# MAC: Modular Autonomous Cart, Group 8 Senior Design Project

Lisian Shehu, Trevor Taulien, Brandon Silva and Justin TenEyck

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — Modern homes are now filled with smart devices intended to automate and improve the daily life of people. Robots are a subset of smart devices that help automate daily needs, the most common being vacuums. With a growing and competitive market for robots, there is not much development for robots to help those who really need them, like elderly or disabled people. As of 2014, approximately 26% of elderly live alone and as of 2016, 1 in 4 adults in the US live alone with a disability affecting mobility, cognition, hearing, and vision, according to the CDC. A mobility disability is the most common, affecting 1 in 7 adults. This project aims to build an autonomous and robust robotic system that will improve the daily lives of this demographic by easing the transportation of home objects such as laundry, groceries, and small furniture items, with modular attachments to give the MAC different functionalities easily.**

## I. Introduction

Disabled and elderly people have always had a tough time living on their own. Through many different technologies, it has become easier for these people to live relatively normal lives without assistance from others, like in a nursing home or assisted living facility. There are devices to help people into their showers, up and down stairs, and make food preparation and other household chores easier. With this, robotics in the home has slowly but steadily been increasing in popularity, with one the most common applications being robot vacuums for automated cleaning of household floors. However, automatic carts or robots to move objects have little market for the average consumer, let alone someone with specific disabilities or needs. For such a common task of needing to lift or transport large bulky objects or many small objects, there needs to be a device that can reliably transport cargo from one part of a home to another, thus the need for a device like the MAC. According to the AARP, the average income of someone aged 65 or older is $31,742 per year, with half of the people falling below the median income of $19,604. For disabled people, the average yearly income for 2021 is around 16,000 dollars, about 1300 dollars a month. The MAC is made with this in mind.

Using the Robot Operating System (ROS), commonly available sensors, and our own chassis, we built the MAC to solve this issue and fill the gap in the robotics market for home use targeting elderly and disabled demographics. The main goal of the MAC is to automate transportation of objects around the household. Because we are targeting an older population, the other goals of the MAC is to have an easy to use interface, along with a reliable robot that won't require frequent maintenance or recalibration. Whether young, old, technology competent or not, the MAC should be easy to use by all.

Besides accessibility, the MAC is designed also with safety in mind, especially with software restrictions on operation to ensure the MAC does not damage itself or its environment. The MAC needs to ensure the safety of the user by maintaining safe following distances, give extra clearance to obstacles, and operate slow enough that it can stop or avoid obstacles without failure. All of these motivations combine into our main objective: to improve the lives of disabled and eldery people by making living on their own more convenient and possible, to avoid the need for assisted living.

## II. Previous Work

### A. Related Products and Supporting Technologies

There are plenty of examples across various industries of both autonomous and modular vehicles such as the MAC. Such industries include manufacturing, distribution, delivery, and retail applications. The examples listed below demonstrate the growing demand for such products in relation to the utility they bring to various enterprises. The focus in highlighting the examples below is to not only emphasize their importance but also discuss the various technologies they make use of.

#### *1. Manufacturing and Distribution*

Products such as Canvas Technology Carts, Fetch Robotics CartConnect AMRs as well as Twinny NarGo and Targo autonomous carts are all examples in this category. Autonomous carts in this sector make use of full vision systems which perform object identification, tracking and action prediction. They are built to be highly robust, loading and storing heavy equipment to transport across factory or warehouse floors. A variety of integrated

sensors allow them to navigate the environment efficiently, reducing the likelihood of accidents and congestion. Additionally, these autonomous carts typically integrate with other enterprise software such as inventory trackers or other robotic systems present [1]. Data collected by these vehicles is often passed to these other systems, optimizing the paths they travel as well as time spent completing various jobs.

*2. Delivery*

The delivery sector features autonomous vehicles such as Postmates' Serve (now under Serve Robotics), Amazon's Scout, FedEx's Roxo, and Starship's autonomous delivery robots. These autonomous vehicles are focused largely on transporting goods over long distances [2], and most are currently being tested in limited urban areas, as the infrastructure required to support suburban/rural transportation via these vehicles is both lacking and expensive. Their load capacity on the other hand is much less than that of its counterparts in manufacturing and distribution, ranging between 22 to 100 pounds. Some are being used to deliver mail or packages while others are used to deliver food or other services. Technologies such as LiDAR, cameras for vision (including obstacle detection and avoidance), and GPS are used. They are built to navigate on sidewalks or other similar paths rather than major roadways. These products help delivery companies reduce the number of couriers overall and simplify delivery routes for couriers.

*3. Retail*

In retail, some autonomous vehicles include E-mart's Eli, Amazon's Dash Cart, and the Caper Cart. Such products are modeled after modern shopping carts, and as a result tend to range between sizes of small to large shopping carts. They tend to store maps of the stores or locations they service, directing or navigating customers to goods they want. Similar to products in the manufacturing and distribution sectors, these products integrate with inventory systems to provide customers with real-time data of what goods are available. Some let customers pay for the goods they have in their cart using card readers and weight sensors while others make use of online accounts (such as Amazon) to make your payment. Additionally, some of these products automatically queue for charging stations as well as return to traditional cart-lineups when fully charged [3]. Such products reduce the traditional congestion of checkout lines while also making for a contactless shopping experience, targeting higher customer satisfaction.

## III. Subsystem Explanation

### *A. Mechanical*

This section provides a brief overview of the mechanical aspects of MAC. Despite the focus and nature of this project being directed towards the electrical and software aspects, we've elected to include a section to discuss the mechanical subsystems as well as complete assembly, being that they play a critical role in meeting the original requirements and specifications set forth by our team. The mechanical subsystems consist of the following: chassis, scissor lift, electrical board, sensor mounting (as well as enclosures), and interchangeable or modular parts.

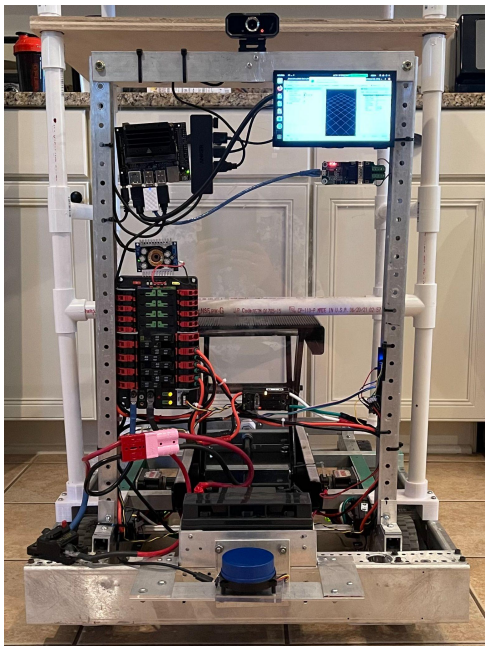

Fig. 1. Block Diagram of Remote and Receiver Communications

*1. Subsystems*

The chassis was modeled after FIRST (For Inspiration and Recognition of Science and Technology) Robotics' standard "Kit of Parts" chassis, the AM14u4 six wheel, drop center drive. The chassis sports treaded plaction wheels, and is belt driven from the center wheels on each side. A pre-fabricated gearbox, specially designed for the am14u4 allowed for two 2.5in CIM motors to drive each side. The load floor is made of 1 x 1 inch 6061 aluminum box tubing, providing structural support, weight distribution and mounting points for the other mechanical subsystems.

The scissor lift is a pre-built motorcycle jack, which is one-stage, screw driven. The lift can support up to 1100 pounds, reach a height of 14.5 inches, and has a platform surface area of 14.5 x 9 inches. The lift is driven by a 100:1 VersaPlanetary gearbox and BAG motor assembly, using a 3D printed hex shaft coupler for direct drive.

The electrical board consists of two sections of 1 x 1 inch aluminum c-channel, supporting a vertically mounted section of polycarbonate, with dimensions of 20 x 17.5 inches. The majority of this project's electrical components would then be mounted to the polycarbonate board, as discussed in the following section. The polycarbonate board is mounted 6 inches above the load floor to allow for ease of replacement and access for the 12V battery below.

Our LiDAR was mounted on top of a bottom plate which formed a "U" shape around the battery. The aluminum flooring provided a contact point ahead of the LiDAR should MAC collide with obstacles in the environment. Our camera was mounted at the top of the electrical board using an aluminum crosspiece in order to maximize its effectiveness at detecting people in the environment. Our IMU was centered and mounted underneath the bottom plate of the scissor lift, providing a central location to measure from. A single limit switch was mounted against the pre-built motorcycle jack, stopping the lift from retracting past a certain point. Lastly, the wheel encoders were mounted directly to the exposed output shafts of the chassis gearboxes, as intended by the manufacturers.

The interchangeable or modular part constructed was a two-tiered tray made up of plywood and PVC. PVC t-joints and end caps were used to add cross-sectional support between the PVC legs and mounting points to the plywood for each leg, respectively. Four 3D printed slotting points for each leg were mounted on the load floor. These slotting points allowed for easy removal of the modular part while also providing enough support to keep the part steady.

*2. Complete Assembly*

The load floor of the chassis, built using 6061 aluminum box tube, would provide mounts for the rest of the mechanical subsystems in this project. The scissor lift would mount directly to the two longer cross-sections while the electrical board would be mounted to the shorter pieces which simply provided mounting points. The 3D printed slotting points for the modular parts were designed to mesh with the original chassis and added box tube load floor, using both for structural support (to which then modular parts would slot into). Mounting points for various sensors has already been covered above.

*B. Electrical*

This section details the electrical design aspects of MAC. The design and the components that went into achieving its goals. As well as the devices that we designed when current solutions were not available.

*1. Processing*

There are several features that MAC is designed to be able to accomplish. From scanning and mapping an environment to camera object detection to smooth control of its chassis to transport cargo. Additionally, the cost of MAC should stay reasonable for the average individual to acquire if MAC is expected to perform in the home environment. Furthermore, MAC needs to remain safe at all times and be capable of actively handling problems it encounters, if software crashes or the battery is starting to run low MAC should have a safe way of handling dependencies. For these reasons the decision was made for all of MAC's processing to be done onboard with a multi-core single board computer. This allows MAC to perform several different tasks, such as driving the motors and mapping its environment, at the same time which is necessary due to the nature of the project.

After much research of single board computers that currently exist on the market we determined that the Nvidia Jetson Nano [7] would be appropriate for our needs. The Nano offers a 4 core ARM processor as well as a 128 core GPU that will be put to good use for doing object detection and environment mapping.

*2. Power and Distribution*

The two components that make up the power and distribution aspects of MAC are the 12V battery and the power distribution panel.

MAC uses a 12V battery system for several reasons. The first and most important reason is that our sponsor was able to provide the batteries. Although we chose to use this battery system because it was provided at no cost; if we had the choice from the beginning we probably would have ended up with a similar solution. The batteries provided are of small form factor (~6" cube) which are able to provide 18Ah which, after testing, is more than enough power to meet our goals for battery life at both loaded and unloaded configurations. Additionally, the batteries are of a sealed lead acid chemistry. Both of these features are appropriate for our applications for several reasons. Being a sealed construction is important because MAC is designed for use inside the home. If a chemical such as lead acid were able to leak and/or vent into a home the chemical can pose health risks to both people and pets inside. This risk is unacceptable and thus a sealed design is required. As for being lead acid, this is advantageous

because of its low cost and wide scale use. In the event a battery runs flat it can be cheaply and readily replaced.

The power distribution panel has a similar story to the battery. Specifically, it is an AndyMark Power Distribution Panel (PDP) [8] and if we had the choice we would have selected a similar product. The PDP connects directly to the battery and has several fused channels with different current ratings to fit different needs. Additionally, what makes the PDP a good option for our project is that the PDP has built in CAN communication for monitoring. CAN connections are of great use because they allow us to monitor the current that each fused channel is consuming, allowing us to measure power consumption as well as identify if components are consuming too much current.

*3. Sensing*

In order for MAC to safely transverse its environment it has to have a good understanding of its environment. Thus the importance for appropriate and effective sensors is demonstrated. The main sensors that MAC uses to detect its environment are as follows: LiDAR, Camera, Limit Switch, IMU. We will now discuss what each sensor is and its big picture purpose for MAC.

Light Detection and Ranging, or LiDAR for short, is an optical sensor that is used to measure distances. The underlying principle of operation for LiDAR is to pulse a laser and measure the time it takes for the laser to return. With this time measurement and knowing the speed of the laser you can simply calculate the distance the detected object is from the LiDAR. LiDAR themselves are efficient ways to measure distances accurately but for the purposes of MAC, we must be able to scan an entire environment. To accomplish this a LiDAR is mounted onto a spinning disk which allows for 360 degree range measurements. 360 degree measurements are critical because it allows MAC to update its internal map while its position changes. If a fixed direction LiDAR is used MAC would have to spin in a circle to everytime it wants to update its map after it moves.

In order for MAC to see users for its following feature we determined to use a camera based object detection model called YOLO, more details on why we chose this model can be found in the software section of this paper. The software works by analyzing images that are captured by the camera; thus a camera of high fidelity that can capture large amounts of spaces is critical. After considering several different cameras available on the market considering both its capability as well as keeping cost in mind we determined a camera that is capable of recording with a resolution of 1920 x 1080 at 30 frames per second. Additionally, after considering the size of MAC and the size of the typical home environment MAC will find itself in, a 110 degree field of view is appropriate.

MAC needs a way of determining when the lift has reached its top and bottom positions. This is important because if the lift motor is left on too long both going up and going down damage to the lift and/or the motor can occur. Furthermore, if cargo is loaded onto MAC's lift and the lift were to be over driven the damage can result in a potential safety issue. This is not an acceptable situation so it was decided that MAC's lift system should be run in a closed loop configuration. To accomplish this we used two simple limit switches mounted to the lift at different locations. One is mounted in a way to be pressed when the lift is at its lowest position and the other mounted in a way to be pressed when in its highest position. This solution simplifies the code as well, the Jetson just needs to look at the state of the switches and if neither are pressed the lift can turn both ways. If one or the other is pressed the lift can no longer turn in said direction.

Finally, MAC maps out the room it is in and then measures its movements to keep track of itself. This movement measurement is important for MAC to stay aware of its location in order to achieve the high levels of precision in navigation. The more precise measurements that can be made the more precise MAC will be. With this in mind, we decided to incorporate an inertial measurement unit into MAC's sensor suite. An inertial measurement unit (IMU) is a collection of gyroscopes, magnetometers, and accelerometers. The combination of these sensors in certain arrangements allows for the orientation of MAC to be measured directly.

*4. User Controls*

Keeping in mind MAC's goals to be simple to use we determined that there should be multiple ways a user can interface with MAC directly or indirectly. There are several different ways to accomplish both of these tasks but the underlying principle will be similar. The direct interfacing involves user inputting commands to an interface on MAC itself, and the indirect interface involves user inputting commands to some sort of device that sends a signal to MAC. After much consideration we determined that the best method for direct interfacing is a touch screen, and for indirect interfacing is a dedicated handheld remote. We will now explain how we arrived at this conclusion for both direct and indirect interfacing.

There are several ways of accomplishing direct interfacing. The simplest and cheapest method would be a few switches mounted onto MAC that are used specifically for dedicated actions. Initially, this design would have sufficed. However, a way to interface with the Jetson Nano itself without having to have a separate keyboard and

monitor connected is useful for prototyping. With that in mind when we researched different types of screens that we could integrate into MAC the idea of a touch screen was proposed. A touch screen is advantageous for multiple reasons. The first reason being that users can not interface with MAC in one simple package which would have required a mouse and keyboard can now be done simply by touching the screen. This reduces the total number of parts needed and lets development time be spent on more important things like developing the UI of the display itself.

For indirect interfacing there were two avenues that were considered. A mobile app and a wireless remote. A mobile app has several advantages and disadvantages. The biggest advantage of a mobile app is all of the hardware responsibilities are provided by the user, as long as the user has a phone that is WiFi enabled then with enough software development an app can be implemented. The biggest disadvantage of a mobile app is the amount of software that has to be written is much higher than on a dedicated remote. Software needs to be written for multiple different operating systems depending on the user and needs to be constantly updated as the phone operating system gets updated. On the other hand a dedicated remote requires hardware and software constructed for it, but overall the path is more simple and is a one and done solution.

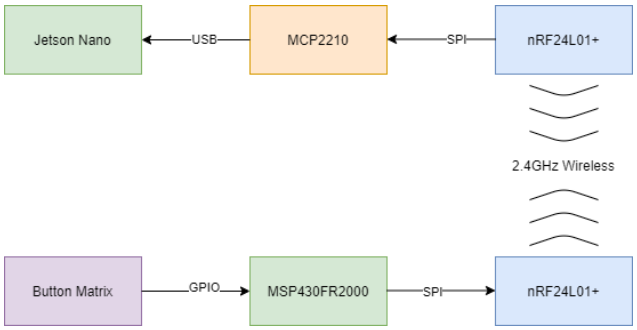


Fig. 2. Block Diagram of Remote and Receiver Communications

With these considerations in mind we decided to develop our own dedicated remote. The idea is to create something similar to a wireless USB mouse so that we can just plug into one of the Jetson's USB ports and the remote can easily send data. We looked at several currently existing USB remotes on the market and determined that they would not be appropriate because the form factor was built for televisions, MAC does not need a remote with volume buttons. So the indirect interface will consist of the remote itself and a USB receiver. The receiver will connect to the remote wirelessly using the 2.4GHz ISM band. This band is used because it is free to use and requires no special certifications, additionally, there are several low cost IC's on the market that are developed for this band. After much research we determined that the nRF24L01+ single chip transceiver [4] was a good pick because it requires one chip for both sending and receiving and breakout modules for it are readily available at a low cost. The remote will consist of buttons, a microcontroller, batteries, an ON/OFF switch and the nRF24L01+ module. The microcontroller will handle the reading of the buttons and configuring and commanding the nRF module to send button presses to the receiver. The buttons will have the following functions: Forward, Reverse, Turn Left, Turn Right, Raise Lift, Lower Lift, LOS Follow, Stop, and Go Home. In order to limit the number of GPIO pins needed on the microcontroller a button matrix [5] was used. The remote is battery powered with an ON/OFF switch to conserve battery power when not in use.

For the receiver we wanted the Jetson Nano to be the one running the show so that all the programming can be done on the Jetson simplifying the design. To accomplish this, after much research we found the MCP2210 [6] which is a dedicated USB to SPI converter. SPI is how the nRF module is both programmed and commanded to send and receive messages. By using the MCP2210 we can interact with the nRF module directly from the Jetson. In the end the user should just plug in the receiver to the Jetsons USB port and turn the remote ON and everything should be ready to go.

### *C. Software*

The software that runs on the Jetson Nano is shaped by the Robot Operating System 2 (ROS2) framework. ROS is the industry standard for the development of robotic projects and makes up a significant amount of our software architecture. Using the ROS publisher-subscriber architecture, we were able to develop a scalable and flexible system that allows the MAC to operate in a few different modes while maintaining safety considerations. The network of asynchronous nodes publish data and subscribe to data published by other nodes to several topics that act as the channel in which the nodes communicate. To summarize, the MAC is able to be run in tele-operated mode with a remote-control or in autonomous mode using the NAV2 navigation stack provided by ROS. In addition, the software also provides safety mechanisms when operating the lift to ensure the robot is not allowed to move while the on-board scissor lift is being operated.

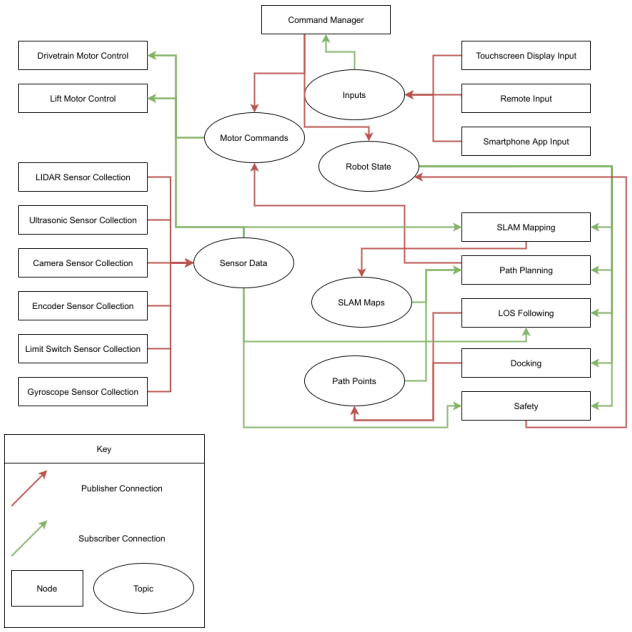


Fig. 3. ROS Node Topology Diagram for our software

*1. Odometry, Mapping, and Navigation*

A significant part of the software and a building block for running the ROS navigation stack is the odometry system implemented using the wheel-encoders on board the chassis. The wheel encoders provide velocity data from the center wheels on the left and right side of the chassis. The left and right wheel velocities are first converted to the distance traveled by each wheel and then averaged to find d_center which is the length of the arc between the wheels. Due to the non-holonomic (the MAC cannot strafe or move in the y-axis) nature of the MAC, the distance traveled in the y-direction is set to 0. The left and right wheel velocities can also be used to derive the angular velocity by finding the change of angle or current orientation ($\boldsymbol{\phi}$) of the robot in radians. We will then use the previous orientation of the robot ($\boldsymbol{\theta}$), d_center and the previous x, y position of the robot to find the new x`, y` position. The new orientation of the robot is found similarly by adding $\boldsymbol{\theta}$ to $\boldsymbol{\phi}$ which gives the new orientation of the robot $\boldsymbol{\theta}$` [10]. This data is then published to a topic that will be subscribed to by the robot_localization package that provides an Extended Kalman Filter (EKF) to filter and tune provided odometry data.

An Extended Kalman Filter is a popular and most often used approach in the industry for conducting state estimation for non-linear models such as navigation systems and GPS. The robot_localization package in ROS uses an EKF to tune and fuse odometry data by conducting probabilistic estimates [9]. Our system contains both an IMU that is being used to measure angular velocity in the yaw direction (rotation with respect to the z-axis) and wheel encoders that measure velocity as mentioned above. These are both continuous sensor measurements that will change as the state of the robot changes as well. The robot_localization takes in the type of data being fused and from which source. Therefore, in the case of the MAC we will have the wheel encoders providing data in the x, y and yaw directions with the IMU providing angular velocity.

ROS contains a built-in navigation stack called NAV2 that provides the functionality for developing a map of the robot's environment, localization nodes and a plethora of navigation nodes for autonomous navigation in a robots environment. To generate a map of the MAC's environment we will be using our LiDAR sensor mounted at the front of the chassis with the Simultaneous Localization and Mapping (SLAM) algorithm that is implemented in the SLAM toolbox node in NAV2. The SLAM map will be initially generated via tele operating the robot through the robot by the user in order to create an accurate map of the household environment. Once the map is sufficiently generated it will be used in the localization mode of the SLAM toolbox node in order for the robot to be properly localized within the map. Finally, this allows for various navigation nodes to generate a path for the robot to follow within the map.

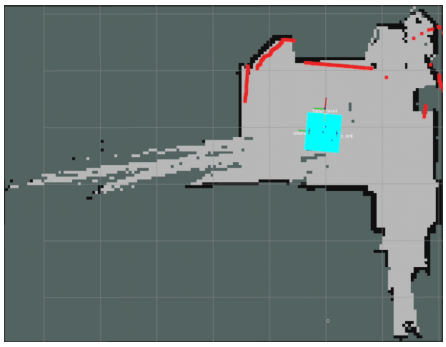

Fig. 4. MAC SLAM map of household

*2. Line of Sight (LOS) Following*

The MAC will feature a forward facing camera for detecting a single person to follow for our LOS (line of sight) following feature. The camera will be connected directly to our Jetson Nano which will be running a deep learning model called YoloV5 ("you only look once") [11]. YoloV5 is a fast and accurate fully convolutional neural network for object detection and localization, out performing other models for the same task by a factor of

4. Typically, it is used for object detection of everyday household items (COCO dataset [12]), but we will be using it just to detect what is needed: people and those in wheelchairs or scooters. The network will output bounding boxes of the detections, which will then be transformed into a distance and angle to calculate a goal pose for the MAC to reach.

The distance to an object (in this case, the detected user) is calculated using the triangle similarity formula. First, given an object of known distance *D*, width *W*, and width in pixels *P*, the focal length *F* can be calculated (1):

$$F = \frac{P * D}{W} \tag{1}$$

Once *F* is known, the distance to detected users can now be calculated based on the bounding box calculated from YoloV5. The bounding box gives the top left and bottom right coordinates in pixels, which then the width in pixels is calculated by subtracting the bottom right pixel width coordinate from the top left pixel width coordinate. With the known pixel width *P*, the distance to the object can be calculated by solving for D in (2):

$$D = \frac{W * F}{P} \tag{2}$$

Finally, the angle deviation from the center of the camera needs to be calculated with the distance in order to properly set a goal pose for the MAC to drive to. The angle deviation is calculated by calculating the number of pixels from the center of the camera to the center of the calculated bounding box, and multiplying by the number of degrees per pixel. The degrees per pixel is calculated by taking a few measurements of the difference in pixels from the center of the bounding box to the center of the image in the width axis, the measurement of the angle in actuality using a protractor, then dividing the actual degrees by the pixel offset from the center. A few measurements were taken and the average calculated.

With those values calculated, the LOS following node will take these measurements with the current pose of the MAC, which contains the x, y, and θorientation of the MAC. The distance *D* has 2 feet subtracted from it in order to ensure the MAC doesn't collide with the user. If the distance is less than 2 feet to begin with, the LOS node will not send a new goal pose for navigation. The updated θ, θ' is calculated by adding the negated angle (to swap the direction) calculated from the formula above. From this the update for x and y, $\Delta x$, $\Delta y$is calculated by taking the calculated distance, *D,* and the following formulas:

$$\Delta x = sin(\theta') * D$$
$$\Delta y = cos(\theta') * D \tag{3}$$

(4)

The reason why the x update uses sine instead of cosine is because x offset is the opposite side of the right triangle created from the angle θ'and distance D makes. With the goal pose calculated, this is sent to the navigation node to plan a path to the new goal and send the velocity commands to the motor controllers to reach it.

### *3. Safety Features*

The MAC will have a number of features in order to ensure the safety of not only the user, but the user's home, other people in the home, and any other pets or obstacles. These features will use the existing sensors to impose constraints on the MAC's movement including speed, distance to obstacles, and operation of multiple functions at once.

For speed, the MAC will travel a maximum of 2.5 feet per second, with a slower 2 feet per second when following a user. This will ensure that any load won't be displaced while traveling while also leaving plenty of room from the user to avoid collisions or being tripped. The lift will travel at most 2 inches per second, which from our testing the lift only travels at 0.5 inches per second. With distance, the MAC will maintain at least 6 inches from obstacles in front of it, while the LOS following will be at 2 feet from the user.

The MAC has different states in which it is currently operating in, and for safety the MAC cannot be in two states at once. This means that if the user is operating the lift, the MAC will not respond to input for teleoperated or LOS following modes. Similarly, when the MAC is driving the lift cannot be operated, nor can the other modes be toggled. In the feature this would include the docking mode as well.

## IV. Testing

### *A. Test Methodology*

In general, due to the complex nature of building out a robot from the mechanical, hardware and software subsystems there was testing done at each stage or milestone. Testing was done on the physical sensors and components such as LiDAR, wheel encoders, motors, ultrasonics and others as well as the software drivers/processors for these sensors. Before integration within the system, we iteratively tested these sensors independently with sample test code since many required different types of communication interfaces such as CAN, analog, and serial. Once sensors were individually tested they were iteratively integrated into the system dependent on the type of feature being implemented. Integration tests were then conducted to verify that the components and

sensors are able to function together in the ROS node network. Integration testing revealed critical bugs and areas of improvement for the software.

The final part of our testing was to fully evaluate the system and compare our test results to the specifications and requirements outlined before the building of the project.

To delve deeper into our testing methodology we will briefly outline the test case for each specification. To test the cost specification, all purchases made by the group were totaled and compared to the initial budget set out. For the size requirement, the robot was measured and then also tested to verify that it is able to fit within a tight kitchen opening in a household. The durability was tested by dropping 12 pound weights from 8 inches to validate that there was no damage or harm done on the system. The lift weight load was validated by having group members stand on the lift while it was operated. Lift speed was validated by measuring how long the lift took to reach maximum height while being operated at maximum power. The speed of the MAC was tested by driving the MAC at a maximum speed of 0.8 m/s and viewing the velocity feedback data from the wheel encoders. The stopping distance was validated by placing a line of tape to signify the stopping point and measuring the distance where the MAC stopped after releasing the drive button. The range of the remote control was tested by configuring two NRF modules with two arduinos and gradually increasing the distance between the two until communication became unreliable. The intelligence was tested by verifying the reliability of the SLAM mapping and navigation notably as well as the speed of computation for path planning and replanning. The battery life was calculated by measuring the current draw of the MAC while driving and deriving the hours of operation with no load or full load based on the 18Ah rating. The charging time was simply tested by draining the battery to a 11.6 V dead voltage and measuring the time until it was fully charged. The ease of use specification was untested due to shortcomings in completing and achieving remote functionality. Ease of use would have been tested by finding a set of outside users and gathering feedback on the remote control design.

### *B. Test Results*

Overall the system testing was successful with several test results exceeding the specifications and requirements. The cost, size, weight load, and durability all met the specifications.. The testing done for the intelligence specifications had mixed results with succeeding at accomplishing any path planning in 5 seconds or less but also failing in having the MAC navigate to a set goal pose. Overall, the MAC accomplished several on-board processing tasks such as generation of a SLAM map reliably, implementing EKF's for fusing sensor data and performing both local and global localization. The testing results table is presented down below.

TABLE 1:
SYSTEM TESTING RESULTS

| Requirement | Test Result | Test Status |
|---|---|---|
| Cost | $1,050 | Success |
| Size | 24 x 26 inches with a 36 inch diagonal | Success |
| Durability | Withstood force of 12 pounds dropped from 8 inches | Acceptable |
| Weight Load | Lift was able to withstand up to 200 pounds of force | Success |
| Lift | The scissor lift operated a speed of 1/2 inch per second | Acceptable |
| Speed | Average drive speed measured to be 2.5 feet/second by wheel encoders | Success |
| Stopping Distance | Stopped at a maximum of 4 inches resulting in better performance than initially expected | Success |
| Intelligence | 1. Path planning and replanning both took 2-3 seconds 2. MAC did not navigate successfully | 1. Success 2. Failed |
| Ease of Use | Untested | Untested |
| Range of Remote Control | The NRF module was able to communicate within a distance of approximately 150 feet | Acceptable |
| Battery Life | The battery life measured with no load was 2.75 hours and 1.9 hours with a full load | Success |
| Charging Time | Charged full from empty in 2.6 hours | Success |

## V. CONCLUSION

The MAC aims to fill a gap in the home robotics market for an autonomous cart to move items around the home, further targeting the elderly and disabled demographic. MAC is mainly powered by ROS, the industry standard for autonomous robots, to handle odometry, filtering, SLAM, and navigation. We also incorporated safety features and AI into the ROS program using Python and YoloV5 to provide a line of sight following feature for the MAC to transport goods in nearly any situation. It is also robust enough to handle dropped items and loads well over 100 pounds while also having enough power to transport those loads. In the end we are on our way to accomplish our goal completely given more debugging time.

## Acknowledgement

The authors wish to acknowledge the contribution of Bartholomew Nash, who provided close to $750 worth of materials and resources to the project. Bartholomew Nash is a LCS teacher and Robotics Coach at Lake Minneola High School in Minneola, FL.

## References

[1] Lampinen, Megan. "Voith Joins CANVAS Technology at IMTS 2018 to Showcase Intelligent Robotic Solutions Collaborating for Process Automation." Automotive World, 6 Sept. 2018, www.automotiveworld.com/news-releases/voith-joins-canvas-technology-at-imts-2018-to-showcase-intelligent-robotic-solutions-collaborating-for-process-automation/

[2] Scott, Sean. "What's next for Amazon Scout?" About Amazon, Amazon, 23 Sept. 2020, www.aboutamazon.com/news/transportation/whats-next-for-amazon-scout.

[3] Anna. "E-Mart Introduces Eli Autonomous Shopping Cart." Robotics & Automation News, 4 May 2018, roboticsandautomationnews.com/2018/05/04/e-mart-introduces-eli-autonomous-shopping-cart/17113/.

[4] "Nrf24 Series." Nordic Semiconductor, www.nordicsemi.com/products/nrf24-series.

[5] "Introduction to Microcontrollers - Button Matrix & Auto Repeating." *EmbeddedRelated.com | Embedded Systems*, www.embeddedrelated.com/showarticle/519.php.

[6] MCP2210 USB to SPI Converter with GPIO (Master Mode). Microchip, www.microchip.com/en-us/product/MCP2210.

[7] "Jetson Nano Developer Kit." *NVIDIA Developer*, 14 Apr.2021, developer.nvidia.com/embedded/jetson-nano-developer-kit.

[8] *Power distribution PANEL*. AndyMark, Inc. (2018, August 24). https://www.andymark.com/products/power-distribution-panel.

[9] D'Alfonso, L., Lucia, W., Muraca, P., & Pugliese, P. (2015, July 17). *Mobile robot localization Via EKF and UKF: A comparison based on real data*. Robotics and Autonomous Systems. https://www.sciencedirect.com/science/article/abs/pii/S0921889015001517?via%3Dihub.

[10] *Jetson nano developer kit*. NVIDIA Developer. (2021, April 14). https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[11] Redmon, Joseph. YOLO: Real-Time Object Detection, pjreddie.com/darknet/yolo/.

[12] Lin, Tsung-Yi, et al. "Microsoft COCO: Common Objects in Context." Computer Vision–ECCV 2014 Lecture Notes in Computer Science, 2014, pp. 740–755., doi:10.1007/978-3-319-10602-1_48.