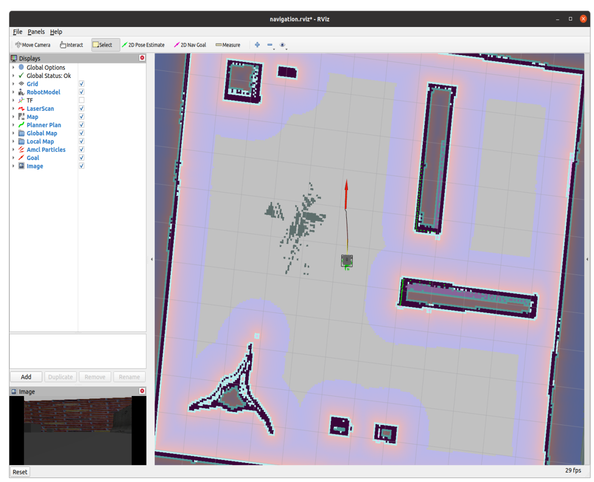

Figure 2.4: Setting 2D Navigation Goal in RVIZ

2.4.4 Inflation Radius Parameters

As part of the path planning the Cleanbot must maintain a fixed distance from all flight hardware in cleanrooms as specified by JPL. To achieve this goal, the team had to alter path costmap parameters. The costmap calculates obstruction area, potential collision area, and a robot transportable area. The costmap is a value that is between ‘0’ and ‘255’, which is used to identify whether a robot is either colliding with an object or is movable [4]. The calculation of a given area depends on the costmap parameters. Some of the configuration parameters include the inflation radius and the cost scaling factor. The inflation radius is the radius in meters to which the map inflates obstacle cost values. **Figure 2.5** shows the inflation radius value was decreased to 0.5 which resulted in a smaller distance between the wall and the TurtleBot3, as shown by the red in **Figure 2.6.** This was not optimal because the team wanted to maintain a larger distance from any flight hardware. The cost scaling factor is a scaling factor to apply to cost values during inflation. To avoid flight hardware, we increased the inflation radius parameter to 1.0 which allowed the turtlebot to take a more efficient path away from any flight hardware or walls, this can be seen in **Figure 2.7**.

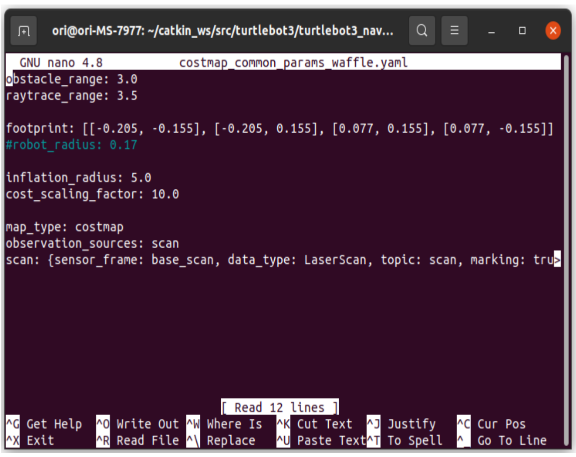

Figure 2.5: Setting Inflation Radius in costmap file

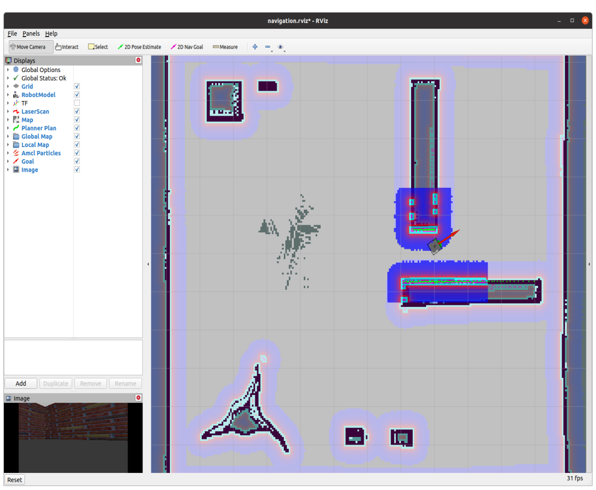

Figure 2.6: Setting Inflation radius to 0.5

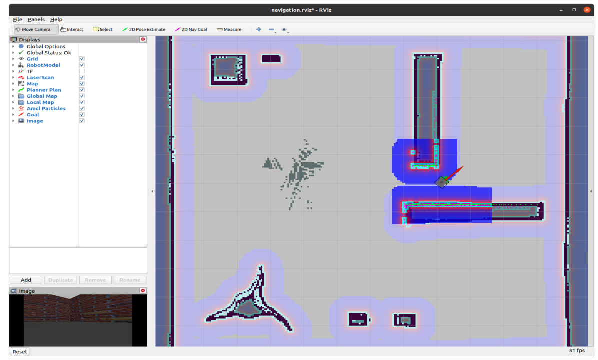

Figure 2.7: Setting Inflation radius to 1.0

2.5 ROS Gazebo/Simulator

The following sensors were implemented in order to have the Cleanbot operate more effectively: the PiicoDev Motion Sensor MPU-6050, the PiicoDev Precision Temperature Sensor TMP117, and the PiicoDev Distance Sensor VL53L1X. The motion sensor detects linear and angular motion, which is critical in order to program a kill switch if the Cleanbot tips and exposes the UVC LEDs. The temperature sensor is used to measure the internal operating temperature of the battery compartment, while also implementing an emergency shut off if the Cleanbot overheats due to internal or external temperature changes. The distance sensor is a backup sensor to the lidar and allows the Cleanbot to effectively measure its distance from objects, walls, doors, corners, and more. In order for the Cleanbot to effectively and safely navigate the real world, the motion, distance, and temperature sensors will be used. These three sensors are inexpensive and are easily implemented using the Python library and connecting them to the Raspberry Pi. **Figure 2.8** shows the current setup and testing of each sensor, using the Raspberry Pi Pico microcontroller.

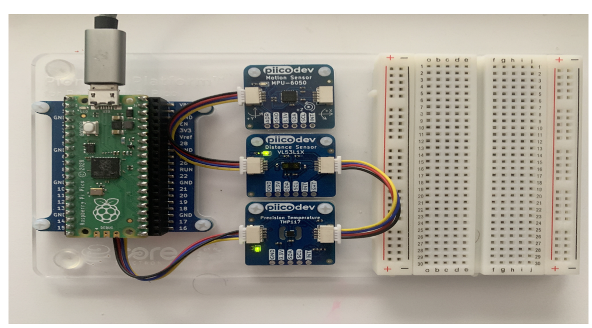

Figure 2.8: Testing the sensors using Raspberry Pi Pico Microcontroller

2.6 TurtleBot3 Burger

The Mapping and Navigation team updated the ROS distribution from ROS Melodic Morenia to ROS Noetic Ninjemys. The purpose of this update was to address concerns regarding further use of ROS Melodic. ROS Melodic has an end-of-life (EOL) date of May 2023, supports Python 2, and supports Ubuntu 18.04 [4]. Python 2 reached the end of its support in January 2020 and Ubuntu 18.04 will reach its EOL in April 2023 [5]. It is for these reasons that the distribution was updated to Noetic as ROS Noetic supports Python 3, has an EOL of May 2025, and supports Ubuntu 20.04 [6]. The Navigation team also decided to implement the update now rather than delaying it as the project's current timeline allows for troubleshooting to occur without impeding progress if any errors were to occur when updating ROS.

The Mapping and Navigation team ran into some issues while upgrading the software to ROS Noetic Ninjemys on the TurtleBot3. The first issue was the TurtleBot3 LiPo battery would not hold a charge and a new battery had to be ordered. The second issue was that when running RVIZ, a wheel link error was received, shown in **Figure 2.9.** Initially, the Mapping and Navigation team thought that this error was due to the TurtleBot3 running on the shore power connection because of the bad LiPo battery. After receiving the battery the wheel link error still persisted. The Navigation team is currently troubleshooting this error with Tribotix, a robotics shop in Lambton, Australia.

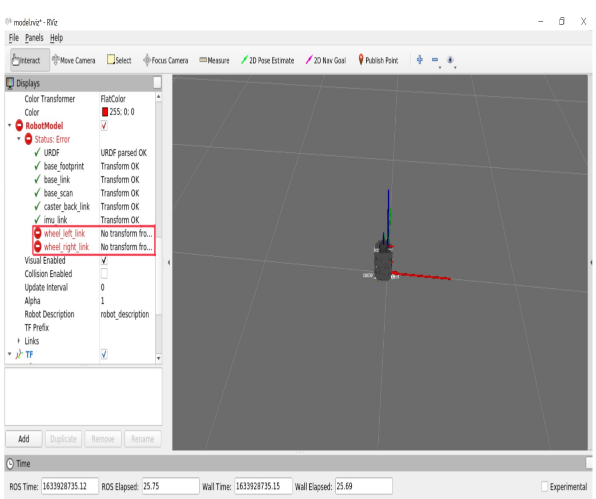

Figure 2.9: Wheel Link Error in RVIZ

3. Marker Localization

3.1 Previous Work

During the previous semesters, the Marker Localization team conducted research using fiducial markers known as ArUco markers. These markers are placed in the viewpoint of an imaging system and appear in the system's generated image, where they are used as a reference point and to gauge distance. ArUco markers are binary squares with a black background and a distinct, generated white design that is easily detectable by an imaging system. They can be created in any size and each one has a unique identification. They are currently being used to mark where flight hardware is located in JPL's cleanrooms.

The hardware that was used for the imaging system during previous semesters included the Raspberry Pi 3B+ and the Raspberry Pi V2 camera. The goal for the Spring 2021 semester was to obtain data from a marker at any location in a room. To test this, an ArUco marker was placed at a fixed location and the imaging system was moved to various points around the room with respect to the marker. The markers were then tested at three different sizes: 3.5 x 3.5 cm, 14 x 14 cm, and 16 x 16 cm. The baseline size for the marker was set at 14 x 14 cm and was able to be read by the imaging system at 4.62 meters. By increasing the size of the marker to 16 x 16 cm, the recognition distance was increased to 7.01 meters and inversely, reducing size of the markers to 3.5 x 3.5 cm, the maximum recognition distance was reduced to 1.37 meters.

3.2 Plan of Action

The plan of action for the Marker Localization team during the Fall 2021 semester was to achieve greater recognition distances in detecting ArUco markers as well as conduct testing in low-light environments. This semester, the Marker Localization team transitioned to the Raspberry PI High Quality (HQ) camera with the 6mm Arducam lens from the Raspberry PI Camera Module V2. The Camera Module V2 offers an 8 megapixel still resolution while the HQ camera offers a 12.3 megapixel still resolution [7]. Another factor that led to upgrading the camera was the small size of the light-sensitive elements in the Camera Module V2 that resulted in poor sensitivity and a low signal-to-noise ratio. Additionally, the

Camera Module V2 is not suitable for mobile applications due to its fixed focal length. The HQ camera can be used with any standard C or CS-mount lens (6mm and 16 mm) which can take higher-resolution images and is better for use in mobile applications than the Module 2. The only drawback of the HQ camera is that since it takes a higher resolution image, it consequently uses more memory than the Module 2 camera.

3.3 HQ Camera Calibration

To begin the HQ camera calibration process, the camera calibrations drivers need to be installed by using the Linux command:

```
rosdep install camera_calibration
```
[8].

After installing the driver, the following commands are used to calibrate the camera:

```
roslaunch raspicam_node
camerav2_1280x960_10fps.launch
enable_raw:=true
```
[8].

```
rosrun camera_calibration
cameracalibrator.py --size 8x6 -square
0.074 image:=/raspicam_node/image
camera:=/raspicam_node
```
[8].

Running these commands opens the calibration window, where a large 8x6 checkerboard with 74mm squares is used to calibrate the camera, as shown in **Figure 3.1**. The program has three measurements, X, Y, and size and they each have a colored meter to indicate how much information they have calibrated, showing red when still lacking information and green when a sufficient level of information has been calibrated.

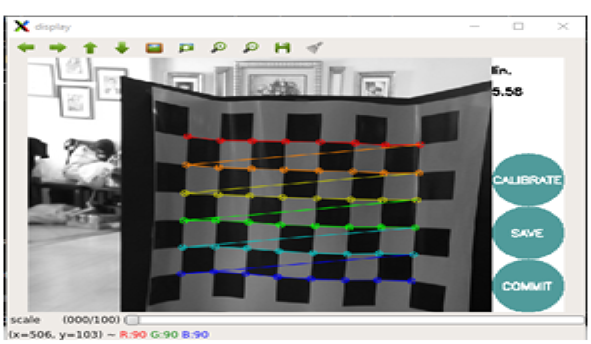

Figure 3.1: Calibration process with checkerboard

3.4 ArUco Marker Test Process

Once the HQ camera is calibrated, it is ready to be used for fiducial marker detection. The Markers team generated ArUco markers and set them up at various distances, heights and angles in a room. It takes four separate Linux terminals to run the fiducial detection software. The first terminal command is:

```
roslaunch raspicam_node
camerav2_1280x960.launch
```
[8].

This command confirms that the camera was calibrated and is accessible to the Raspberry Pi. The second terminal command is:

```
roslaunch fiducial_slam
fiducial_slam.launch
camera:=raspicam_node
```
[8].

This command displays the number of markers detected as well as the number of markers in memory as shown in **Figure 3.2**. The third command is:

```
roslaunch aruco_detect
aruco_detect.launch
camera:=raspicam_node
```
[8].

This terminal creates a map based on the robot's position, as shown in **Figure 3.3**. The fourth terminal command is:

```
roslaunch fiducial_slam
fiducial_rviz.launch
```
[8].

The combination of the last three session commands allowed the markers to be detected by the camera and opened the RVIZ software to visualize the markers.

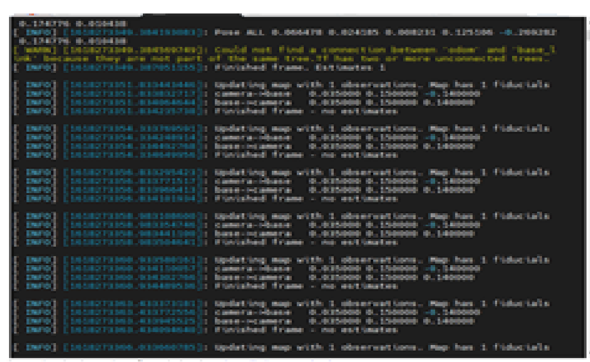

Figure 3.2: The Fiducial Detect Terminal.

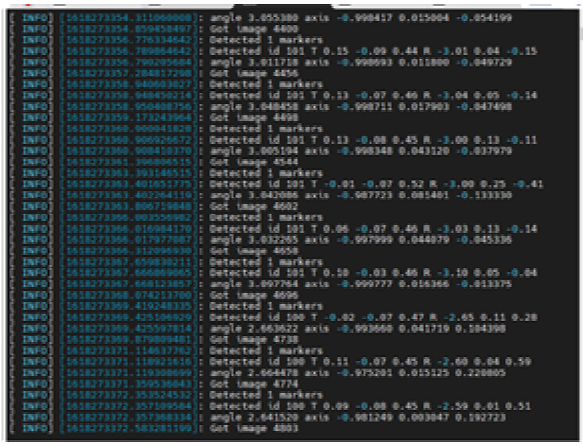

Figure 3.3: ArUco detect Terminal

3.4.1: Testing the Camera in Room

After all four terminals were initialized, the Markers team tested the marker detection of the HQ camera. The markers were set up in various locations in rooms with different lighting conditions to see if the HQ camera could detect them. **Figure 3.4** and **Figure 3.5** show the markers set up in a dark room and well-lit room, respectively, and both figures show that the markers were properly detected. When a marker is detected by the camera, its position is translated to the RVIZ software and placed in a grid. With the camera detecting the markers, different conditions can be set up to test things like the sharpest angle and furthest distance at which the markers can be detected.

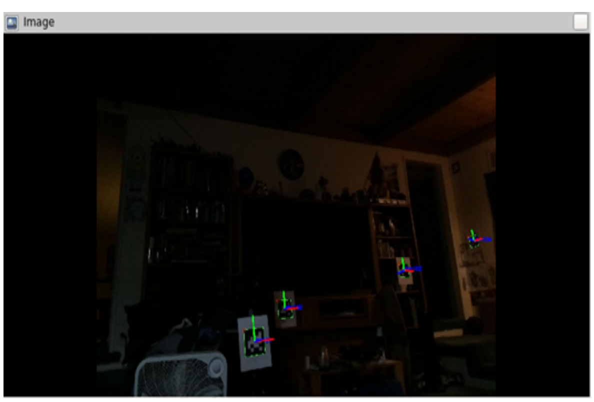

Figure 3.4: Marker test in dark room

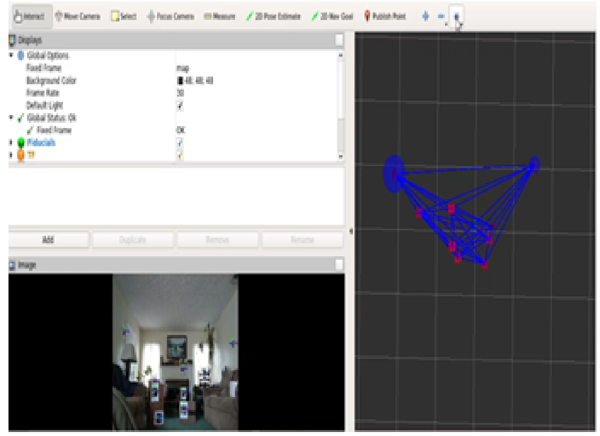

Figure 3.5: Marker test in well-lit room with RVIZ software visible

3.4.2: Testing in Larger Area

Next, the team tested the furthest distance that the HQ camera could detect the ArUco markers. Two different sizes of markers were used. The larger markers were 11 x 11 inches and standard size markers were 9.5 x 9.5 inches. As confirmed, the HQ camera could successfully detect all markers at 30ft, as shown in **Figure 3.6,** but could not detect all the smaller markers at 40ft, as shown in **Figure 3.7**. At 60ft, the HQ camera could only detect the larger markers and not detect any of the smaller markers, as shown in **Figure 3.8**. **Figure 3.9** shows the markers and mapped in RVIZ.

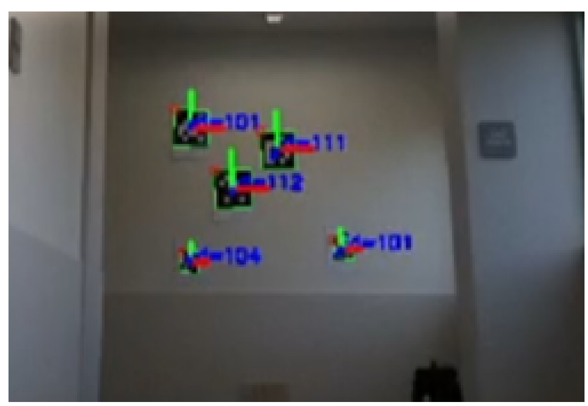

Figure 3.6: Markers detected at 30ft

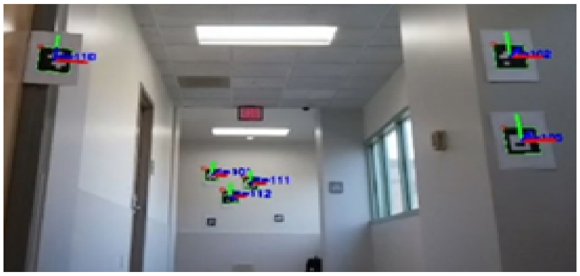

Figure 3.7: Markers detected at 40ft

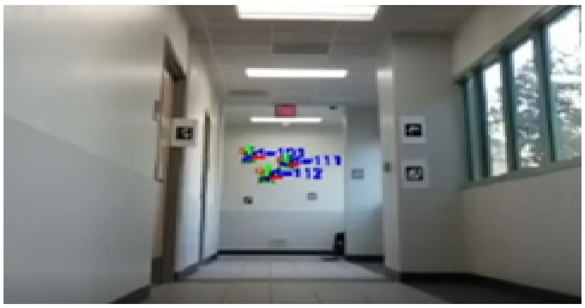

Figure 3.8: Markers detected at 60ft

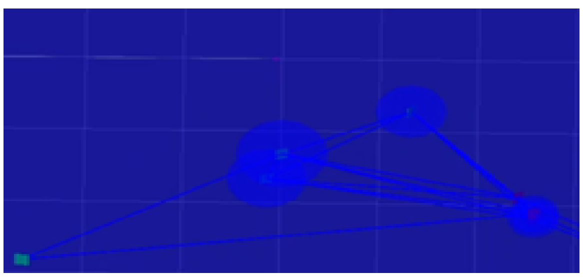

Figure 3.9: RVIZ mapping

The Marker Localization team also tested the camera with two different angles. The first test position was at 19ft away with a 27 degree angle, as shown in **Figure 3.10**. The second test position was at 15ft away with a 23 degree angle, as shown in **Figure 3.11**. In both conditions the markers were successfully detected.

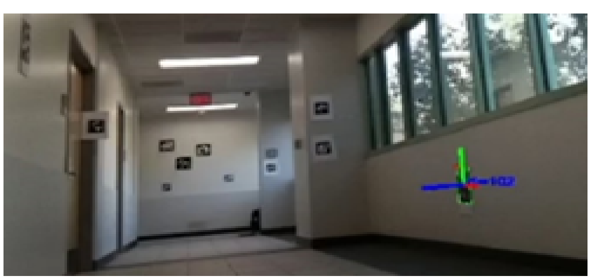

Figure 3.10: Markers detected at 19ft with 27 degree angle

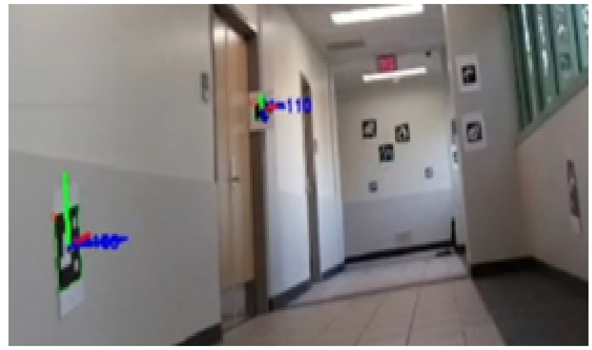

Figure 3.11: Markers detected at 15ft with 23 degree angle

4. Power & Mechanical Design

4.1 Previous Work

In previous semesters, the Power and Mechanical Design team created a power budget, a Simulink model of the Cleanbot's power system, and a potential four-wheel, two-motor, eight-battery design in addition to continuing research on a docking station for charging the Cleanbot.

However, during the Spring 2021 semester, JPL gave the team the unfortunate news about budget cuts due to the Covid-19 pandemic, so efforts were made by the Power and Mechanical team to create a design and choose parts for a two-wheel drive design for the Cleanbot based on the TurtleBot3 Burger model, instead of the ideal four-wheel drive design, to make creating a prototype more financially feasible. In order to visualize this new design, some members of the team began creating a digital model of the Cleanbot using the Solidworks CAD program and successfully modeled a motor mount. The battery team members sized-down from an 8-battery design to a cost-efficient 2-battery design and reached out to Inspired Energy to get a quote on the cost of their NH3054HD34 batteries. The motor team members utilized the RobotShop Drive Motor Sizing Tool [9] to produce a list of several potential brushless DC motors with enough torque to travel over the cable cover ramps placed throughout the JPL cleanroom floors. Per the recommendation of the JPL team, the chassis would be constructed of 80/20 anodized aluminum and the wheels would be made of polyurethane.

4.2 Plan of Action

For Fall 2021, the Power and Mechanical Design team planned to continue research on the essential mechanical components (chassis, wheels, and motors) and the power components (batteries and docking station). Each of the Power and Mechanical sub-task teams would be researching or creating both ideal and low budget options to accommodate the possible lack of funding, and to have designs ready for when the team receives adequate funding in the future. Additionally, the team came into the semester still missing several important pieces of test data from the JPL team, therefore final decisions on components could not yet be made.

For the chassis, team members would get acclimated with the Solidworks CAD software and produce more detailed digital models of the Cleanbot components and chassis. Due to the complexity of designing a part on Solidworks, the team decided to continue simulating the four-wheel design of the Cleanbot. As for the wheels, the team planned on compiling a list of potential polyurethane wheel options of various radii, with an emphasis on cost efficiency. For the motors, the team planned to provide multiple options to drive the Cleanbot. One of those options would be a brushless DC motor with an integrated speed controller and the other would be a BLDC motor with a separately wired speed controller. Lastly, for batteries, the team planned to continue building on potential options for the docking station and embedded charger, as well as finding more cost-effective battery options to provide enough power to the Cleanbot for a reduced cost.

4.3 Power System Design

4.3.1 Batteries & Charging Station

Building upon previous semester's research, the battery team continued the search for a compatible docking station for the TurtleBot3. Unfortunately, after reaching out to Robotis, the manufacturers of the TurteBot3, the team received word that there are no existing docking stations for the TurtleBot3 platforms at the moment, but there is hope that one will be developed in the next year or so. Due to budget constraints, the battery team decided to continue using the battery from Inspired Energy, NH3054HD34 [10], as well as the embedded smart charger, EB330A [11], while continuing to compile a list of other batteries and charging options that are more affordable and provide the necessary energy requirements. The battery team looks forward to exploring more cost-effective battery options that can get the Cleanbot one step closer to producing a prototype.

4.4 Mechanical Design

4.4.1 Solidworks

Solidworks is a solid modeling computer-aided design and computer-aided engineering computer program [12]. The main goal of using this program is to create a detailed visualization of the Cleanbot's four-wheel design. This semester, the Solidworks sub-task team members were able to design and model three components: a motor mount (**Figure 4.1**), a fan (**Figure 4.2**), and a chassis (**Figure 4.3**).

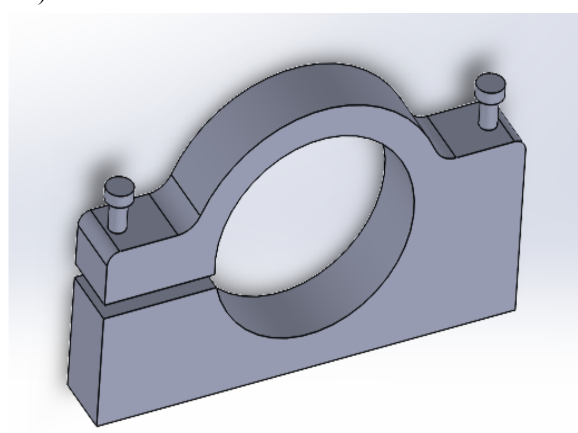

Figure 4.1: Updated motor mount modeled in Solidworks

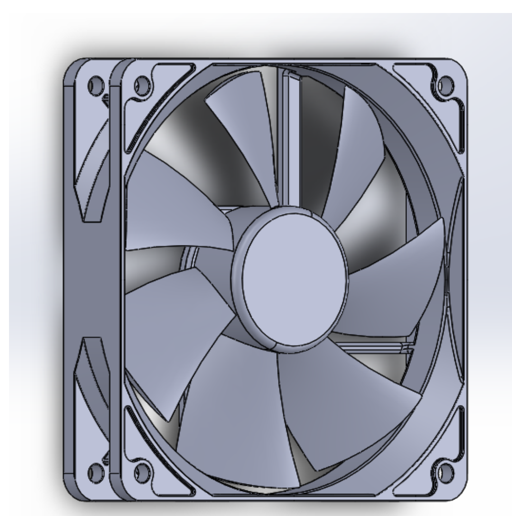

Figure 4.2: Cooling fan modeled in Solidworks

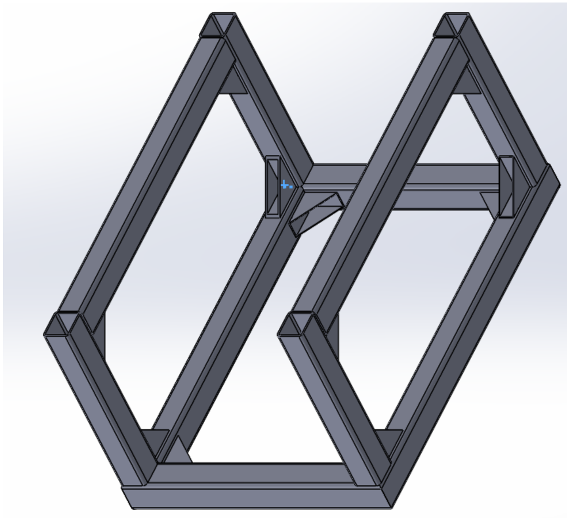

Figure 4.3: Preliminary four-wheel chassis modeled in Solidworks

The motor mount was a component that was modeled last semester, but it was recreated this semester with a few improvements to the design. The new members of the Solidworks team also recreated this component on their own as a way to practice using Solidworks. After learning more about the Solidworks program, the team as a whole began working on other components such as the fan and the chassis mentioned above.

Last semester, the Solidworks team researched thermal modeling to take into consideration the heat dissipation from the Cleanbot during its operation. Although thermal modeling research was not conducted this semester, the team decided to create a component of a fan this semester. The Solidworks subteam plans to incorporate one to three of these fans into the Cleanbot's final four-wheel chassis design. These fans will help cool down the Cleanbot components that dissipate the most heat (i.e. batteries, light engines, motors, microcontroller) and prevent them from reaching temperatures that could harm themselves or the components surrounding them. After modeling the fan, the team moved onto modeling a preliminary design for the four-wheel chassis (**Figure 4.3**). After completing the preliminary model, the team moved onto designing a more detailed version of the four-wheel chassis. After several weeks, the team was able to complete the detailed four-wheel chassis design, seen below in **Figure 4.4**

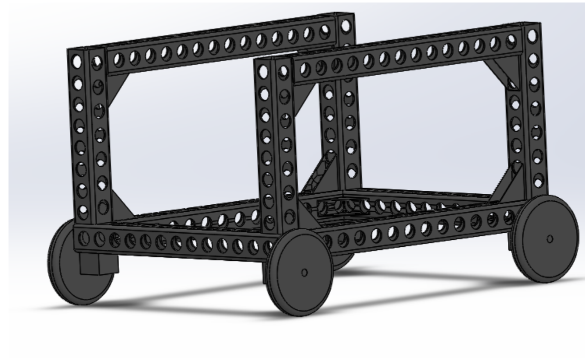

Figure 4.4: Four-wheel chassis with stand-in wheels and sheet metal material modeled in Solidworks

Although **Figure 4.4** is more detailed than **Figure 4.3**, the team could not include the motor mounts and the fans that have previously been modeled because a majority of the Cleanbot components have not yet been decided upon. Moving forward, the Solidworks team plans to use this chassis design as a base model to build upon as more and more of the Cleanbot's components are finalized and a prototype will soon be visualized.

4.4.2 Motors

The responsibility of the motors subtask group is to ensure the continuous movement of the Cleanbot throughout the JPL cleanrooms. The biggest obstacles that the Cleanbot will encounter in the cleanroom aside from flight hardware and other objects to be avoided are the cable covers spread throughout the cleanroom floors. Since the previous semester's team decided that the Cleanbot would be driven using two motors and four wheels, this semester's team members continued the search for the most reliable and cost effective options with enough torque capability to push the robot over the cable ramps. Taking into account the torque calculations performed from last semester's motors team, this semester's team searched for motors that could make at least 290 ounces of force per inch of torque in order to make sure that the 25 lb Cleanbot can make it over the cable covers. Another motor characteristic the team wanted to narrow down was whether or not the motor would have an integrated speed controller or if the speed controller would be wired separately. Speed controllers allow the user to program the RPMs of the robot. The Cleanbot's speed has yet to be determined as the JPL engineers experiment with the speeds necessary to kill the desired percentage of bacteria with each pass. The first motor/speed controller configuration that has been considered is using an externally wired speed controller that is compatible with both the motor and the Raspberry Pi 3 that our team is currently using. A speed controller “HAT” for DC and stepper motors from Adafruit [13] can be seen below in **Figure 4.5**.

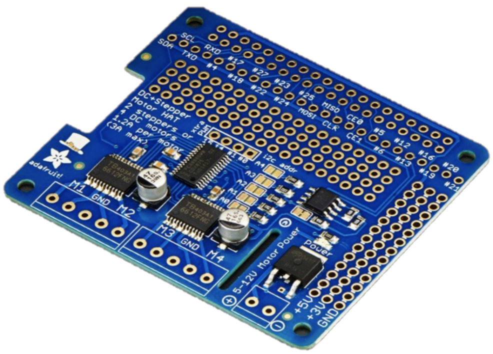

Figure 4.5: DC Motor Speed Controller “HAT” for Raspberry Pi [13]

The benefit of an external speed controller is that it provides the robot with more customizability and flexibility in terms of component placement on the chassis. Additionally, the speed controller can be changed to maintain compatibility with the Cleanbot’s microcontroller in the case the microcontroller is changed in the future. The alternative design to an externally-wired speed controller is an integrated motor. Integrated BLDC motors have an electronic speed controller already included within the motor’s housing, thereby decreasing the amount of components that need to be wired and saving space on the Cleanbot’s chassis. Several options for integrated motors were found on Anaheim Automation [14] and an example can be seen below in **Figure 4.6**.

Figure 4.6: Brushless Motor with Integrated Speed Controller [14]

4.4.3 Wheels

Due to the Cleanbot’s greatly decreased budget, and given the fact that the team had access to the TurtleBot3 for research purposes, the members of the wheel design team decided to use the TurtleBot3 as a reference and followed its two-wheels/caster wheel design. The team continued with its decision to use polyurethane for the wheels in order to reduce markings on the floor and minimize the amount of bacteria caught on and transported by the wheels. A potential option for the polyurethane wheels found during the previous semester can be seen below in **Figure 4.7**, but this semester’s wheel team compiled a list of more cost-effective polyurethane wheels that can be accessed by future members to choose from.

Figure 4.7: Polyurethane Wheel [15]

An issue arose for the wheel team due to the fact that the polyurethane caster wheels found last semester included exposed wheel bearings in their design. The JPL team determined that this could be highly problematic due to the grease inside the wheel bearings, which would break down and outgas over time, counteracting the Cleanbot’s effort to clean the cleanroom’s floor. After conducting research to find possible alternative caster wheels, the team narrowed down the acceptable caster wheel options to ball caster wheels made of plastic or metal. The Wheels team’s most up-to-date documentation of its polyurethane wheel findings includes a breakdown of a ball caster [16] as shown in **Figure 4.8**. In upcoming semesters, the Wheels team will finalize the Cleanbot’s exact wheel dimensions and continue searching for suitable options, whether it be a product already in existence or a custom order sent to a manufacturer by the Wheels team.

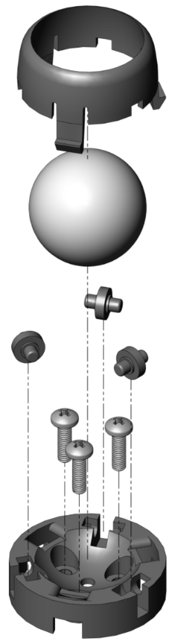

Figure 4.8: Breakdown of a Ball Caster [16]

5. Grant Research

Due to the previously mentioned budget cuts from JPL, the Cleanbot leadership was encouraged by Professor Flynn to start a research subteam focused solely on applying to grants that could help fund the Cleanbot's continued research and development. The new Grant Research team specifically looked for opportunities on Grants.gov from agencies sharing the same interests as our project: autonomous robot intelligence.

The team focused their efforts on The Foundational Research in Robotics Grant program from the National Science Foundation (NSF), which supports research on robotic systems. According to the NSF, a robot is defined as "intelligence embodied in an engineered construct, with the ability to process information, sense, and move within or alter its working environment" [17]. Additionally, the NSF states that "...intelligence includes a broad class of methods that enable a robot to solve problems or make contextually appropriate decisions." [17] For this grant, the grants.gov submission website requires six forms which must be completed and uploaded with additional supporting documents such as a cover page. The grant research subteam divided these forms among its three members evenly. Since the forms require many specific details about the project lead and budgeting information, it's critical for this research team to work closely with Professor Flynn throughout the application process. Although the grant application has no set deadline, the Grant Research team plans on completing its submission as soon as possible in order to continue the research and development of the JPL Cleanbot without its current financial limitations.

6. Conclusion

As the Fall 2021 semester comes to an end, the Cleanbot team continues to make progress even with budget cuts at JPL and the COVID-19 pandemic still affecting in-person group meetings. The research that has been completed this semester has allowed for good progress on the Cleanbot to be made.

The Navigation team resumed its research on path planning algorithms from the previous semesters. The Robot Operating System (ROS) for both the TurtleBot3 and Gazebo was upgraded this semester from Melodic to Noetic because Melodic was reaching its end of life. Simulations of the TurtleBot3 were completed in Gazebo where different inflation radius and cost map were tested for an optimal coverage map. The future goal of the Navigation team is to upgrade the hardware to the Raspberry Pi 4 Model B. Currently, the Navigation team is in possession of a Raspberry Pi 3 Model B+, but its product lifetime ends January, 2023 [18]. Upgrading the hardware would provide a stronger CPU in the form of a Broadcom BCM2711SoC and a variety of RAM options, ranging from 2GB, 4GB, and 8GB [19]. A Raspberry Pi 4 Model B would also improve the performance when running real-time simulations and would help "future-proof" our progress. Whether the team can procure a Raspberry Pi 4 Model B or not is dependent on budget restrictions.

The Marker Localization team resumed its research on making fiducial markers using the ArUco library to approximate the location of the Cleanbot. This semester, the Marker Localization Team upgraded its camera from the Raspberry Pi Camera Module V2 to the Raspberry Pi HQ camera with the 6mm Arducam lens. The HQ camera was able to successfully detect 11 x 11 inch markers at a distance of 60 feet. While the HQ camera is better used for mobile applications and has a higher pixel count than the Module 2, it uses more memory and increases latency. The Marker Localization team plans to borrow a Zybo Z7-10 FPGA development board from CSUN next semester to see if this will help with the latency issues.

The Power and Mechanical team continued its research on the Cleanbot's power system and mechanical components, as well as the development of a detailed digital model on Solidworks. The battery/charging station team reached out to manufacturers to get a better understanding of the kinds of docking stations available that are compatible with the TurtleBot3 that the Cleanbot team has been using for research. The team plans to continue finding cost-effective options that are compatible with the Cleanbot's current NH3054HD34 battery and EB330A embedded smart charger from Inspired Energy [10], [11]. The motor research team compiled a list composed of two types of brushless DC motor options with acceptable torque outputs. The first option being a motor with an integrated speed controller and the second option being a BLDC motor with an externally connected speed controller that would interface with the Cleanbot's microcontroller. The motor team will continue comparing the two configurations in order to determine the optimal design for the Cleanbot. The wheel research team updated the polyurethane wheel documentation to include more cost-effective options and hopes to finalize the exact dimensions of the wheels in order to purchase a wheel that is already in production or to place a custom order with a manufacturer. Lastly, the Solidworks/chassis design team successfully modeled preliminary designs of the Cleanbot's chassis, motor mounts, and cooling fans on Solidworks and will continue to develop more detailed designs and models as the Cleanbot's components are finalized in the coming semesters.

The Grant Research team was created this semester to assist the team in securing future funding for the Cleanbot project. This semester, research was completed on the initial process of signing up for grants using the Grants.gov website. The primary focus is completing the paperwork for the Foundational Research in Robotics Grant program from the National Science Foundation (NSF). The goal is to be able to submit all the paperwork for this grant, with hopes of receiving funding during the Spring 2022 semester.

References

[1] D. J. Smith, "A Balloon-Based Payload for Exposing Microorganisms in the Stratosphere (E-MIST)," *Gravitational and Space Research*, vol. 2, no. 2, Dec., pp. 70-80, 2014.

[2] "NIH Human Microbiome Project defines normal bacterial makeup of the body," National Institutes of Health, 31-Aug-2015. [Online]. Available: https://www.nih.gov/news-events/news-releases/nih-human-microbiome-project-defines-normal-bacterial-makeup-body.

[3] "ROBOTIS e," Manual. [Online]. Available: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/.

[4] "Wiki," ros.org. [Online]. Available: https://wiki.ros.org/Distributions.

[5] "Sunsetting Python 2," Python.org. [Online]. Available: https://www.python.org/doc/sunset-python-2/.

[6] VarHowto Editor Welcome to VarHowto! Profile Posts, V. H. Editor, Welcome to VarHowto! Profile Posts, Profile, Posts, Kansaisays: VarHowto Editorsays: August 9, V. H. Editorsays: P. Author, and N. *, "ROS Noetic: What You Need to Know," VarHowto, 15-Jul-2020. [Online]. Available: https://varhowto.com/ros-noetic/#Noetic_vs_ROS_Melodic_What's_New.

[7] IkikV. C. 2 years ago, IkikVC, IkikV. C. S. author's posts, and S. author's posts, "Raspberry Pi Camera: Comparison of High Quality Camera with Camera Module V2," Latest Open Tech From Seeed, 29-May-2020. [Online]. Available: https://www.seeedstudio.com/blog/2020/05/14/raspberry-pi-camera-comparison-of-high-quality-camera-with-camera-module-v2/#:~:text=Camera Module V2 VS HQ Camera , 12.3 Megapixels 8 more rows .

[8] "Wiki," ros.org. [Online]. Available: http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration.

[9] Robotshopmascot, "Drive Motor Sizing Tool," Robotshop, 03-July-2013. [Online]. Available: https://www.robotshop.com/community/blog/show/drive-motor-sizing-tool.

[10] "NH2054HD34 Battery," Inspired Energy. [Online]. Available: http://inspired-energy.com/products/n-series-batteries/details/3/80-nh2054hd34.

[11] Dhiren, "Inspired Energy Charger EB330A," Accutronics. [Online]. Available:

http://www.accutronics.co.uk/inspired_energy_chargers/eb330a.html.
[12] "SolidWorks," Wikipedia, 25-Sep-2021. [Online]. Available: https://en.wikipedia.org/wiki/SolidWorks.
[13] A. Industries, "Adafruit DC and Stepper Motor HAT for Raspberry Pi - Mini Kit," adafruit industries blog RSS. [Online]. Available: https://www.adafruit.com/product/2348.
[14] BLY34MDC - Brushless Motors with Integrated Speed Controllers. [Online]. Available: https://www.anaheimautomation.com/products/brushless/brushless-integrated-item.php?sID=879&pt=i&tID=97&cID=48.
[15] McMaster Carr. [Online]. Available: https://www.mcmaster.com/standard-wheels/polyurethane-wheels-7/.
[16] "Pololu Ball Caster with 1″ Plastic Ball and Plastic Rollers," Pololu Robotics & Electronics. [Online]. Available: https://www.pololu.com/product/2691#lightbox-picture0J6615.
[17] M. 5 and F. 9, "Foundational Research in Robotics (Robotics)," Beta site for NSF - National Science Foundation. [Online]. Available: https://beta.nsf.gov/funding/opportunities/foundational-research-robotics-robotics.
[18] Raspberry Pi Datasheets. [Online]. Available: https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf.
[19] Raspberry Pi, "Raspberry Pi 4 Model B specifications," Raspberry Pi. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/.