Cleanbot 3000

Harut Ketsoyan, Eros Aldin Lachica, Shayan Reza, Gary Chiv, Vicente Zavala, Lilit Aghanes, Cristian Miramontes, Samuel Ochoa, Joe Barchanowicz, Daniel Vu, Karl Garcia, Jesus Bolaoen, Ivan Marron, Ori Dabach, Mher Gharakhanian
Advisor: Professor James Flynn
California State University Northridge
April 20, 2021

Abstract - The objective of this paper is to present the research and current progress of the Cleanbot 3000 autonomous sanitation robot conducted by the senior design team during the Spring 2021 term. The team works under the advisement of Professor James Flynn to research, design, and assemble an autonomous robot tasked with traversing the cleanrooms of Jet Propulsion Laboratory (JPL) to sterilize the floors while adhering to JPL's specifications and cleanroom standards. Due to the magnitude and complexity of robotics design, the team was split into three different subteams which consist of mapping navigation, marker localization, and power. The subteams used what resources were available to build upon work done from previous semesters by creating software based designs, computer simulations and live experiments to test algorithms, broadening research, and organizing future plans.

1. Introduction

The primary focus in creating the Cleanbot 3000 is to create a robot that will autonomously navigate the Jet Propulsion Laboratory (JPL) cleanrooms with the purpose of sanitizing the rooms by dosing the floor with UVC light to kill and/or inactive microorganisms that may be present within the rooms. Any microorganisms and microbials that exist within the cleanrooms will be considered contaminants as they pose a threat to the terrestrial ecology of Earth's upper atmosphere if they are transmitted via JPL's flight hardware [1]. These organisms are capable of invading and contaminating the environment and can be problematic in posing as false positives as life sources found in outer space if transited on flight hardware. To resolve these issues, the Cleanbot will be tasked with the elimination of microorganisms/microbials while minimizing the amount of human interaction needed to perform this task, as humans serve as primary sources of contamination in the cleanrooms [2]. Although JPL laboratory workers have taken necessary precautions to try to better prevent the amount of contamination they output while working in the cleanroom, the less human interaction involved with the process of cleaning the rooms would be better to further minimize sources of contamination.

The Cleanbot team is a relatively new senior design group which was started in Spring 2019 and has made several changes to the requirements desired in order to match to the constant changes in the specifications set by the project. The scope of the Cleanbot is to design an autonomous robot that will navigate JPL cleanrooms and sanitize the surface area of the floors using ultraviolet C (UVC) LEDs. To prevent any possible damage to the flight hardware while cleaning the rooms, the robot must be programmed to remain at least one meter away from any designated and restricted zones.

The Cleanbot team comprises three different subgroups: Mapping Navigation, Marker Localization (Aruco Markers), and Power. These groups have been making continuous progress by resuming the research and/or progress made in the previous semesters in order to compile the multiple considerations made to create an optimal design for the Cleanbot. With close communication between these teams, it is possible to iterate designs of the Cleanbot that can consider the research done by the different teams. By creating sub-teams, the multiple work and design considerations needed to be done to get a Cleanbot assembled and fully operational may be done in a matter that considers the organization needed to get a working prototype together.

The mapping navigation subteam is tasked with building upon the SLAM (Self Localization and Mapping) algorithm implemented from the previous semesters to set two dimensional navigation goals for autonomous movement. The team has also worked on research for path coverage algorithms to perform the task of autonomous navigation within a room that is mapped. The marker localization subteam is tasked with localizing the robot's position within the cleanroom with use of pre-placed markers. The team is also working on setting perimeters of the cleanroom and restricted zones. The power subteam is tasked with designing a system capable of powering the entire cleanbot, as well as the physical hardware and mechanical design of the robot.

2. Mapping Navigation

2.1. Previous Work

In the previous semester, the Navigation team was able to create a map using a RPLIDAR A1 mounted on a Pi-Car by manually moving the car through an enclosed area and running SLAM algorithm commands through Linux. While a map was able to be successfully created, the Sunfounder Pi-Car S did not meet the hardware requirements needed in order to navigate with a newly saved map. After researching the Navigation stack/code for ROS (the Robot Operating System), it was determined that the Sunfounder Pi-Car S would no longer be deemed useful as a developmental tool for the design of the robot as it could not perform any of the navigation goals due to the constraint that it is a rear wheel drive vehicle and does not have the capability to be transformed into a holonomic drive robot.

2.2 Plan of Action

The plan of action for the navigation team for the semester of Spring 2021 was to incorporate the SLAM mapping algorithms with navigation goals and path planning. Since the Pi-Car robotic car no longer meets the hardware requirements for the ROS navigation package the team decided to move forward with ROS Gazebo simulator to run SLAM and navigation goals through a simulated world and turtlebot3, holonomic and differential drive robot.

2.3 ROS Gazebo/Simulator

2.3.1 Making Worlds

This semester the Navigation team decided to further develop their simulation and testing opportunities by using a new type of simulator, Gazebo simulator. Gazebo is a 3D robotics simulator, with many companies and/or organizations in the industry, including NASA, Toyota and Office of Naval Research to name a few, holding competitions where contestants would operate robots in a virtual environment.

The Gazebo simulator allows the team to create, save, and load customized worlds ranging from simple indoor room walls to complex outdoor settings [11]. The Navigation team built small worlds for different testing scenarios that helped prove and/or disprove concepts or methods that would narrow down parameters for different navigation algorithms. One of the worlds that the team were experimenting with gave visuals within the environment that would outline objects such as walls, boxes, and even trees to develop an in depth map of the world. After the creation of the worlds are completed, the navigation team still has the option of making edits to the environment to adjust the experiments accordingly.

2.3.2 Running SLAM

After creating a world in the Gazebo simulator, the team proceeded to place the turtlebot in the world and run SLAM. Simultaneous Localization and Mapping is a computational problem of building a map of an unknown environment while keeping track of the robot's location as it moves.

The Navigation team ran testing with Hector SLAM, Gmapping, and Google Cartographer during the previous semester with the PiCar. The Hector SLAM algorithm is an open source algorithm that uses laser data from a lidar to create a map of its surroundings. However the algorithm did not incorporate the robot's odometry data. Gmapping is a SLAM approach that uses particles as individual maps and particle filters to learn grid maps. The algorithm used laser range data and odometry, however Gmapping had trouble adapting to the team's current hardware. This is because the Gmapping algorithm has not been optimized for the Rplidar, and parameterized given its max and min range data readings, operating frequency, and accuracy. The team lastly tested with Google Cartographer, which uses both laser data and odometry. The advantage in using Google Cartographer is the real-time mapping and configurability with many types of sensors, including the RPLIDAR - this is shown in. The team was able to accurately create maps of living rooms and bedrooms, including small corners and legs of chairs - this is shown in **Figure 2.1**.

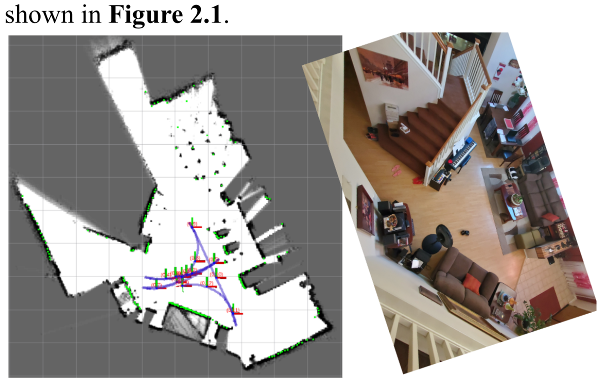


Figure 2.1: Google Cartographer

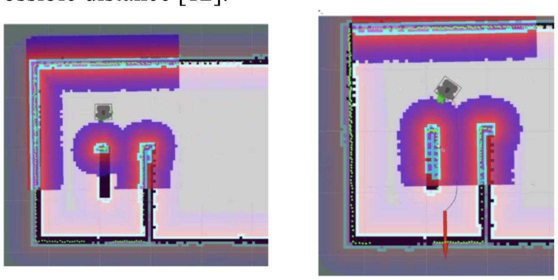
Thus following the success of the previous semester, the team uses Google Cartographer to map the created virtual world. Once the entire world is mapped, the map is saved onto the Raspberry Pi to be reopened with the navigation package for setting navigation goals.

2.3.3 2D Navigation Goals

When the Gazebo simulated world is mapped and saved, the world is reopened in Rviz using the ROS navigation package. Once it is opened in Rviz, 2D navigation goals could be set using the UI tool. A 2D navigation goal works by setting a coordinate on the map. After the navigation goal is set, the robot will choose the shortest path to reach the coordinate while avoiding obstacles.

2.3.4 Inflation Radius Parameters

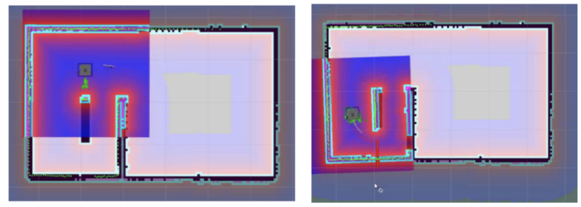
Once the 2D navigation goals were successfully achieved the navigation team worked to alter the algorithm to meet JPL's specifications. In order to keep a one meter distance from all flight hardware in the cleanrooms, the inflation radius parameter was altered. Inflation relates to the inflation of obstacles in the map (red zone) seen in **Figure 2.2**. Increasing the inflation parameter results in the red zones to grow larger causing the robot to take wide turns to get to a 2D navigation goal. Decreasing the inflation parameter causes the robot to take tighter turns to reach the navigation goal with the shortest possible distance [12].



inflation: 0.5

Figure 2.2: Test Case Low Inflation Region (inflation = 0.5)

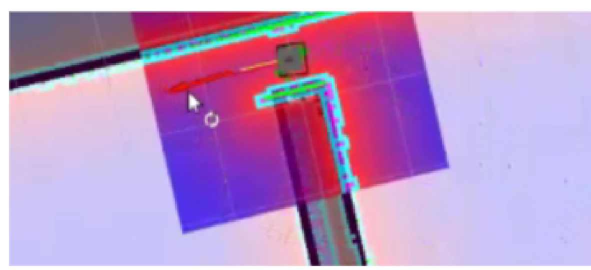
This solves the problem of keeping a set distance away from flight hardware when it is a single body in a large room. However, a test was conducted to see if the inflation region parameter adjustment can have the robot completely avoid certain areas of multiple flight hardware or obstacles that are too narrow to traverse through. A scenario world was created and mapped in Gazebo to test the robot's decision making algorithm. This scenario shows the robot in front of a wall with a 2D navigation goal set on the other side. To eliminate the factor of chance in choosing a side for the robot to traverse, the navigation goal was set slightly to the right. As shown in **Figure 2.2** when a small inflation region is chosen the robot will choose the narrow side which is the shortest distance to the navigation goal. However, in **Figure 2.3**, when the inflation region is increased to 1.0, the robot chose the longer wider path. This test is significant because it proves that by altering the inflation region, the robot can avoid paths that are confined to less than one meter.



inflation: 1.0

Figure 2.3: Test Case High Inflation Region (inflation = 1.0)

In a separate test, the team experimented further with the inflation radius to see if increasing it high enough will cause the robot to ignore a navigation goal set in a tight area. A scenario was created and tested below where a 2D navigation goal is set on the other side of a choke point. As shown in **Figure 2.4**, the robot completely ignored the red zones of the large inflation region that had been set to 3.0, which is three times more than the high inflation test case used in the previous experiment. It was concluded that the 2D navigation has precedence over other parameters and will ignore them in order to always reach its goal. This test was significant in that it proved that 2D navigation alone could not be used as means of autonomous navigation since it may ignore parameters in certain situations. It also proved the necessity of including Aruco markers in the cleanroom as a backup measure to insure complete avoidance of certain areas of the cleanroom.



inflation: 3.0

Figure 2.4: 2D Navigation Goal precedes inflation parameters

2.4. Sensors

2.4.1 RPLIDAR A1

The RPLIDAR A1 was used last semester as it was a low cost 360 degree LIDAR which is capable of measuring distances up to 12 meters with a sample rate of 2000-8000Hz and a typical scan frequency of 5.5Hz. The goal for using this lidar was to be able to create maps through ROS and use various slam algorithms to create maps that would allow the robot to have a reference map for navigation.

The navigation team is planning to acquire a lidar better than the RPLIDAR because the JPL Cleanrooms are much larger in scale to the rooms that are currently being scanned with the RPLIDAR. A better LIDAR would allow for faster scanning of maps and more detailed resolution of maps. Scanning the maps in a short span of time is important because it would lessen the amount of time needed to scan the rooms so that the power consumption can be used for cleaning the cleanrooms.

2.4.2 360 Laser Distance Sensor LDS-01 (LIDAR)

The LDS-01 is a LIDAR that is packaged with the TurtleBot3. The Lidar is capable of using a 2D laser that can sense at an angular range of 360 degrees to collect data sets to be used for SLAM (Simultaneous Mapping and Localization). The detection distance is significantly less than the RPLIDAR A1 with a detection range between 120mm ~ 3500mm (max 3.5 meter distance range) with a sampling rate of 1800Hz and a scan frequency of 5Hz.

The LDS-01 is currently being used on the TurtleBot3 and will later on be replaced with the RPLIDAR A1 that is in possession or will be replaced by a more robust LIDAR.

2.5 Path Planning Algorithms

For the path planning algorithm the Navigation team wants to implement a cleaning path that will optimize the amount of resources needed as well as cover the whole space of the JPL clean room. To do so the navigation team plans to use map segmentation in order to maximize the cleaning coverage, determine optimal cleaning locations, and it will allow us to break up the room in sections and assign sections to a swarm of robots.

Map segmentation works by dividing the map into segments called "C" which can be denoted as a pixel. For example if C = 21, then it means the size of the segment would be 21 x 21 pixels. This allows for the Cleanbot to work within segments using a linear guided path thus using less resources and a more efficient clean since it only has to manage each section at a time. This can also ensure cleaning is done properly because the Cleanbot can travel over the same segment multiple times. Such path planning algorithms would display a record of floor coverage as the example in **Figure 2.5**.

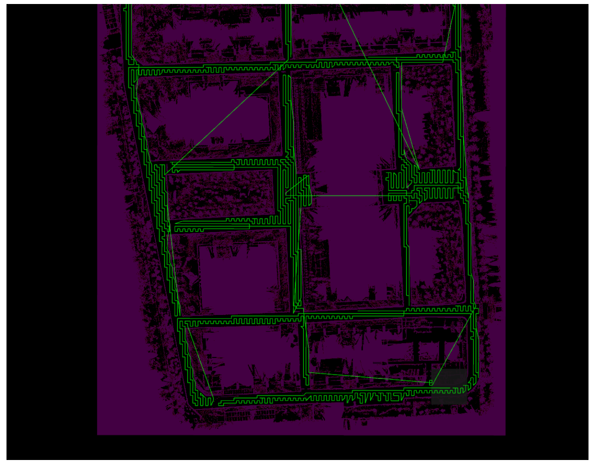


Figure 2.5: ROS Path Planning Algorithm

2.6 TurtleBot3 Burger

This semester, the navigation team purchased a TurtleBot3 burger. The TurtleBot3 burger is a holonomic, programmable, ROS-based mobile robot that the navigation team used for prototyping the path planning algorithm. The TurtleBot3 burger uses SLAM(simultaneous localization and mapping) to build and map a room. The TurtleBot 3 burger uses 360 degree Lidar for SLAM and navigation purposes. It uses a Raspberry Pi as a single board computer. The OpenCR 1.0 is used for power and sensor control, and has been developed for ROS embedded systems that provides completely open-source hardware and software. These components, which make up the Turtlebot3, make it completely customizable and useful for the prototyping needs of the navigation team path planning algorithm. Shown below in **Figure 2.6** is the assembled TurtleBot3 burger.

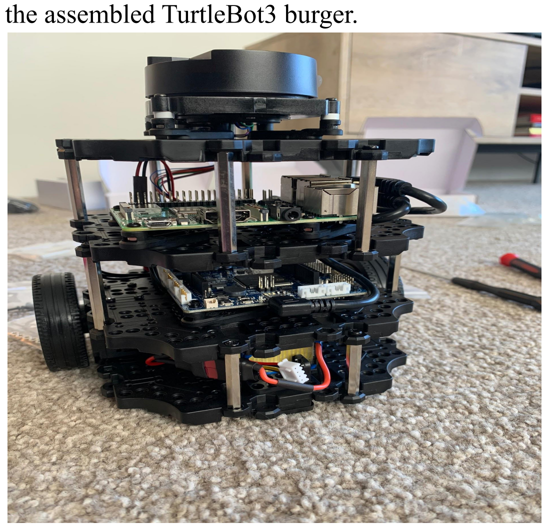


Figure 2.6: Built TurtleBot 3

3. Fiducial Marker Localization

3.1 Previous Work

Last semester, Fall 2020, the fiducial marker was introduced to the Marker Localization team. The hardware that was used are the Raspberry Pi v2 camera and Raspberry Pi 3B board. On the Pi board, the SD card was formatted and a swap file was created. A camera calibration was done for the ArUco detection algorithm to detect the fiducial markers. This was done with the help of a checkerboard of size 35”x 44” 8x6 checkerboard and moving it in various directions. **Figure 3.1** demonstrates the calibration process.

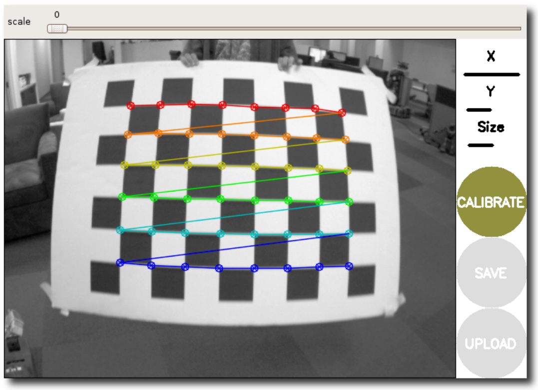


Figure 3.1: Calibration testing

3.2 Linux OS Commands

Prior to running commands on the terminal, the SD card must be formatted so the raspberry pi can use more memory on board. The linux commands are: roslaunch fiducial_slam.launch camera:= raspicam_node, this command handles the detection of the fiducial markers as shown in **Figure 3.2**, roslaunch aruco_detect aruco_detect.launch camera:= raspicam_node, this builds and estimates a map based on the robot’s position as shown in **Figure 3.3**, roslaunch fiducial_slam fiducial_rviz.launch, this is the launch command to view the map in RVIZ [3]. Once all of the commands are executed, an RVIZ fiducial GUI will appear for the user to visualize and interact with. The other 2 terminals will display a continuous update when a marker is detected.

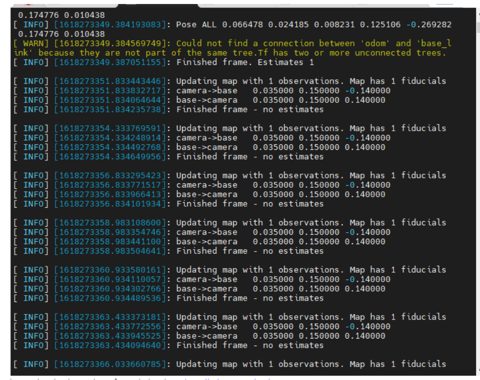


Figure 3.2: The terminal displays the number of markers being observed as well and number of markers in memory.

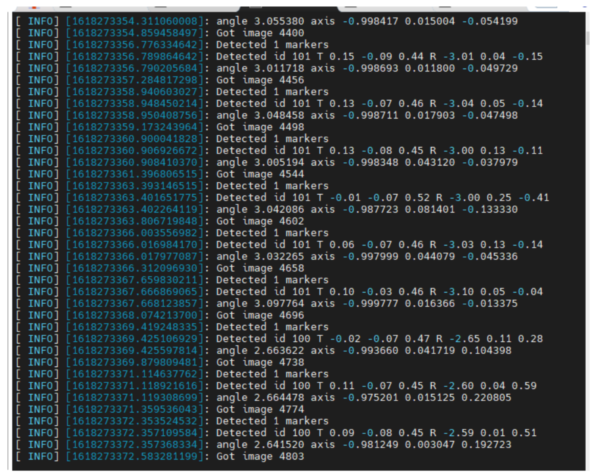


Figure 3.3: Terminal displaying the changing angels and axis of the marker, this can be visualized on the RVIZ fiducial GUI.

3.3 Testing and Results

For Spring 2021, one of the goals of the Marker Localization team was to extract data from the fiducial markers at any location in a room. A marker was placed at a fixed location and the camera was moved around, closer and further away from the marker. A relative distance between the camera and the marker was obtained but the error margin between the camera’s reading and actual reading was about 50% erroneous. However, when mapping the room with multiple markers, the reading was much more accurate. In **Figure 3.5**, the highlighted part shows a reading of 4.688m and when measuring it, shown in **Figure 3.4** the actual distance was 4.622m, a big improvement.

Figure 3.4: Tape measurement from the fiducial markers to the camera's position

When increasing the original size of the fiducial marker from 14 x 14 cm to 16 x 16 cm (a 14.29% increase), the detection distance was 7.01 m. However, the minimum reading distance was also increased to 50 cm. The next test was with smaller markers of size 3.5 x 3.5 cm, the results were a max detection of 1.37 m and a minimum distance of 11 cm. As a result of the testing, the conclusion is there are trade-offs between increasing/ decreasing the size of the marker and the detection distances.

A map has to be created with the fiducial markers before any localization can be determined. To do this, two or more markers must be captured by the camera at the same time and continue this pattern until the first marker is reached, shown in **Figure 3.6**. The result will be a mapped perimeter reflecting where the markers are at relative to the robot, shown in **Figure 3.5.** Once established the robot will be able to move freely around the room and know its location when it detects a fiducial marker. [4]

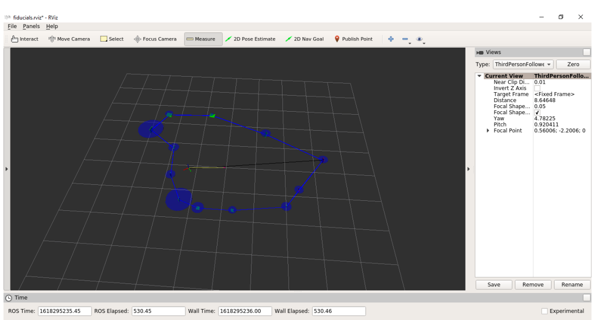


Figure 3.5: RVIZ fiducial GUI

Figure 3.6: Multiple markers placed about 5 feet apart.

4. Power

4.1 Previous Work

In the previous semester the power team chose batteries, created a Simulink model, created a power budget and continued the research on the light engines and docking station. The students from the previous semester also started researching different types of tires and motors that would be used for the Cleanbot. For the batteries, the team had agreed to go with 12 lithium ion smart batteries from Inspired Energy NH2054 since the batteries are high rated, meaning discharge and charge rate is greater than a previous battery that was chosen. The Simulink model consisted of the power distribution system of the Cleanbot. In the power budget the total power consumption was calculated. In the previous semester the power team had an idea of what type of wheels and motors to use which were 2 omnidirectional wheels and two polyurethane wheels with 12V BLDC motors but further research had to be done.

4.2 Plan of Action

The power team's plan of action for Spring 2021 was to continue working and researching on the different components for the robot which include the wheels, motors, motor mounts, and different materials for the structure of the robot. The body of the robot was a main concern for the team, as a simulation of the model would need to be built on solidworks. Creating a model of the Cleanbot on solidworks was the main goal since physical materials were unattainable. The simulation was needed to visualize the dimensions and the weight of the components. The initial plan was to create a model on Simulink and once it was completed, the team would request the parts needed from JPL. However, due to budget cuts, the parts for the physical model were not attainable. After redirecting from the previous plan of action, the power team started researching alternate ways to reduce the cost of the creation of the Cleanbot. The initial twelve batteries were reduced to two batteries and a different configuration was created for the wheels and motor in order to be more efficient with the cost of parts and materials.

4.3 Body

4.3.1 Solidworks

Solidworks is a solid modeling computer-aided design and computer-aided engineering computer program. Members of the power team used this program to start the design of the body of the Cleanbot. It is difficult to bring in Solidworks files of the components to design them in the CAD, as there are changes still being made to the layout. The purpose of using Solidworks is to get an idea of the weight of the robot to see if it fits the requirements. Thermal modeling is something that the power team took into consideration when researching Solidworks due to the heat dissipation from the Cleanbot when it is running. Thermal analysis is used to find temperature distribution, temperature gradient, heat flow, and heat exchange between the different components of the robot. After becoming familiarized with the Solidworks program, the team began to construct models of potential parts for the Cleanbot. The following figure is a motor mount created on Solidworks. This design allows for the motor to be placed in the center while the flat side is mounted on the aluminum chassis frame as shown in **Figure 4.1**.

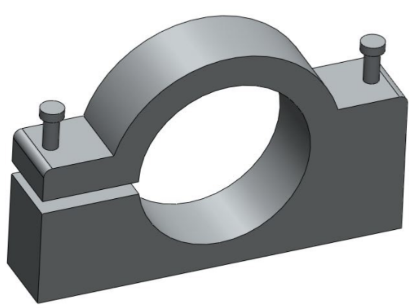


Figure 4.1: Motor Mount

4.3.2 Material

The power team also decided on the material that will be used for the chassis, with the consultation of JPL engineers. The body of the chassis will consist of aluminum (80/20 T-Slot aluminum) as shown in **Figure 4.2**. The base and the walls of the Cleanbot will be made of acrylic sheets. Anodized aluminum was chosen on the basis that it has great durability and doesn't weigh as much as other metals. Since there are requirements for the amount of weight the Cleanbot 3000 can be, this option is very suitable. Choosing acrylic sheets compared to any other alternate, the acrylic is lightweight and more flexible than glass. These two materials will create the body of the chassis.

Figure 4.2: Aluminum 80/20 T-Slot [9]

4.4 Mechanical Design

4.4.1 Motors

This semester the power team decided that the Cleanbot will consist of two BLDC motors that are capable of moving up to 10 RPM. The team was researching different motors that fell within this category but then was informed by JPL that the JPL facility has cable ramps for the wires that they use. This made the research become more complex because the motor with a required torque was needed so the Cleanbot is able to go over the cable ramps. A great resource called RobotShop Drive Motor Sizing Tool was utilized to help the power team find the right motor that was needed for the Cleanbot to execute the goal of overcoming small ramps and maneuvering throughout the cleanrooms that it was assigned to do. The BLDC motor in **Figure 4.3** is 12V and has an RPM of 4800 [7].This could be a potential motor used on the Cleanbot, once the engineers at JPL inform the team of the torque requirements.

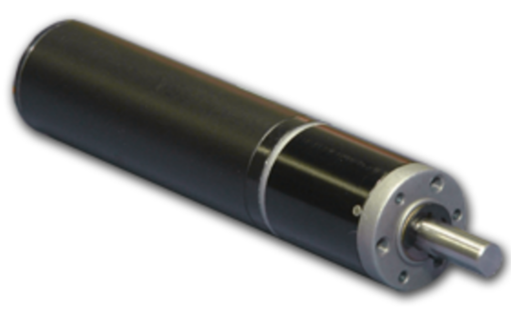


Figure 4.3: BLDC Motor [7]

4.4.2 Wheels

After extensive research, it was decided that the robot will be built with two motors and will consist of four polyurethane wheels. However, due to the budget cuts this semester, the current plan is to design a two wheel drive robot that consists of two wheels and one caster wheel to work out the balance of the robot and two motors. Once this design is built, the team can start testing the wheels and motors that were chosen. After a substantial amount of consultation with the engineers at JPL, the team decided that polyurethane was the best material for the wheels as shown in **Figure 4.4**. This was the optimal choice because of its ability to keep the dispersion of dust and bacteria at a minimum.

Figure 4.4: Polyurethane Wheel [8]

5. Battery and Charging Station

Last semester, the power team decided on the NH3054HD34 battery from Inspired Energy [5] as well as the EB330A smart embedded charger (**Figure 4.5)**. The team contacted the manufacturer and received a quote for these products. The initial Cleanbot design consisted of 12 batteries, however due to budget cuts, the new design will utilize 2 batteries. The charger mounted on the docking station will transmit signals to the sensors installed on the Cleanbot. When the Cleanbot detects its battery needs to be charged, it will use these signals to move to the docking station. Once the Cleanbot is properly docked, the batteries can begin charging. [10]

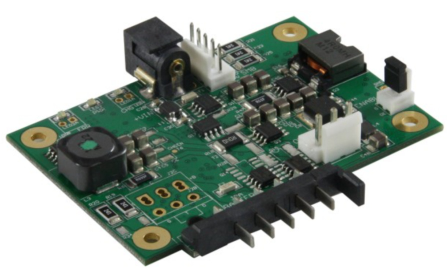


Figure 4.5: Inspired Energy EB330A Charger [6]

6. Conclusion

As the team went through another semester in this pandemic with budget cuts, there were still advancements made on this project. The continued research and collaboration during the development of this project has allowed for major improvements to be made.

The mapping navigation team is continuing with their findings of last semester in which it was found that the hardware requirements needed to be changed as a result of discovering that the robot needs to be a holonomic differential drive vehicle in order to operate for autonomous navigation. Simulations of a holonomic robot (TurtleBot3) in Gazebo were made as the current Sunfounder Pi-Car S in possession would not be possible to integrate any autonomous movement since it is not a holonomic robot. The team has recently procured a physical TurtleBot3 (model "Burger") and has assembled it and begun testing for autonomous navigation. Once the team has solved the path planning algorithm they want to use that will eventually be used to navigate the JPL Cleanrooms, the navigation team can start coming up with plans to prototype the actual Cleanbot based on the TurtleBot3 hardware design and software applications intended to drive it for autonomous navigation.

The marker localization team used the ArUco library to generate fiducial markers which can be used to estimate the pose of the Cleanbot. During the testing process, the team was able to accurately measure the distance between the Cleanbot and the markers, as well as create a map of a room. Furthermore, the testing also revealed that there are trade-offs to be considered when choosing different marker sizes. The future goals of the team is to continue testing the limitations of the markers, as well as create a more complex map with obstacles to better resemble a cleanroom.

The power team will continue to build models of the Cleanbot parts on Solidworks, and work their way up to a complete model. Once JPL approves funding for the project and updates the team on the results of the tests done for the light engines and motor torque requirements, the team can begin constructing a physical model. Their future goals include having a budget in order to create a prototype of the Cleanbot and begin testing. Furthermore, the power team must continue the research of the docking strategy for the charging station, finalize the materials and components, and create the optimal design for the Cleanbot chassis.

With the hopes of funding for the coming semesters, the Cleanbot will be able to exit its research phase after over two years of comparing models and creating proof of concepts, and enter a stage of prototyping. After polishing the Simulink hardware designs and finalizing the SLAM, autonomous path coverage, and marker localization algorithms, a first prototype model can be built with the specified materials and hardware. Further testing can be done once a physical model is created to analyze its behavior and ensure that the cleanbot meets all of the requirements. A prototype model will be a huge step forward in the project and in robotics, as a safer method of providing UVC sanitation in not only cleanrooms, but hospitals and other facilities.

References

[1] Smith, David. (2014). A Ba lloon-Based Payload for Exposing Microorganisms in the Stratosphere (E-MIST). Gravitational and Space Research. 2. 70-80.

[2] MacDougall, Raymond. “NIH Human Microbiome Project Defines Normal Bacterial Makeup of the Body.” *National Institutes of Health*, U.S. Department of Health and Human Services, 31 Aug. 2015, www.nih.gov/news-events/news-releases/nih-human-microbiome-project-defines-normal-bacterial-makeup-body.

[3] “Wiki,” *ros.org*. [Online]. Available: http://wiki.ros.org/fiducials. [Accessed: 16-Apr-2021].

[4] “Mapping and Localization from Planar Markers,” *Mapping and Localization from Planar Markers | Aplicaciones de la Visión Artificial*. [Online]. Available: http://www.uco.es/investiga/grupos/ava/node/57/. [Accessed: 16-Apr-2021].

[5] *Inspired Energy*, inspired-energy.com/products/n-series-batteries/details/3/80-nh2054hd34.

[6] Dhiren. *Accutronics*, www.accutronics.co.uk/inspired_energy_chargers/eb330a.html.

[7] “Brushless Motors with Planetary Gearboxes.” *BLWRPG09 - Brushless Motors with Planetary Gearboxes*, www.anaheimautomation.com/products/brushless/brushless-gearmotor-item.php?sID=155&pt=i&tID=98&cID=47.

[8] “Polyurethane Wheels.” *McMaster*, www.mcmaster.com/standard-wheels/polyurethane-wheels-7/.

[9] “80/20 1010-72 1’ X 1’ T-Slotted Profile, 72’ Stock Bar.” *Global Industrial*, www.globalindustrial.com/p/101072-1-x-1-tslotted-profile-72-stock-bar.

[10] M. Doumbia, X. Cheng, V. Havyarimana, “An Auto-Recharging System Design and Implementation Based on Infrared Signal for Autonomous Robots”. IEEE, 2019, [Online]. Available: https://ieeexplore-ieee-org.libproxy.csun.edu/document/8813317/authors#authors [Accessed: 16-Apr-2021]

[11] N. Lamprianidis, “Navigate Gazebo Worlds,” nlamprian, 26-Dec-2019. [Online]. Available: https://nlamprian.me/blog/software/ros/2019/12/26/navigate-gazebo-worlds/. [Accessed: 19-Apr-2021].

[12] K. Zheng, “ROS Navigation Tuning Guide,” *KaiyuZheng*, 02-Sep-2016. [Online]. Available: https://kaiyuzheng.me/documents/navguide.pdf. [Accessed: 19-Apr-2021]