Auto-following Luggage Platform

David Chen

Lyuxing He

Final Report for ECE 445, Senior Design, Spring 2023

TA: Xiangyuan Zhang

02 May 2023

Project No. 47

Abstract

This paper describes a smart luggage platform that can identify its owner and automatically follow him or her, therefore not requiring any effort from the owner to carry the luggage around while traveling. The paper will start off with a discussion of the motivation, desired functionalities, and benefits of this project. This section will also introduce the high-level expectations of the overall system. Then, the paper will elaborate on the overall design of the project by covering the design details and requirements for each subsystem. After that, the paper will discuss the verification approach for each of the subsystem requirements listed in the previous section. Following that would be a brief overview of the cost and labor analysis of this project. Lastly, an overview of the accomplishments, uncertainties, ethical considerations, and future work of the project will be presented at the end.

Contents

1. Introduction

1.1 Motivation

Sometimes carrying the suitcase might be the most unsatisfactory part of a trip. This dissatisfaction can also grow into annoyance when the amount of luggage is too large to be carried without the help of transportation tools. Therefore, people have dreamed about a suitcase that can automatically track its owners on its own, without requiring external forces from the owner to steer it. There have been so-called "smart suitcases" made for sale with different features including USB-port for charging, GPS localization, etc. However, the price is too high for the public to afford [1], and only a few with exceedingly high expenses might be capable of achieving the fully automatic following feature. The product currently being marketed online that meets the demands described above with the lowest price of 799 EUR can be found here Functions of Airwheel SR5 Intelligent Suitcase [2], which is still too costly for normal families.

1.2 Solution

The paper proposes the Auto-following Luggage Platform (AutoLug) project that aims to solve the problem mentioned above. More specifically, the owner of AutoLug is able to register himself as an owner of AutoLug, and AutoLug is able to identify the registered owner uniquely using the camera equipped onboard and autonomously following the owner safely without collisions. The advantages of this solution compared to the "smart luggage" available in the market are as follows. First of all, AutoLug is much more affordable, as its overall cost is approximately 300 USD dollars and is much cheaper than those "smart luggage". Secondly, AutoLug is more versatile, as the nature of being a platform enables AutoLug to carry a wide range of different suitcases or even items other than suitcases. Lastly, AutoLug is a one-time purchase, as the owner is capable of switching suitcases to be carried on top to enjoy AutoLug's features, while the features of each "smart suitcase" are only subject to its own.

1.3 Visual Aid

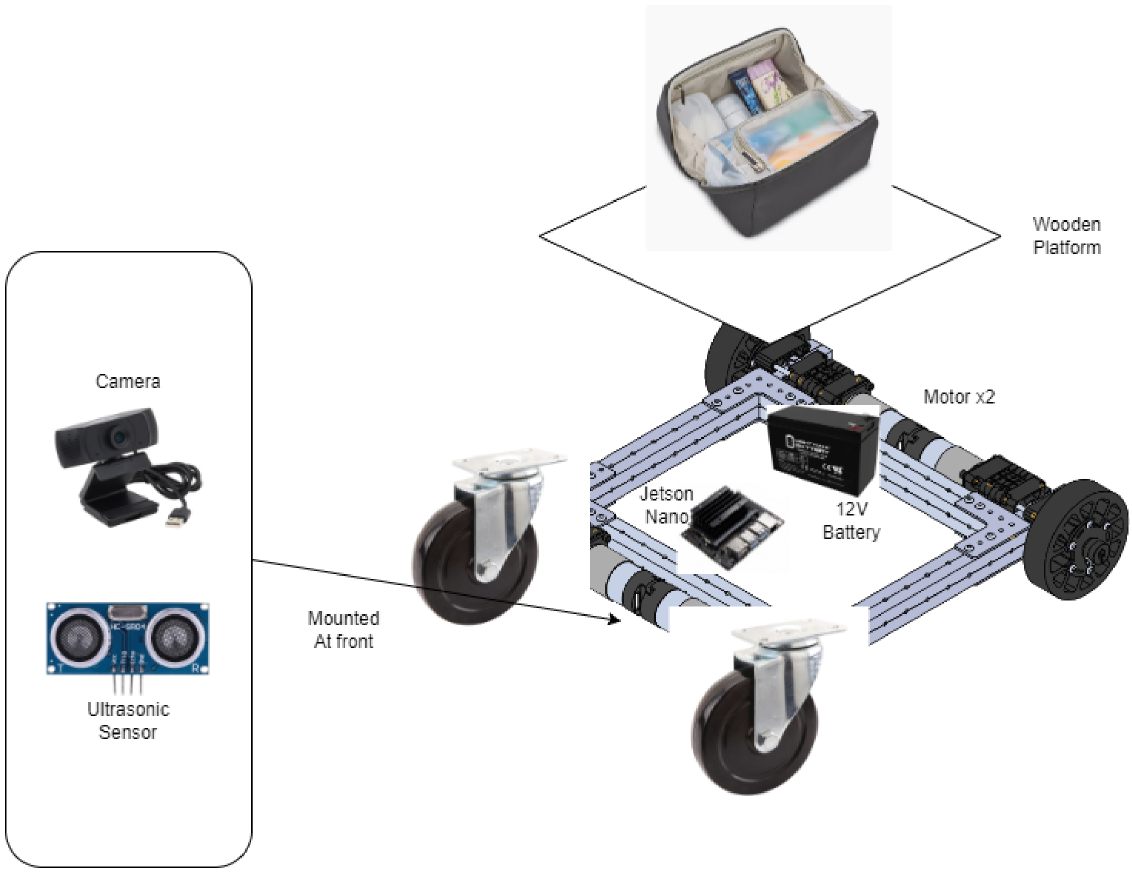

Figure 1: Visual Aid

1.4 High-level Requirements

- The machine is able to carry weights up to 20kg while maintaining a safe speed between 2-4 mph.
- The machine is able to follow the owner when the owner is in the camera frame and maintain a safe distance of more than 1 meter from the owner.
- If the owner is actually perceivable nearby, the machine is able to locate the owner (put the owner back into the camera frame) autonomously when camera tracking is lost within 10 seconds.
- The machine is able to avoid collisions with obstacles (objects of heights more than 1 inch) and humans 95% of the time.
- The machine is able to remain active (tracking the owner) for at least 30 minutes when the battery is fully charged.

1.6 Block Diagram

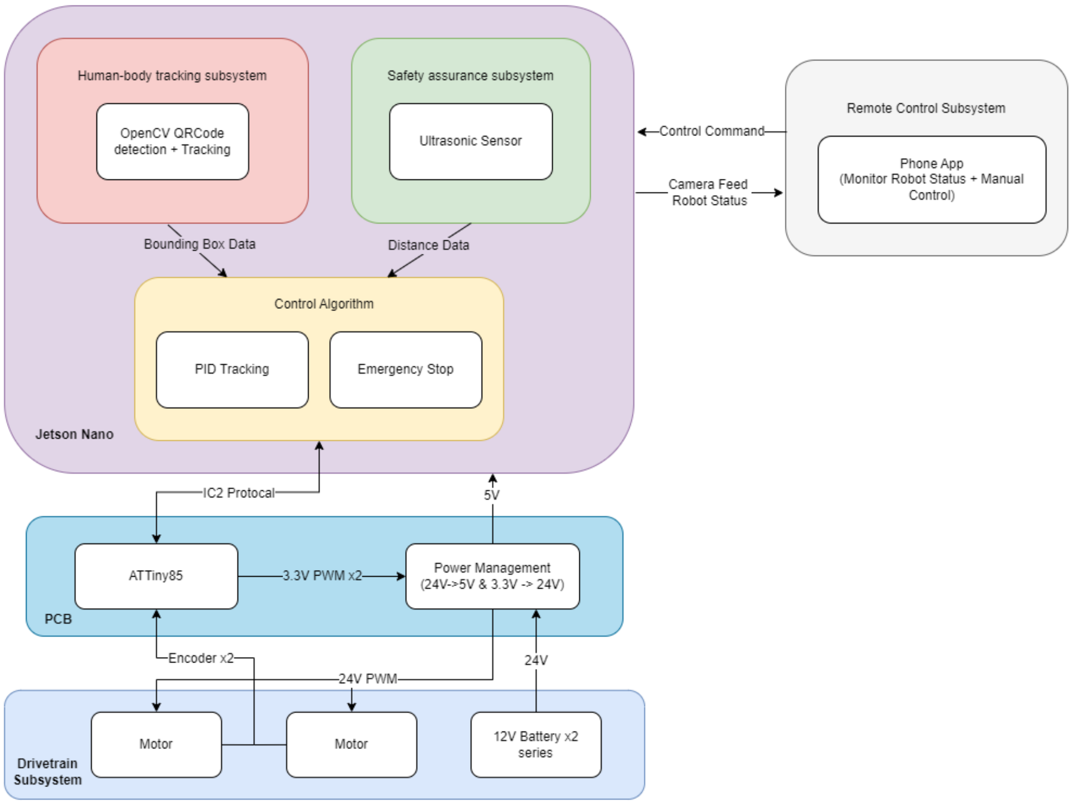

Figure 2: Block Diagram

1.7 Subsystem Overview

1.7.1 Drivetrain

Drivetrain is the mechanical component of AutoLug. It consists of 2 motors, 2 12V batteries, A PCB with a microcontroller, 2 wheels, and 2 caster wheels. It receives instructions and sends encoder data to Jetson Nano. It will carry weights up to 20kg and maintain a speed of 2-4 mph.

1.7.2 Control Subsystem

The control subsystem sits inside the software. It is a program that calculates the PWM output value of the motor in order to achieve the desired speed and steer the robot according to the data from the Human-body identification subsystem.

1.7.3 Human-body tracking subsystem

The human-body tracking subsystem identifies the owner in the camera frame. and provide bounding box data to Control.

1.7.4 Safety assurance subsystem

The safety assurance subsystem detects Obstacles in front of the robot using an ultrasonic sensor and sent alert to the control subsystem to stop the robot immediately.

2 Design

As shown in the block diagram above, there are 4 subsystems in our design. We will discuss each of these subsystems in detail in the following sections.

2.1 Drivetrain

2.1.1 Overview

The drivetrain will consist of two motors and two casters. A PCB with an ATTiny85 microcontroller is used to receive speed instructions and send encoder data to the Jetson Nano board. The microcontroller generates two 3.3V PWM signals based on Jetson Nano's instructions, which get converted to two 24V PWM to power the motors. The drivetrain's left and right motors will be independent, allowing the robot to turn at different speeds on the left and right. In addition, the drivetrain is also responsible for supplying power to the entire platform via two 12V batteries.

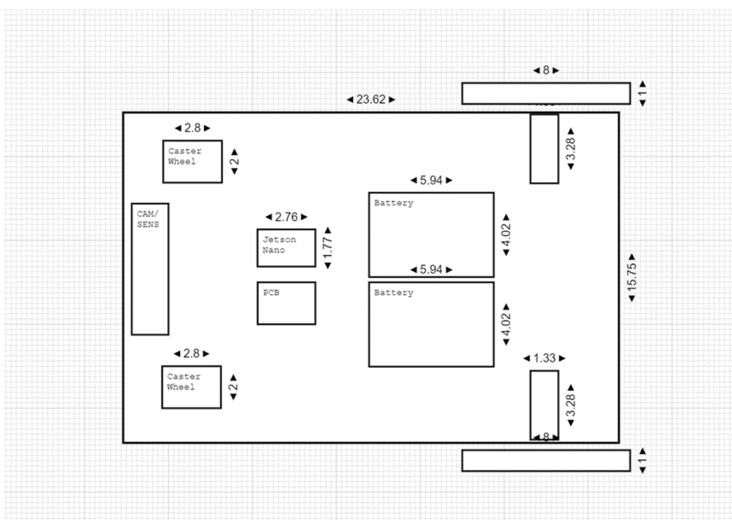

Figure 3: Layout

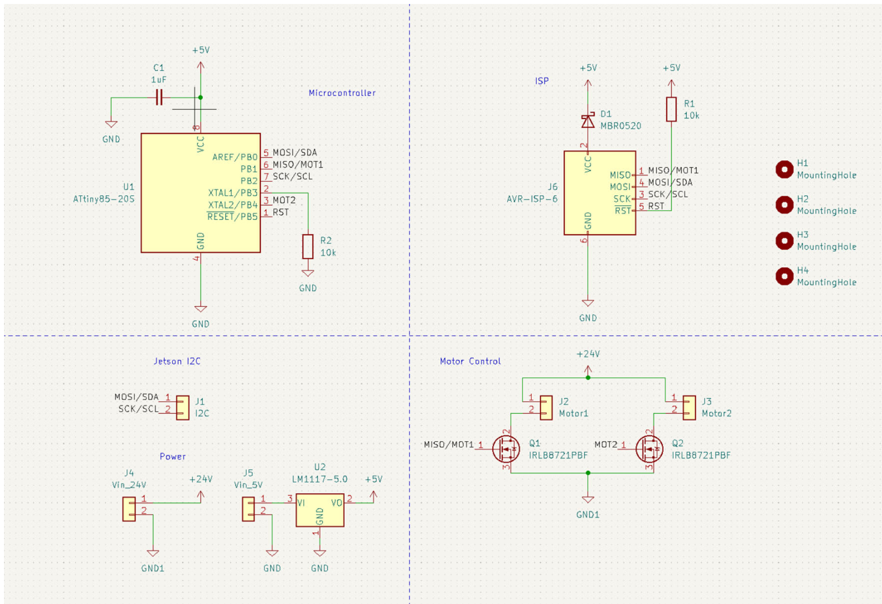

Figure 4: PCB Schematics

2.1.2 Requirements

The subsystem satisfies the following requirements:

1. Battery should sustain Autolug for running 30 minutes at 1m/s.
2. Autolug can still move at 1m/s when loaded with 20kg of weight.

2.1.3 Design Decisions

We choose to use two 24V 200RPM 23kg*cm stall torque motors as they satisfy our 20kg load requirement. 8-inch wheels are used to attach to the motors. To power the motors, we also need a 24V battery as well. However, after some searches, we conclude that it is most cost-efficient to use two 12V batteries and put them in series. To justify our component choices, the following calculations are made:

Our motor is rated for 200 rpm along with a 20cm diameter wheel. So we can derive its maximum velocity:

$$V_{max} = \omega \pi d = 200 \ * \ 3.14 \ * \ 0.2m/60 = 2.09\, m/s = 4.66 mph$$

The traction force that the two motors can provide can be calculated as follows:

$$F_{traction} = \tau / r = 23kg\ cm / 10cm = 2.3kg$$

The maximum coefficient of friction Autolug can handle before stall is therefore:

$$\mu = F_{traction} / F_{total\ weight} = 2.3kg / (20kg + 6kg) = 0.0885$$

Coefficients of Rolling Friction	
0.001 - 0.002	railroad steel wheels on steel rails
0.001	bicycle tire on wooden track
0.002	bicycle tire on concrete
0.004	bicycle tire on asphalt road
0.008	bicycle tire on rough paved road
0.006 - 0.01	truck tire on asphalt
0.01 - 0.015	car tires on concrete, new asphalt, cobbles small new
0.02	car tires on tar or asphalt
0.02	car tires on gravel - rolled new
0.03	car tires on cobbles - large worn
0.04 - 0.08	car tires on solid sand, gravel loose worn, soil medium hard
0.2 - 0.4	car tires on loose sand

Figure 5: Coefficients of Rolling Friction[3]

We can see from the chart that our design can handle most of the scenarios without stalling.
Next, we check if our battery configuration can satisfy our requirement in theory.
We will use two 12V 9 Ah batteries to power our entire platform. Using an estimated 80% efficiency in the circuits, we can derive the maximum possible battery life of our platform:

$$T = \frac{E_{battery}\eta}{P_{motors} + P_{jetson}} = \frac{2*12V*9\,Ah*0.8}{12W*2 + 20W} = 3.9\,h$$

Based on our calculations, all the requirements of the subsystem is met.

2.1.4 Design Changes

We didn't change the design and the components of this subsystem during the project. However, one improvement that we had to do is to wrap the I2C pins and the motor output pins with electrical tape. In the original design, they are right next to each other which caused a lot of interference on the I2C bus.

2.2 Control Subsystem

2.2.1 Overview

The control subsystem has two modes: tracking and locating. In the tracking mode, the human-body identification subsystem has found the owner and therefore the control subsystem should instruct the drivetrain to track the owner. Using the bounding box data calculated from the Human-body identification subsystem, we can calculate the deviation angle, and use PID to track and minimize this error. We will also use a separate algorithm to control the speed of the robot. Using an estimated distance value, we will speed up and slow down the robot accordingly as well. In the locating mode, the human-body identification subsystem failed to find the owner in the camera frame. Therefore the control subsystem should instruct the drivetrain to locate the owner. This is done by instructing the motors to run at different speeds to turn the platform around circularly in order to locate the owner in the frame.

2.2.2 Requirements

The subsystem satisfies the following requirements:

1. Maintain a set velocity v where $0m/s \leq v \leq 2m/s$ and a straight trajectory.
2. The platform turns accordingly when the input deviation angle changes.

2.2.3 Design Decisions
To maintain the speed of each motor, the encoder senses the revolution count and we can take the derivative of this count to get the velocity of the motor. We chose to use PID controllers to control the actual PWM value (0-255) that we send to Drivetrain PCB through the I2C protocol because PID controllers are easy to implement and have relatively good performance.

2.2.4 Design Changes
We didn't change the design and the components of this subsystem during the project.

2.3 Human-body tracking subsystem

2.3.1 Overview
We will use Yolo5 for human recognition and segmentation to produce bounding boxes. Each bounding box will be made into gait silhouettes and used for a gait-matching algorithm to identify the owner of the suitcase. This subsystem will return a boolean value that represents whether the owner is found in the current frame, and, if true, the correct bounding box of the identified owner. This boolean value and the offset will be sent to the control subsystem for controlling the drivetrain.

2.3.2 Requirements
The subsystem satisfies the following requirements:

1. Achieve at least 90% accuracy in identifying the owner, regardless of the environment settings.

2.3.3 Design Decisions
To take advantage of the motion information of the owner from the camera, we believed that gait recognition is the best approach to tackle the human-body identification and tracking problem. Since it is really crucial for the owner to easily train AutoLug to recognize him/herself, the sample efficient technique is the first priority during our algorithm search. After reviewing multiple state-of-art gait recognition algorithms, GaitSet [4] is found to be the most suitable algorithm for our project. GaitSet is extremely sample efficient, as it only requires a set of gait silhouettes, does not require them to be consecutive, and is not dependent on the permutation of each silhouette image. GaitSet is also fast training as it only requires 7 min to perform eval on OU-MVLP. In addition, it's very effective in terms of accuracy, as it achieves Rank@1=95.0% on CASIA-B and Rank@1=87.1% on OU-MVLP, excluding identical-view cases.

2.3.4 Design Changes
Several severe issues were found when trying to implement the GaiSet-based human recognition subsystem on the AutoLug. Since the platform is supposed to be tracking the back of the owner, the back-view gait image is entirely different from the testing case of the side-view gait image (which is more informative). Secondly, since the camera is mounted at the bottom of the platform, if the platform is getting too close to the owner it will not be able to capture the entire gait motion and therefore fail to perform identification. Therefore, we decided to switch gears and use a QR-code-based method to tackle the human identification and tracking problem. More specifically, the owner will first register him/herself with AutoLug, and AutoLug will return a unique QR code that represents the owner, and the identification

problem is reduced to QR code detection and QR code decode problem. Using the QR code library from OpenCV to detect and decode the QR code, the next step is tracking the moving QR code (thus the owner). This is done by first replying to the bounding boxes returned by the QR code detect-decode method. If the owner is lost, then the latest bounding box will be used to initialize the OpenCV object tracking method to track the corresponding object in the current frame.

2.4 Safety assurance subsystem

2.4.1 Overview

The ultrasonic sensor equipped will report the distance to obstacles as well as a boolean value that represents safety with respect to possible collisions. The robot will stop immediately if the returned value is false to avoid collisions.

2.4.2 Requirements

The subsystem satisfies the following requirements:

1. avoid collisions with obstacles (higher than 1 inch) with 90% safety distance (50cm*0.8=40cm) away from the target

2.4.3 Design Decisions

We used an ultrasonic sensor as it is low cost and easy to implement in software. Since our robot will only drive forward, only a single front ultrasonic sensor is used to save costs. To override the normal behavior and stop the motor immediately when the sensed distance is smaller than the safety distance, I implement logic that forces the PWM to be 0 in the controller code right before the PWM signal is sent to the microcontroller.

2.4.4 Design Changes

We didn't change the design and the components of this subsystem during the project.

3. Design Verification

See Appendix A for Requirement and Verification Table.

3.1 Drivetrain

3.1.1 Torque Verification

To verify whether the drivetrain subsystem can provide enough torque ($10.4\ N * m$ according to to Appendix A) to drive the entire platform up to a maximum speed of 2 m/s, we first tried to directly measure whether the motor is able to provide a torque of $10.4\ N * m$. As illustrated in Appendix A, we hung weights down the wheel and started the motor, and then we examined whether the motor is able to rotate the wheel and counteract the weight. To see whether one motor can provide $10.4\ N * m$ torque with approximately 10 cm wheel, the weight needed should be 10.6 kg approximately. As we gradually increased the weight hung down the wheel from 2 kg to 15kg, we did not see any sign of the motor having difficulty lifting up the weight. This is an indication that our motor is powerful enough to drive AutoLug under a maximum speed of 2m/s with no problem. We later proved our assumption by loading a total weight of 20 kg onto AutoLug and driving it under 2 m/s using the manual control mode. Indeed, AutoLug can easily achieve the maximum speed of 2 m/s easily, indicating that we've verified the torque requirement stated in Appendix A.

Figure 6: AutoLug carrying a load of 20kg

3.1.2 Battery Life Verification

To verify whether the drivetrain subsystem can provide enough battery power to sustain AutoLug to run at least 30 minutes, we decided to instruct both motors to keep running and Jeston Nano active for an hour and monitor the voltage reading of the battery output. According to the datasheet provided by the manufacturer [5], the 12v battery needs to be recharged when its output voltage drops below 11v. Therefore, for our 24v battery or 2 12v batteries in series, we need to recharge our power source when its output drops below 22v. We then instructed both motors to keep running and Jeston Nano active for an hour while monitoring the voltage reading of the power source, and the data are recorded in the following table:

Table 1: Voltage reading of power source over 1 hour window while remaining AutoLug active

Discharge Time (min)	15	30	45	60
Voltage Reading (V)	24.21	24.06	23.96	23.87

As we can see, the voltage reading of our power source never drops below 22v, which is an indication that we've verified the battery life requirement stated in Appendix A.

3.2 Control Subsystem

3.2.1 Set Desired Speed Verification

To verify that the motor controllers can maintain set speeds for both motors, we first set the desired speed of the motor in code to 1m/s and observe the encoder measured speed. If the average speed is close to 1m/s, we say that the motor achieved the requirement. We can perform this test at various desired speeds and observe its behavior.

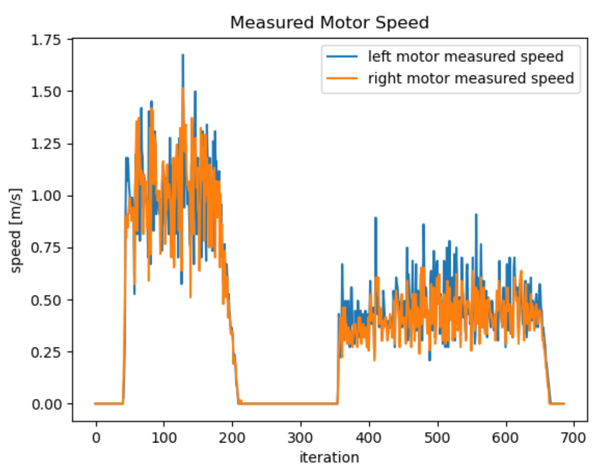

Figure 7: Setting the speed to 1m/s and then 0.5m/s

The above plot is generated by first setting the speed to 1m/s and then 0.5m/s. The average speed of the motor is 0.97m/s and 0.49m/s, respectively. This plot confirmed that the motor controller is working as intended.

3.2.2 Steering Deviation Verification

To verify that the robot turns accordingly with different deviation angle inputs, we set the deviation angles to various values and observe the measured motor speeds. The deviation angle is defined as the angle between the center axis and the target direction. This angle is negative if the target is to the left of the robot, and is positive if the target is to the right of the robot.

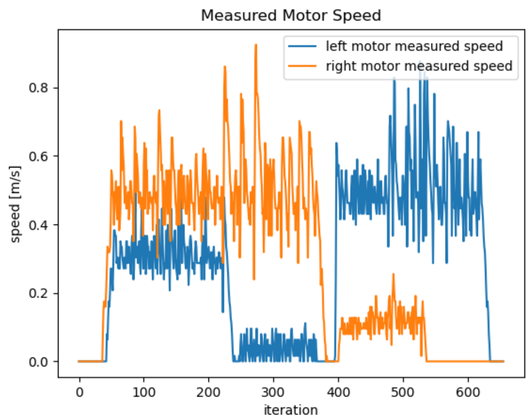

Figure 8: Setting deviation angle to various values

In the above verification, we set deviation angles to -17°, -45°, 33°, and 45° for each phase. The speed difference between the left and the right motor aligns with the deviation angle value, so we passed this verification.

3.3 Human-body Tracking Subsystem

3.3.1 Human-body identification Verification

To verify whether the human-body tracking subsystem can correctly identify its owner with an accuracy above 90%, we've conducted extensive experiments on examining its performance on various settings as shown in the table below. Note that accuracy is measured as

$$Accuracy \ = \ Corrected\ Identified\ Frame\ /\ Total\ processed\ Frame$$

over a tracking period of 30 seconds with a processing rate of 20 FPS, where the target human is always present in the frame. The reported accuracy is averaged over 5 trials.

Table 2: Accuracy of human-body identification algorithm under various setting

	Single Human	Multiple Humans
Non-distracting settings (5m)	97.89%	97.72%
Non-distracting settings (10m)	67.12%	63.77%
Irrelevant Objects (5m)	96.63%	97.02%

Strong Ambient Light (5m)	81.12%	79.54%
No Ambient Light (5m)	74.88%	66.51%

By examining the accuracy of single human versus multiple humans, we can see that the number of humans presents in the scene does not greatly affect the accuracy, which is an indication that our algorithm is robust at decoding the information if QR code is detected.

By examining Row 1 and Row 2 where only humans are present and only distance from the camera from humans varies, we can see the farther humans are away from the camera, the lower the accuracy rate is. This indicates that distance is an important factor in the performance of our system. One possible reason could be that the resolution of our camera is quite low. In addition, the effect of motion blur can be more significant when the target is further away and thus smaller in the camera view.

By examining Row 1 and Row 3 where comparisons are made to examine the effect of distracting objects like water bottles, chairs, etc. The accuracy remains the same for both scenarios, which is an indication that our algorithm has a very low false positive rate of detecting the QR code.

By examining Row 1, Row 4, and Row 5 where comparisons are made to examine the effect of ambient lights, we can see that our algorithm is very sensitive to ambient light intensity. The accuracy of detecting the owner drops significantly for extreme cases (strong/ no ambient lights). One possible reason could be that strong ambient lights could turn some black pixels white while weak ambient lights could turn some white pixels black in the camera view, causing the decoding part of the algorithm to fail to recognize the owner.

3.4 Safety Assurance Subsystem

3.4.1 Collision Avoidance Verification

To verify whether the safety assurance subsystem can avoid collisions with obstacles (higher than 1 inch) with a 90% safety distance (50cm*0.8=40cm) away from the target, we picked various test subjects and drove AutoLug toward the test subject using manual control under different speeds. The distance between the test subject and AutoLug, after it stopped, is recorded in the following table. Note that 0 cm means that a collision happened.

Table 3: Measured stopping distance of AutoLug from different test subjects when driven under different speeds

Test Subject/ Speed	0.25 m/s	0.5 m/s	1 m/s	1.5 m/s
Human	0.46 cm	0.28 cm	0 cm	0 cm
Wall	0.47 cm	0.45 cm	0.43 cm	0.41 cm
Glass Wall	0.46 cm	0.46 cm	0.43 cm	0.40 cm

AutoLug did pretty well at avoiding concrete obstacles like walls and glass walls, as it was able to keep at least 40 cm away from the test subject for all speeds tested. However, AutoLug did poorly at avoiding collisions with humans, as the stopping distance dropped below 40 cm starting at a speed of 0.5 m/s, and were not able to avoid collisions with humans thereafter. One possible reason could be that the radio wave from the ultrasonic sensor passed through the open space between the human feet, and therefore in the view of AutoLug there were no obstacles or humans ahead.

4. Costs

4.1 Parts

Table 4: Parts costs

Part	Manufacturer	Retail Cost ($)	Quantity	Summed Cost ($)
NVIDIA Jetson Nano Developer Kit	NVIDIA	149.0	1	149.0
LaView 32GB Micro SD Card	Laview	8.99	1	8.99
ML9-12 - 12 Volt 9 AH, F2 Terminal, Rechargeable SLA AGM Battery	MightyMAX Battery	22.99	2	45.98
CQRobot Ocean: 50:1 Metal DC Geared-Down Motor	CQRobot	33.99	2	67.98
2 inch Swivel Caster Wheel	HOLKIE	4	4	16
Total				**307.94**

4.2 Labor

We assume a typical ECE engineer can earn $50/hour, and we also estimate to work 20 hours per week on the project, throughout a span of 11 weeks.

Table 5: Labor costs

Salary ($/hour)	Number of Engineers	Total Work hours	Total Cost ($)
50	2	220	11,000

Grand Total = $11000 + $284.94 = $11284.94

5. Conclusion

5.1 Accomplishments

We were able to build a successful drivetrain with sufficient torque to drive AutoLug up to 2 m/s while having enough energy storage to sustain AutoLug to actively function for at least an hour.

We were able to build a successful control subsystem that is able to correctly steer the motors to drive AutoLug under desired speed and in a desired direction with a small tracking deviation.

We were able to complete a human identification subsystem that can achieve above 90% accuracy at identifying and tracking AutoLug's owner when multiple registered humans and distracting objects are present in the scene. With varying ambient light conditions and changing distances from the owner and AutoLug, the human identification subsystem is still able to maintain at least 60% accuracy.

We were able to complete a safety assurance subsystem that guaranteed to avoid collisions with large concrete objects regardless of the speed of AutoLug. In terms of humans, the safety assurance subsystem is able to stop AutoLug from colliding with humans when it's driving under 0.25 m/s.

5.2 Uncertainties

One uncertainty that we considered was the incorporation of the owner's re-locating functionality. The original intention was to, when AutoLug lost its owner, it should find or relocate its owner back to the camera frame by initiating a self-turning motion that rotates itself 360 degrees. However, as we progressed through the project, we realized the physical footprint of the robot doesn't allow us to perform this maneuver safely. It has a turning radius of 1m which might cause collisions with the surrounding when it spins around, so we ultimately decided to abandon this feature.

Another uncertainty was some strange failure cases for the QR code detection & decode method. As we were monitoring the camera feed from our App, sometimes we could observe clearly the QR code pattern from the frame but the identification subsystem failed to produce a bounding box. We made several assumptions and proposed various solutions, including changing the camera mount point and performing image transformations to restore possible distortion before feeding the frame for identification task. However, none of them turned out to be effective. We worked around this problem via having an additional object tracking algorithm as an alternative method when the QR code pipeline failed (as detained in Section 2.3.4).

5.3 Ethical Considerations

As progress through the project, we will firmly adhere to the Code of Ethics described in IEEE and ACM. We'll ensure a fair distribution of workload and a healthy working environment free of discrimination and racism according to the IEEE Code of Ethics II [6]. Equal rights and mutual respect will be valued the most as we are working on the project. In addition, we'll also respect and appreciate all external helps we receive. We will seriously consider and sincerely appreciate any advice given by course TAs and professors. We'll also carefully cite and give credit to all the external works done by others that have helped us along the way.

Regarding the safety of the project, one potential hazard will be the usage of a lithium battery. Therefore, we'll perform all the safety protocols to prevent any hazardous events arising from the lithium battery [7], including but not limited to keeping the temperature of the battery within the safety range of 32 to 130 Fahrenheit, avoiding sudden and drastic movement of the battery carrier, etc. Moreover, since the project involves a moving component at the ground level, there exists a possibility that it could crash on humans unexpectedly. Therefore, we'll limit the max movement speed of the project to 4 mph to prevent injuries if collisions happen. We'll also put a high priority in the safety assurance subsystem to prevent such collisions from happening under any conditions.

5.4 Future Work

AutoLug can be improved upon further. Firstly, we can add Kalman filters to all sensor data which should improve the smoothness of PID controls. In addition, we would want to keep exploring few-shots object detection models that can track humans in the frame without the aid of a QR code. "Few-shots" means that we only need a handful of training images of the owner to train the model and detect the correct owner. We could also add a depth estimation algorithm to help AutoLug avoid obstacles better. Finally, we would like to reduce the robot size and also the turn radius, so that AutoLug can spin in place and relocate its owner on its own.

References

[1] mygita.com, 2023. [Online]. Available: https://mygita.com. [Accessed 8 Feb. 2023]

[2] "Functions of Airwheel SR5 Intelligent Suitcase." Airwheel Official, www.airwheel.net/home/function/sr5. [Accessed 9 Feb. 2023]

[3] "Coefficient of Rolling Friction" Ron Kurtus. Available: https://www.school-for-champions.com/science/friction_rolling_coefficient.htm#.ZFMswaDMIQ9 [Accessed 20 Feb. 2023]

[4] Gaitset: Cross-View Gait Recognition through Utilizing Gait As a Deep Set - IEEE Xplore. https://ieeexplore.ieee.org/document/9351667.

[5] "RG1290T2", Raion Power. Available: https://www.raiongroup.com/content/RG1290T2.pdf [Accessed 10 Apr. 2023]

[6] "IEEE Code of Ethics," IEEE. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed 8 Feb. 2023].

[7] Batteryuniversity.com, 'Safety Concerns with Li-ion Batteries – Battery University', 2023. [Online]. Available at: http://batteryuniversity.com/learn/article/safety_concerns_with_li_ion. [Accessed 8 Feb. 2023].

Appendix A Requirement and Verification Table

Table 6: Drivetrain subsystem requirement and verification table

Requirements	Verifications	Verification status (Y or N)
Torque provided by each motor must be at least (with a coefficient of friction of) 10.4 N*m to drive the entire machine forward under a full load of 20kg with speed above 2 mph but not exceeding the 4 mph.	We will hang weights using a string at the edge of the tire and then turn the motor on. We will keep adding weights and examine whether the motor is capable of lifting the weights up. If the motor failed to lift the weights up as we slowly increase the total weight, then the torque of the motor can be calculated as total_weights * wheel_diameter. The requirement is met if the calculated torque exceeds 10.4 N*m.	Y
When the battery is fully charged, the machine should be able to remain active for at least 30 min.	We will first fully charge the battery before the verification process. Then, we will turn both motors and Jeston Nano active. The requirement is met if the machine can remain active for more than 30 minutes.	Y

Table 7: Control subsystem requirement and verification table

Requirements	Verifications	Verification status (Y or N)
In the tracking mode, the machine should follow the trajectory stably, with oscillation within 5%.	To verify the oscillation requirement in the tracking mode, we will simulate the tracking scenario by asking the platform to track the owner along a straight line with tapes on both sides to indicate the maximum deviation allowed when tracking. The requirement is met if the platform is able to track the owner without leaving the region bounded by the tapes.	Y

Table 8: Human-body tracking subsystem requirement and verification table

Requirements	Verifications	Verification status (Y or N)

The identification subsystem should be able to achieve at least 90% accuracy on identifying the owner within the camera frame, with obstacles and other humans present under varying ambient light conditions and changing distances away from the owner.	We will perform the verification under 5 scenarios: a non-distracting environment with humans present at 5 m away, a non-distracting environment with humans present at 10 m away, an environment with distracting objects and humans present at 5 m away, an environment with strong ambient light with humans 5 m away, and an environment with no ambient light with humans 5 m away. Each scenario will be tested for 10 trials, with 5 trials having the owner as the only human and the other 5 trials having other humans (different QR codes) with the owner. The requirement is met if the platform is able to correctly identify the owner with more than 90% accuracy for all 4 scenarios.	N

Table 9: Safety assurance subsystem requirement and verification table

Requirements	Verifications	Verification status (Y or N)
The robot should immediately brake whenever an obstacle is within a safety distance of 1 m, and the distance between the robot when it's stopped and the obstacle should be more than 90% of the safety distance.	We will perform the verification by running 10 trials, each with different obstacles placed at arbitrary distances away from the platform before the test. During the test, we will drive the platform towards the obstacles at arbitrary orientation and following arbitrary trajectory. If the platform stopped before the collision, the distance between the obstacle and the platform will be measured using a meter stick. The requirement is met if the measured distance for all 10 trials is more than 40 cm (90% of the safety distance of 50 cm).	N