ME461: Final Report

Team 21: Project VLAD: Vehicular Laboratory Automation Device

Adam Bahlous-Boldi
Kyle Fieleke
Miguel Ianus-Valdivia
Vlad Pyltsov
Abin George

9 May, 2023

Contents

1 Introduction

The VLAD (Vehicular Laboratory Autonomous Device) is a system that aims to automate laboratory workflows, eliminating the need for constant human intervention and saving valuable time. The goal of this report is to provide a detailed description of the design, assembly, and testing of the VLAD system.

The project was the brainchild of Professor Keith Brown, who needed an efficient and flexible laboratory automation system for his own Laboratory. Professor Brown runs the KABlab, which designs and tests hierarchically structured soft materials. Part of their research is about developing testing systems to be able to systematically and autonomously analyze various material properties of developed structures and materials. However, current testing systems lack modularity and would be challenging to adapt to different laboratory spaces or different experiments. The VLAD aims to help tackle this, by allowing for spacial flexibility in setting up autonomous experimental systems.

2 Customer Interview Summary

On 28 September 2022, we met with Professor Keith Brown to conduct a customer interview. We talked for about an hour, during which he told us that he was looking for an easier way to automate laboratory workflows. Currently, most lab operations depend on people moving samples between instruments, a task that takes time and requires constant manpower. The goal of our project is to design and prototype a modular system that moves samples between instruments.

Professor Brown first provided us with spacial constraints. He wants the solution to have as small a footprint as is feasible such that it can work inside a given room without impinging on day-to-day lab operations. It should be able to move up to 6 inches in a vertical direction so that it can accommodate tables of multiple heights. Additionally, the sample itself has spacial constraints; our system must be capable of transporting up to a six-inch cube weighing 1kg.

Another constraint touched upon during our customer interview was the final cost of the device. Prof. Brown stipulated that ease of access is one of the fundamental goals of the project, and this includes its purchasing cost for the customer, which should not exceed 500 dollars.

Since modularity is one of the system requirements, one must be able to deal with a multitude of sample types coming from multiple different end effectors. Professor Brown told us to design a universal adapter such that people could make their own end effectors for their own applications of the device.

Once the problem statement and constraints were established, a discussion of potential solutions commenced. One solution we came up with was using a cart system on 8020 rail. This is a good solution because 8020 is widely accessible which would make setting it up easier and perhaps help promote the spread of the technology. However, with this solution comes the question of how to navigate a corner; a problem which will surely require more thought. While the rail-system seems to be the most immediately obvious solution, Prof. Brown is open to other creative solutions that achieve the goals within the provided constraints.

A final problem to consider is the user interface with the device. One way to think about the user interface is in terms of levels of abstraction, essentially creating levels of different coding interfaces where some levels are highly modifiable by the users while other levels are set in place. For example, the motion controls of the system about the testing stations will be entirely coded by us to ensure that the general functionality of the system will be consistent regardless of how the test stations are set up. On the other hand, some elements of the coding interface should be editable by the user such that the users can add code that is specific to their research needs and testing stations. This idea of abstraction from a software perspective is analogous to the idea of modularity in the mechanical design of the system. In addition to this, Professor Brown stated that the hardware element of connecting the peripherals should take into consideration that user may want to add a few peripheral devices, sensors, or actuators, that complements their research objective.

While discussing the software and electrical components of this project, the professor stated that an added feature, but not a requirement, for this project could be to create a clean user interface to enable the user to easily arrange their lab setup that is specific to their environment and test stations, which essentially would make it much easier to facilitate the setup of their modular experimentation lab. Another added feature would also be to send real-time data from the sensors or send the results from the experimentation to an online platform such that the user can view the progress and current status of their experiments. This way the researchers can always keep track of the experiments that are running at the lab and have the ability to stop the experimentation if need be.

2.1 Key Takeaways

- Main purpose of the system is to move samples to and from testing stations set up by the user
- The system should be completely modular.
  - The system on which a sample is traveling should be able to be adapted to any environment and testing requirements
  - The mechanism that holds a sample should either be universal or the user should be able to insert their own model that holds the sample
- The whole system should be relatively easy to build and inexpensive

- System should account for small changes in elevation since table heights or floors are not entirely level
- The system should be able to carry at most a mass of 1 kg
- Time taken to travel between stations should range between 1 to 5 minutes
- An online platform to send real-time experimental and kinematic data is desirable but not essential.

3 Benchmarking Report

The purpose of this benchmarking report is to evaluate the elements of this solution against existing similar technologies and against different solutions devised to solve the same problem.

3.1 BEAR Lab

Similar to our senior design project, the BEAR lab at BU, supervised by Professor Brown, utilizes the same principles of autonomous experimentation to conduct its research. The BEAR lab's setup consists of five 3D printers, a scale, a robot arm, and an Instron. The workflow of this system involves a Machine Learning algorithm that determines different structures to be printed on the 3D printers; on the completion of print, the robot arm will move the structure to the scale and then to the Instron. Finally, the Instron will crush the part in order to compute essential structure parameters such as toughness and critical stress; once the compression test is complete, the robot arm will remove the part from the Instron and drop it in a storage unit. This whole cycle is an iterative process where new parts are constructed from results of previous parts, which unfortunately takes thousands of iterations to finally obtain a structure with the desired qualities.

3.1.1 Advantages of the BEAR Lab system

This lab provides us with the best example of what an autonomous experimentation system should look like. Here are a few of the key advantages of the system at the BEAR lab that we would like to adopt to our senior design.

- Complete Autonomy: At the BEAR lab, there is absolutely no need for any human intervention during the experimentation. Meaning at any point during the experiment, the autonomous system does not require the researcher to provide input, instead, the system has the right sensors and actuators to make a decision by itself. For example, sometimes the 3D printer extrusions get clogged, and the part does not print successfully; in order to account for such cases, the robot arm is equipped with a camera that takes a picture of the print bed and runs it through a neural net to determine if a valid part is on the print bed. This level of autonomy is what we aim to achieve with our project.

- Remote Control of System: Sometimes the autonomous system must be stopped by the researcher, this could be for safety reasons if the system is malfunctioning, or for other reasons such as updating the ML algorithm. In addition to this, one aspect of having an autonomous experimentation system is that researchers do not always have to be present in the lab; they could be working from home if they wish to. To facilitate these aspects of autonomous experimentation systems, the BEAR lab uses a messaging service application called "Slack" that is used to communicate with the autonomous system. With Slack, the researchers can send commands to stop the system, restart the system, or even get a list of things the system is currently working on, all with just a simple message. Through this, the researchers can constantly examine the experimentation being conducted and control the autonomous system remotely. This element of remotely being able to view and control the system is a feature we aim to incorporate into our senior design project.

3.1.2 Disadvantages of the BEAR Lab system:

As good as the autonomous system at the BEAR lab is, there are some features of the system that may pose problems to researchers who want to adopt an autonomous experimentation system in their labs. These are some features we plan to either fix or avoid for our senior design project.

- Cost: One major downside of the autonomous system at the BEAR lab is that equipment used in the system is the top-of-the-line equipment and, thus, very expensive to buy. For example, the robot arm used at the BEAR is lab is from Universal Robots and cost about $ 20,000. This may not be viable for other research groups that do not have much funding. For this reason, we aim to create a new autonomous system that is not only very affordable, but also can be easily recreated with equipment that can undoubtedly be found in any higher institution. This may involve the use of 3D printers to print parts quickly and cheaply. Moreover, we aim to use 8020 aluminum extrusion bars as the base of our system, and these extrusion bars can be found in abundance in any engineering college.
- Limited Use Cases: Another downside of the autonomous system at the BEAR lab is that the system is great only for a particular experimental setup but may not be able beneficial for others. In the context of the BEAR lab, the system is ideal for finding new structures and testing for their structural properties. However, this system is not ideal for an experimental system that mixes two solutions together and tests the solution's chemical properties. In order to account for this, we aim to create a system that is universal to any experimental setup. Of course, we cannot take into consideration every different kind of experimentation setups, however, we aim to provide the necessary mechanical and electrical freedom to customize our autonomous experimentation system to meet a researcher's needs.

3.2 Automated Manufacturing Processes

Autonomy is an essential feature of many industries today, and one of the biggest industries that have almost fully adopted autonomy is the manufacturing industry. Today, many manufacturers ranging from the automobile industry to the food packaging industry, utilize some level of autonomy to fulfill their tasks. By researching this industry, we aim to learn the drawbacks and the advantages of the autonomous systems used in manufacturing; and attempt to adopt the beneficial features and avoid the drawbacks of these systems in our senior design project.

3.2.1 Advantages:

- High Throughput: Through the use of autonomous systems, manufacturers can increase the rate of output. In other words, the manufacturers can produce more of their product in a shorter time frame. This is a huge benefit, even in the context of research where more testing can be done in a shorter time frame. This is one feature we try to achieve in our senior design project by utilizing the right sensors and actuators such that the system runs smoothly and efficiently.
- Lower Error Rates: One huge benefit of employing autonomous systems in manufacturing is the increased levels of accuracy. Sensors that are used by autonomous systems tend to give fairly accurate results which are used by the autonomous systems to perceive the environment and conduct the best set of movements. Moreover, these sensors are essential for certain decision-making tasks by the autonomous systems. For our senior design project, we aim to use good sensors and a good control system to go along with it such that error rates of the autonomous systems are minimized.

3.2.2 Disadvantages

- Power Intensive: Typically, manufacturers that employ autonomous manufacturing utilize machinery that is quite power intensive, which means that over the course of a few months, these systems use up a lot of energy. This makes systems such as this highly undesirable since they create not only huge operating costs but are also very harmful to the environment. Fortunately, the power consumption issue does not pertain to our senior design project since we do not aim to use power-intensive machinery. Instead, we aim to use simple actuators and sensors that do not require a high source of power.

Another highly automated process developed in industry is car manufacturing. This is generally done using conveyor belts that bring the item/sample from one instrument to another, where machines separate from the belt work on it before it moves to the next station. This is a different framework in which we could work in that the machines that move the sample are not actually on the conveyor/track and are stationary until they operate on the sample. We chose to instead make space for a single end-effector that moves along with the conveyor/track. The other important distinction between our solution and this idea is that we are using a track instead of a conveyor belt. A conveyor belt would require much more infrastructure to set up and the materials would likely be less commonly available than 8020 rail, so conveyor belts would be less ideal for the application.

3.3 Freight Trains

The most common and well-known thing that moves along a rail is the train. A train generally moves along two steel tracks similar to construction I-beams that are at a controlled distance from each other. The tracks are low friction and often also provide electrical power and control signals through a third rail. The advantage of using two tracks instead of one is that it is much more stable for heavy train carts moving at high speeds, but this is unnecessary for our application because we will have light carts moving at low speeds. What is more important is that it is easy to assemble, commonly available, and much more precise than a massive train.

We are instead using 8020 rail because it fits our application better than massive I-beams. We have also opted to use some form of wireless bluetooth/wifi connection to control the cart rather than a hard-wired connection along the rail because that would require modifying the 8020 for use and could reduce the adoption of our solution due to the significant additional set up required. The same reasoning also applies to why we are using one rail instead of two.

3.3.1 Advantages of Freight Trains

- High Capacity: Freight trains are able to move massive loads long distances at a high speed. This has a much larger capacity than our project requires, but we can still learn from these systems.
- High Stability: Trains are also very stable because they use a pair of tracks and that allows them to have the high capacity mentioned above. Having two tracks is unnecessary for our application though because we do not need the same capacity.

3.3.2 Disadvantages of Freight Trains

- Lack of Incline Traversal capabilities: Trains are not able to go up inclines typically more than a few degrees. This is due to the sheer weight of a freight train; the engine typically does not have the power needed to go up steep inclines. Additionally, even if they did, the conical wheel design means that there is little friction between the wheels and the track, thus greatly limiting their ability to traverse inclines.
- Cumbersome: These systems are clearly far too large for our application, but we can still use the idea of moving a car along a rail in order to move things around in a lab faster than it has been done before.

4 System Requirements

- Modular system architecture
- Allow for the manipulation of a lab sample $6'' \times 6'' \times 6''$ in size and $1\,\mathrm{kg}$ in mass

- Bring the sample to any location within the lab, allowing for variations in height
- Minimal disruption to the day-to-day happenings of the lab.
- Quick and Intuitive setup
- (Semi) Autonomous
- Self-sustaining & low maintenance
- Ability to keep entities upright during manipulation
- Low cost to the laboratory (<500$)

5 General Solution Overview

Our solution, which achieves system requirements while adhering to constraints, is the VLAD. The VLAD consists of a cart system that runs along standard pieces of 1.5” 8020. The cart consists of two identical trucks on either side of a 2-level platform; the bottom platform hosts the electronics and battery, while the top platform is able to tilt in 1 axis, powered by a stepper motor, in order to remain normal to gravity. One truck hosts a DC Motor to propel the system, while the other hosts an encoder to record the displacement of the system.

Each truck consists of 5 wheel bearings that roll along the top and side profiles of the 80-20 rails (2 pairs on the side, one on the top). The side bearings resist tilting torques and guide the cart along the rail, while the top bearing primarily supports downward loads. A rubber belt-driven wheel attached to the DC motor is mounted to the back of the rear truck, while a rotary encoder is analogously attached to the front truck. Eccentric nuts are used to secure the side bearings to the track such that one can ensure constant and well-tensioned contact between the wheel surfaces and the track. Trucks are attached to the lower platform on each side by a 2-axis gimbal, to allow kinematic independence between both sides of the system, allowing it to go around tight corners and up steep inclines up to 20 degrees. When going around

Since only 3 out of the 4 80-20 side surfaces are occupied by the cart, the 4th can be used for mounting the rail itself.

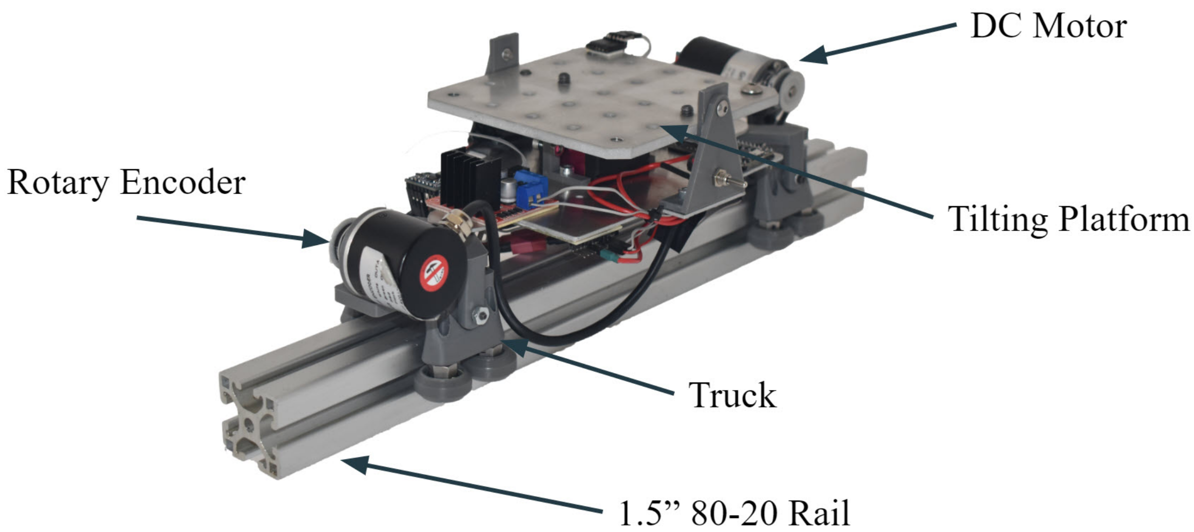


Figure 1: An overview of the VLAD

6 Track Design

6.1 Design Requirements

The track is the base on which the rest of the project sits and therefore had to fulfill many requirements that allow the device to perform its many functions. These included:

- **Three sides for the trucks to attach onto:** The track must have three sides on which the wheels rest and a fourth side that can be used for connectors.
- **Three-dimensional travel:** The track must have pieces that turn left, right, up, and down that allow the device to travel in any direction and around a lab environment freely.
- **Connectors:** The connections between the track pieces must both keep the pieces firmly together and counter-act torque such that they stay co-planar and the device may travel over them without getting snagged on a corner.
- **Multiple options for brackets:** There must be multiple mounting options that keep the track fixed in space regardless of the setting in which it is used. This means making firm brackets that can fix both to tables and walls.
- **Manufacturability:** All of the above pieces must be 3D-printable to allow individual laboratories to make their own.

6.2 Design Evolution

6.2.1 Track Design

The rest of the project depends on the track to operate, so the design of the track was mostly completed during the first semester. We knew from the beginning that we would use T-nuts to keep the track pieces together, but we had not yet figured out how to best counter-act torque. Version 1 is shown below.

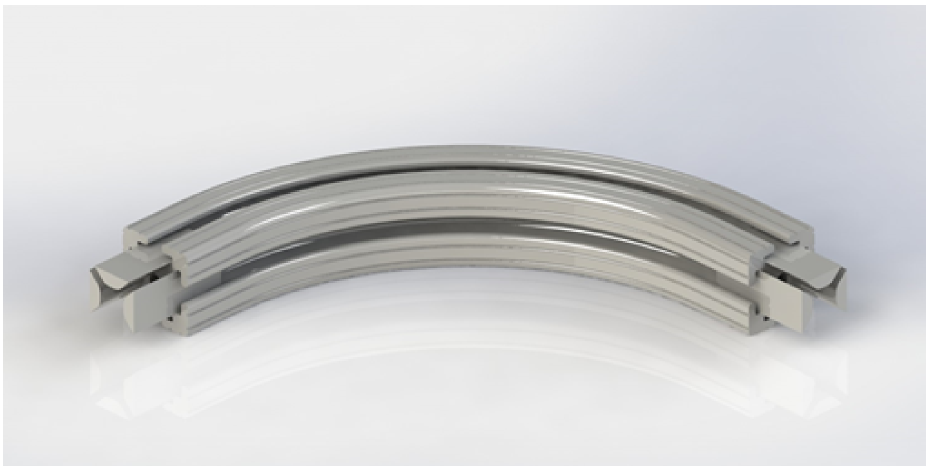


Figure 2: V1 of the track

This version was not easily 3D-printable and did not leave enough space for the wheels to roll across the connectors, so we decided to simplify the geometry of the track profile as shown in the next subsection. We also realized we could make a second connection point using a dowel pin instead of the extrusions shown in the above figure. This is also shown in the next subsection.

6.2.2 Table Bracket Design

The first version of the table bracket used a clamp with a single hole and did not take usability into account.

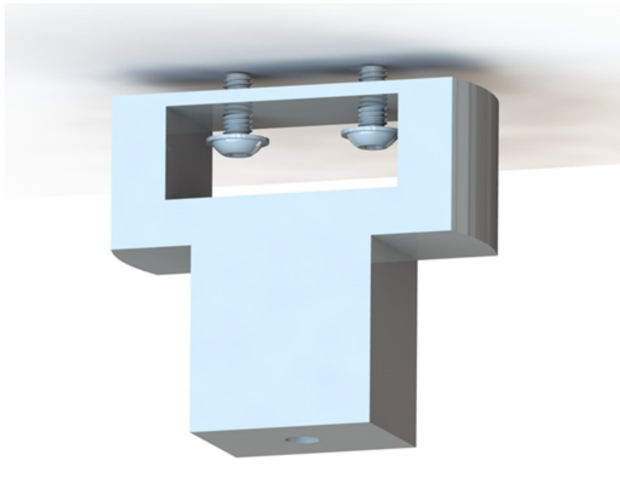


Figure 3: V1 of the table bracket

The above design never made it to the manufacturing stages because it was not 3D-printable, was bulkier than it needed to be, and did not leave enough space to allow for an Allen key to be inserted to tighten the screws. Instead, we opted for a redesign that addresses these issues as is shown in the next subsection.

6.2.3 Wall Bracket Design

The first design of the wall bracket followed the design of the first table bracket. It also had many of the same problems.

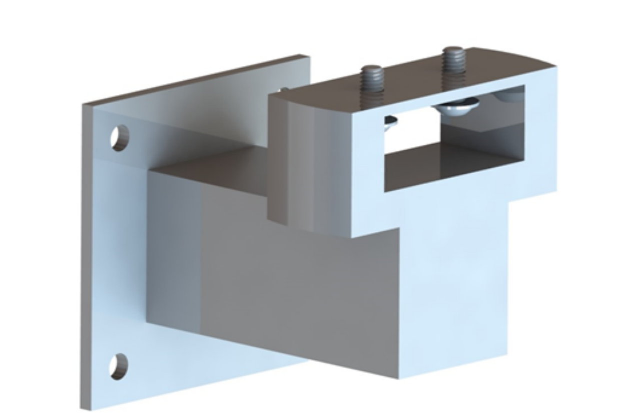
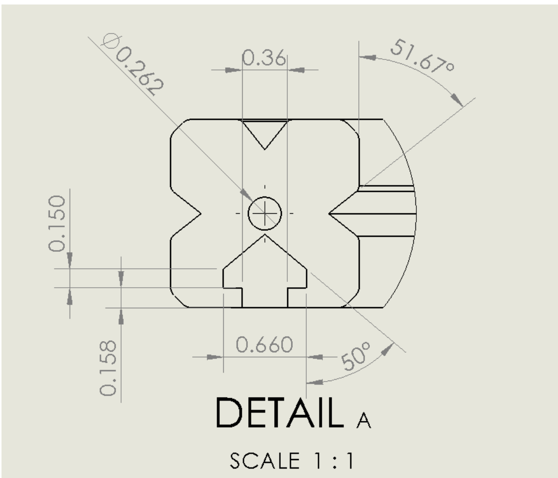


Figure 4: V1 of the wall bracket

This design also likely would not have been able to support the weight of the track and the device because it had sharp corners that when printed would have created places for stress to concentrate. This would have caused cracks and eventually failure when a load was placed on the bracket. A redesign led to the version shown in the next subsection.

6.3 Final Design

The track system is based on 15-series 80-20 aluminum extrusion but it also includes turns that allow the linear drive system to traverse a three-dimensional space. These turn pieces fit the profile of the 80-20 and have features that allow them to be connected together easily but have some modifications to make them 3D-printable. This modified profile is shown below.



(a) The track profile

(b) A render of the track

Figure 5: The printed track profile and the track assembled

Three sides of the track have an angled divot that is easily printable but still constrains the wheels of the truck enough to keep them on the track. The fourth (bottom) side has a geometry similar to the original 80-20 such that a T-nut and dowel pin can be used to connect both to other track sections and to brackets that fix the system in space relative to either a table or a wall. This geometry in practice is shown below.

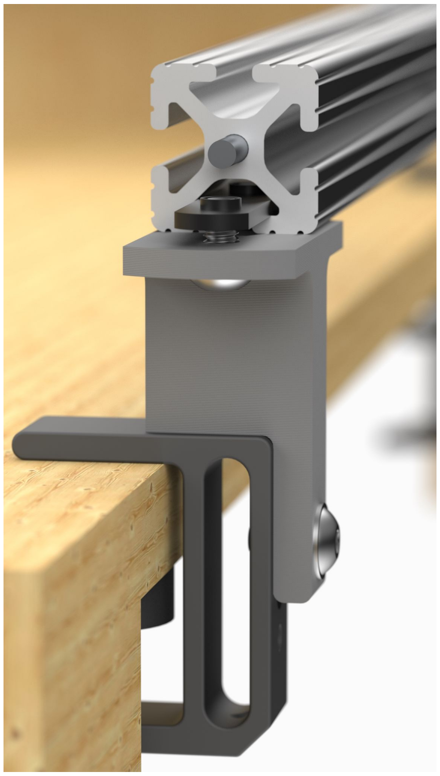


Figure 6: Render showing the connection mechanism and table mount

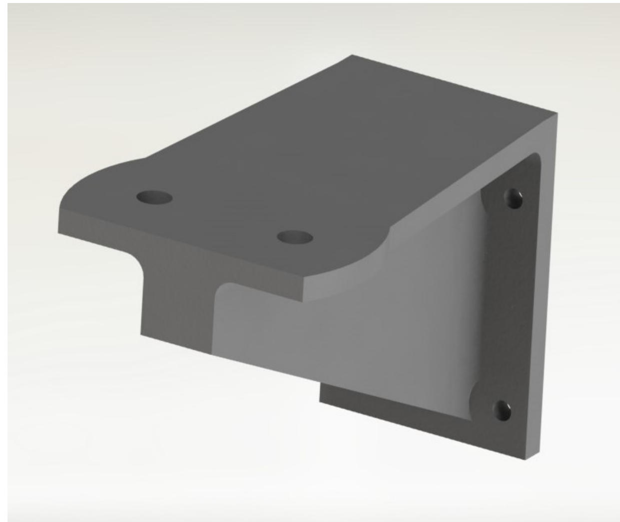


Figure 7: Render of the wall mount

Basing the track system on 80-20 allows it to be more easily adopted in university laboratories because many already have extra 80-20 lying around and are familiar with 80-20 constructions. The connectors and brackets are also modular and intuitive which allows for easy assembly in a near-unlimited number of configurations to conform to any lab.

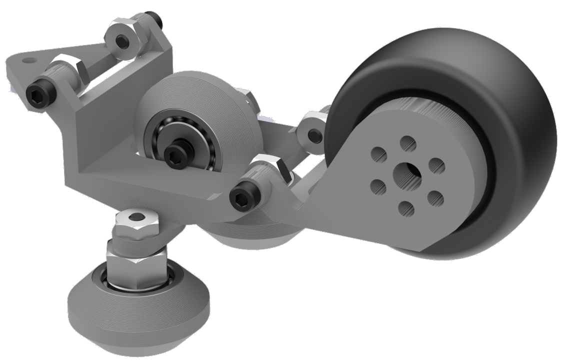
7 Truck Design

In the context of this system, we have elected to refer to the portion which interfaces the tilting platform and the track as the truck. In our case, the truck design differs from those typically used in *linear* extrusion-based motion systems, as it must be able to navigate any track geometry the user desires. Additionally, unlike most linear systems, we are not able to use a belt drive system, again due to the nature of our track. It would be challenging to maintain tension on the belt without adding tremendous cost or friction to the system. Thus, we have devised the following set of sub-requirements for our truck design:

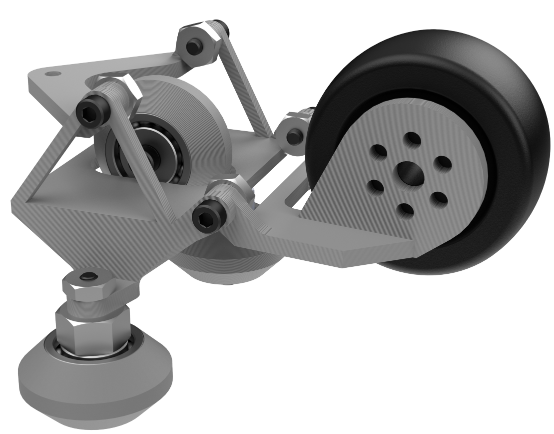
7.1 Design requirements

- **Independent Pitch and Yaw:** For the system to be able to navigate the lateral and vertical turns present within our track, both trucks must be able to independently pitch and yaw with respect to the platform to which they are connected. Otherwise, the entire system would come to a grinding halt the instant it encounters a turn.

- **Tight Grip:** While it is typical that an extrusion-based slider mechanism must maintain a firm grip on the track, it is especially imperative in our case, both to support the load and to provide enough normal force to the drive wheel to be able to drive the system. Without this normal force, there would be insufficient friction between the extrusion and the drive wheel to produce relative motion.
- **Drive/Measurement Wheel:** Along with the previous requirement, each truck must also have wheels that possess a great enough coefficient of friction with both the aluminum extrusion surface and the plastic surface of our custom turn pieces. Without this, the wheels are unable to effectively transfer power from the motor to create relative motion, or in the encoder's case, will cause slip leading to inaccuracy in the measurements.
- **Manufacturability:** Because we intend to open source this system, we want to ensure that all components the user would need to set the system up in their lab are easy to produce and procure. Thus, we placed a strong focus on making all parts as easy as possible to produce for the users, with a particular focus on Fused Deposition Modeling (FDM) 3D-Printing.



(a) V1.0



(b) V2.0

Figure 8: Truck designs, first versions

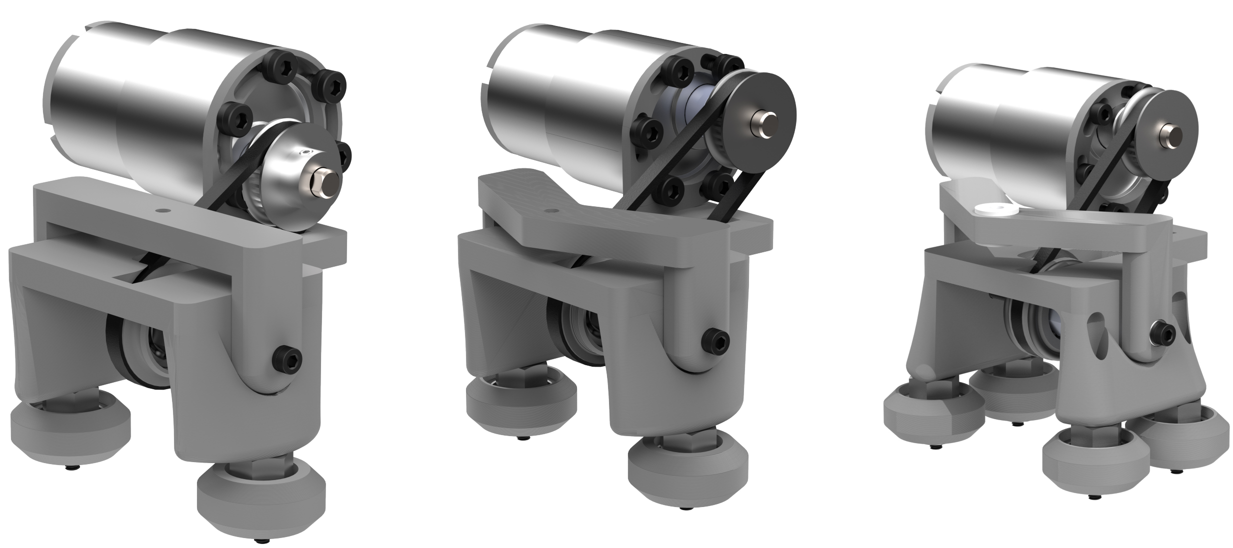
7.2 Design Evolution

Keeping all of these requirements in mind, we began our initial truck design, seeking to meet as many criteria as possible in the first design iteration, but focusing mainly on exploring the number and positioning of joints we need. As seen in Figure 8a, the first iteration of the truck design features three passive rollers, two on the side and one on top, which each contain bearings. Additionally, each passive roller is mounted on an eccentric nut, to allow for the position of the roller to be fine to make just enough contact with the rail. On the right-hand side, we can see a large rubber drive wheel, to which the motor would be mounted directly, resting on a separate platform. This platform is pinned to the main platform and would receive a mousetrap spring on the pin joint to provide the normal force between the drive wheel and extrusion. On the far left-hand side, there is the attachment point to the tilting platform. This joint was carefully designed to allow the tilting platform to both pitch and yaw with respect to the truck.

The only major change from the first to the second version is moving the last pin in the sequence that attaches to the remainder of the platform vertically in line with the top roller, to avoid creating strange moments, seen in Figure 8b. Otherwise, the truck remained largely unchanged in this iteration.

7.3 Final Design

In the later iterations of the truck seen in Figure 9, we put a strong focus on manufacturability which was not present with the prior design iterations. The designs in the prior iterations, unless manufactured from metal, would not be strong enough to sustain the loads demanded by the system, and would be somewhat of a nuisance to machine. The final versions of the trucks are easily 3D-printable on any FDM 3D printer by limiting the number of overhangs when all parts are oriented upside-down on the build plate. Additionally, the last design in the progression shown in Figure 9 features dual side rollers, which provide additional grip to the track and the new drive system. The new drive system leverages the top roller by cutting out the center to provide good grip for a timing belt, and channels on the outside of the roller make it possible to insert O-rings to provide friction on the aluminum surface.



(a) V3.1 (b) V3.3 (c) V3.5

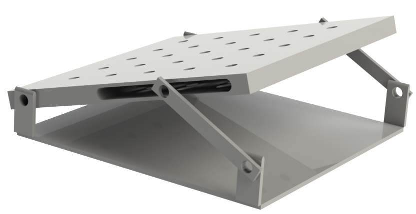
Figure 9: Truck designs, Final Versions

8 Tilting Platform Design

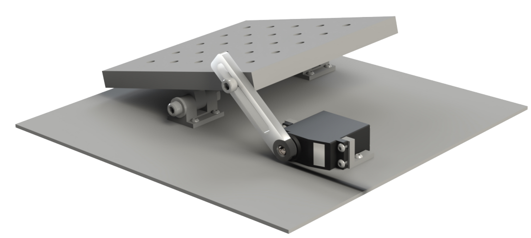
8.1 Design Requirements

Our design constraints were organized after the interview with our client. The list was created as follows. The tilting platform system:

- Must be able to achieve a 20-degree tilt in either direction.
- Must be able to accommodate objects $6'' \times 6'' \times 6''$ in size.
- Must be able to hold a sample of mass of 1 kg.
- Must be able to keep samples level throughout inclination changes.
- Must be able to accommodate all electronic components.
- Must connect to trucks with as little friction as possible at pivots.
- Should have a modular design without sacrificing function.



(a) V1.0



(b) V2.0

Figure 10: Tilting platform, early designs

8.2 Design Evolution

We started off by exploring the idea of driven linkages. It was inspired by exercises completed in the earlier Product Design class. The design first design iteration can be seen in Figure 10a

The idea was to drive linkages such that they tilt the platform. The linkages would be connected through the slot cut in the platform itself. To satisfy the modularity constraint, the platform was designed to have standard optical breadboard holes of 1/4”-20. The platform would be connected to the chassis through hinges. However, the design has multiple drawbacks. First of all, it does not rotate both directions. Secondly, it does not allow accommodation of electronic components. Thirdly, driving this design of linkages can be problematic and complicated. Lastly, the connection through slot cut can obstruct a lot of platform holes which undermines the whole purpose of the holes itself.

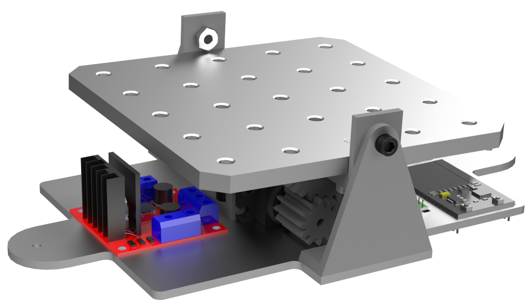
For the next design iteration, we proceeded to expand the idea of linkages by leaving only one linkage such that it can be driven by the servo motor. On top of that, the hinges were moved under the platform such that the platform could rotate both directions around its central axis. The design looked like this:

While certain improvements were made, this design contains a few drawbacks. The driven bar linkage would experience a lot of torque making the system fragile and unreliable. On top of that, the system itself would be bulky as the distance between platform and servo motor is relatively big. After all, driven linkage could also produce some sort of backslash or drift resulting in inaccurate tilting.

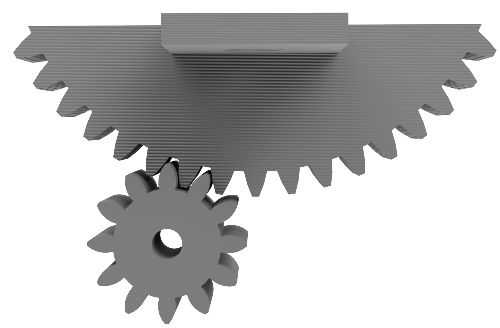
Accounting for various design drawbacks, we proceeded with creating our final design.

8.3 Final Design

Out of these goals and constraints, we came up with a 2-level design. A tilting optical breadboard-style platform pivots about points on two triangular legs connecting to a lower chassis. The chassis underneath the tilting platform is designed to accommodate all of the electronic components without risking collisions during the movement of the tilting platform. The design looks like this, and is driven by mechanical gears:



(a) Tilting platform



(b) Gear system

Figure 11: Tilting platform, final design

A sector gear is attached to the platform, a smaller gear is driven by the servo motor. Our major challenge was to figure out the geometry for this gear system. Precise gear ratio and geometry allow high precision and highly controlled rotation of the platform. The chassis additionally has 2 pin joints (1 on each end), which are equipped with Delrin washers to reduce friction between the chassis and truck, while still maintaining stability

8.4 Geometry

2 geometric constraints are placed on the kinematics of the Tilting platform. Firstly, the goal of achieving a 20 deg tilt places a hard lower limit on the height the platform must

be above the chassis. This is a function of the width of the platform w and the distance d between the pivot and the platform (since these are not inline). The height of the pivot can thus be written as:

$$h = d\cos\theta_1 + \frac{w}{2}\sin\theta_1 \tag{1}$$

Next, to maximize the effectiveness and resolution of the servo, the full travel of the sector gear should be set to the full range of motion of the servo (that is, 180 degrees). However, since the driven gear is offset by an amount d_{off}, the range of motion is slightly less. Specifically, we can compute the total sector angle θ_s as follows:

$$\theta_s = 2\theta_1 + \tan^{-1}\left(\frac{d_{\text{off}}}{h - R_2}\right) \tag{2}$$

where R_2 is the radius of the driven gear. Finally, we can write a few other geometric relations by inspection and the gear motion constraint. These relations can then be solved numerically in Matlab to yield the optimum, radii R_1 and R_2

$$R_1\theta_s = R_2\theta_2 \tag{3}$$

$$(R_1 + R_2)^2 = d_{\text{off}}^2 + (h - R_2)^2 \tag{4}$$

letting $\theta_1 = 20°, \theta_2 = 180°, d_{\text{off}} = 0.45, w = 5\,\text{in}, d_1 = 0.35$ yields:

$$R_1 = 1.1\,\text{in} \tag{5}$$

$$R_2 = 0.238\,\text{in} \tag{6}$$

$$\text{ratio} = \frac{R_1}{R_2} = 4.6 \tag{7}$$

The actual gear ratio we ended up going with is 3.92 rather than the optimum 4.6. This was due to adding an extra buffer to ensure there are no collisions with the electronics under the tilting platform, and extra tolerance added to mitigate gear backlash causing a hysteresis in the tilting performance.

9 Electronics: Hardware

9.1 Sensors and Actuators

From an electrical standpoint, the first step we took was to figure out what sensors and actuators were required for this project. To complete this task, we created a function-means table (table 1).

Table 1: Function Means Table

	Possible Means			
Determine the position of the cart	Quadrature Encoder	Optical encoder	Switch based encoder	DC motor with encoder
Position of different testing sites	Infrared Sensor	Camera (CV)		
Angular position of tray	Accelerometer	Tilting Sensor		
Stabilization of the tray	Hydraulics	Stepper motors	Servo motors	
Actuate cart along the rails	DC motor	Belt and pulley system	Linear actuators	
Microcontroller	ESP 32	Arduino UNO		

9.1.1 Position of the Cart

It is crucial that the position of the cart is known throughout the experimentation routine in order for the system to arrive accurately at a testing site and conduct the relevant actions. We came up with different solutions, the best one being the hall effect encoder. The hall effect encoder has a shaft similar to that in a DC motor which can rotate causing a change in a magnetic field that is converted to a signal that measures the total angular rotation of the shaft. From this angular rotation, the total distance traveled can be determined by multiplying the angular rotation with the radius of the wheel.
Other solutions to determining the position of the cart included the use of an optical encoder, a switch-based encoder, and a DC motor with an encoder attached to it. The optical encoder uses optical pulses that are passed through a slit to determine the change in angular rotation. These optical encoders tend to be less accurate than the hall effect encoder. Another solution was the use of switch-based encoders, which are similar to a potentiometer in the sense that the rotation of the shaft changes the voltages. Using the change in voltage, the total angular rotation can be computed. However, the major downside of using a rotary-based encoder is the high torque required to rotate the shaft, which means that the rotary encoder will be an added layer of resistance to our system. Finally, we thought about using an encoder that is inbuilt within a DC motor, however, if the wheel slips then the encoder will register that the truck moved when it actually did not. For these reasons, the Hall Effect encoder was deemed the best option for our application.

Figure 12: Hall effect encoder

9.1.2 Location of Testing Site

One of the main objectives of our project is to have an almost autonomous system. For this reason, it was important to us that our system could identify testing sites along the track with little to no human input. One solution we thought of was to use a camera and implement a computer vision algorithm that can detect a marker that is placed near the testing site. However, in order to accurately and quickly identify a testing site while moving, not only would we require an expensive camera but we would also require an onboard computer that can handle intensive calculations. This is to say that the use of a camera will make our system significantly more expensive and complicated.
A cheaper and simpler solution is the use of infrared sensors to detect the location of the testing sites. For this solution to work, we will require the users of our system to place a dark sticker on the track of the 8020 extrusion bars at every testing site. The goal is to have an infrared sensor (IR) that is always pointing toward the track of the extrusion bars. The infrared sensor works by having a transmitter that sends infrared signals and a receiver that receives the reflected signal. While the truck is away from a testing site, the signal received by the IR sensor will be more intense than that received when at a testing site. This is because aluminum is more reflective than the dark stickers that would be placed at every testing site. For this reason, we decided on using infrared sensors as our means of determining the location of the testing sites.

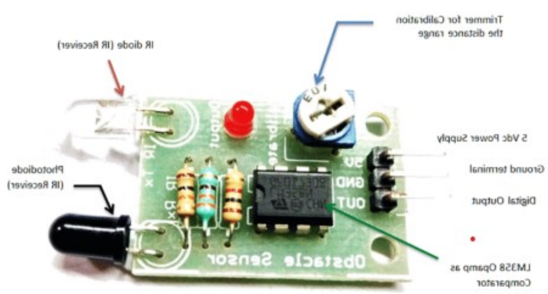


Figure 13: Infrared Sensor

9.1.3 Angular Position of the Tilting Platform

We previously discussed the need for a tilting platform to ensure that the payload does not fall off when moving up or down inclines. To accomplish the task, we need a sensor that can measure the change in the slope of the track with respect to the floor. We found that an accelerometer and a gyro sensor would accomplish this task. Both these sensors work on the same principle of using an inertial measurement unit (IMU). However, the only difference between these two sensors is that the gyro sensor measures angular velocity about the 3-axis, while the accelerometer uses linear acceleration along the 3-axis. Since the goal was to measure the angular change of the system from the ground, the gyro sensor was deemed to be a better fit for our application.

Figure 14: Tilting Sensor

9.1.4 Stabilization of the Tilting Platform

Using the above-discussed gyro sensor, we can determine the angle of elevation of the track. To counteract this and keep the tilting platform horizontal, we will need an actuator to rotate the platform. We discussed many different ideas that can be implemented, however, only one seems to work perfectly for our application which is the servo motor. The servo motor can quickly change angles to keep the platform horizontal and is fairly lightweight. The other ideas we discussed include stepper motors and hydraulics, which are heavier and slower than servo motors. For this reason, the servo motor was chosen as our means to keep the tilting platform constantly horizontal.

Figure 15: Servo motor

9.1.5 Actuation of the Cart

One of the most important peripherals of our system is the actuator used to move the cart along the extrusion tracks. We brainstormed on different systems that can be used such as the belt-pulley system and linear actuators. However, these actuation methods make the setup of our system more difficult for the user. Moreover, implementing these actuation methods also makes our system less modular. The best solution we thought of was a DC motor that can be connected onboard the cart. This makes our system a lot simpler and quicker to set up.

Figure 16: DC motor

9.1.6 Microcontroller

Finally, we need a microcontroller to collect data from the sensors and actuate the system accordingly. There were two main contenders: the ESP32 and the Arduino UNO. The ESP32 (2-core 32-bit processor) can connect to more connection lines and has a faster computing processor than the Arduino UNO (1-core 16-bit processor). In addition to this, the ESP32 is much cheaper than the Arduino UNO. For these reasons, we decided to use the ESP32 as the microcontroller for our project.

Figure 17: ESP32 Microcontroller

Once we decided on the electronics for our system, we made a preliminary sketch of how the electronics will connect and communicate with the microcontroller. The infrared sensor and the hall effect sensor send data to the microcontroller using digital output signals, the gyro sensor uses an I2C communication protocol to send data to the controller, and the servo and the DC motor with a driver are controlled using power width modulation (PWM). In addition to this, it must be noted that we decided to use two microcontrollers to run separate processes. The main motivation for this was to have fewer dependencies between the two controller systems, which will lead to a better-performing and more efficient controller. In addition to this, the ESP32 microcontroller will have more available ports for the users of our system to add their own peripherals for the application.

10 Controls

10.1 Data Flow

10.1.1 Tilting Platform

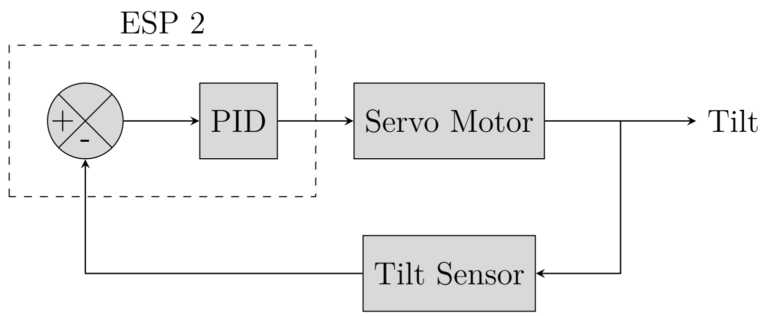


Figure 18: Data flow for the tilting platform

Figure 18 shows the data flow between the peripherals in the tilting platform system. In essence, the tilting sensor provides information on how much the platform tilted and passes it onto the Proportional-Integration-Derivative (PID) controller. The PID controller computes the necessary change to the servo position and sets the servo to that position. This process uses a feedback loop which ensures that the error of the system will stay minimal. Additionally, the tilt sensor was placed on the platform itself, this way the PID controller will ensure that the platform is level regardless of any unbalanced or unaccounted external force applied to the platform.

10.1.2 Truck

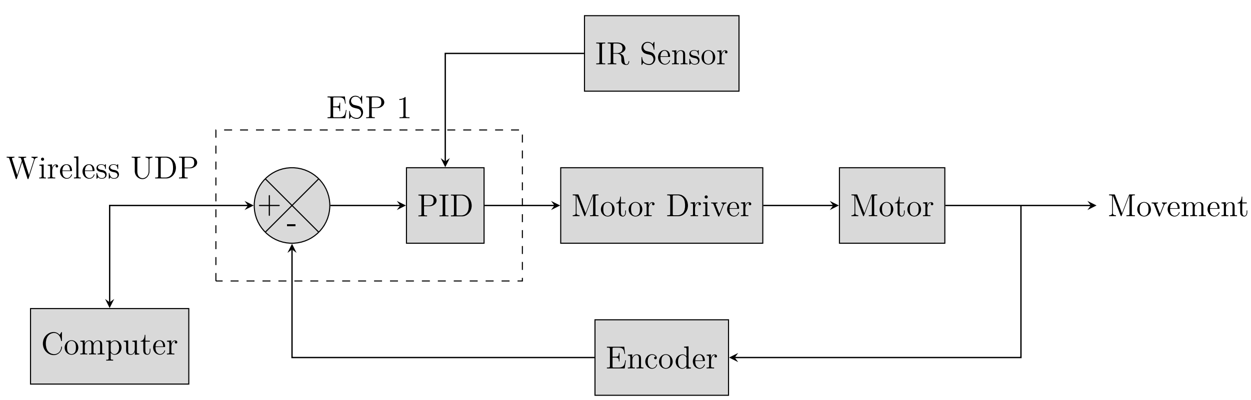


Figure 19: Data flow for the truck

Figure 19 shows how data flows between the peripherals of the truck system. In short, the IR sensor will check whether the truck reached a station. Each station will be marked with black tape and the IR sensor will read a significantly different value for the black tape than either the aluminum extrusions or the 3D printed pieces. The motor moves the truck forward and backward via the motor driver. The ESP32 sends a PWM signal to the motor driver, which converts and sets an appropriate voltage for the DC motor. The encoder sends two signals to the controller which is then decoded to determine how much the encoder rotated. The onboard controller is only responsible for moving the truck a certain distance forward or backward. However, the user's local computer can dictate to which station or how much distance the truck should move. To wirelessly communicate between the computer and the controller, the User Datagram Protocol (UDP) is used which runs over WiFi. Finally, a PID controller is used again to move the truck in a smooth manner, such that the truck will gradually come to a halt at its final destination.

10.2 Tilting Platform

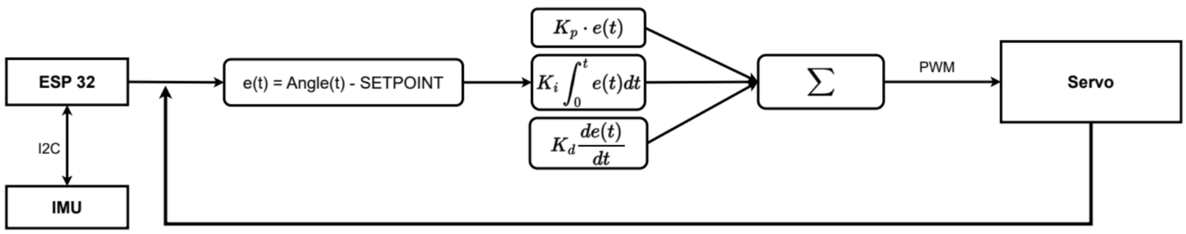


Figure 20: Controls Flow Chart for Tilting Platform

Figure 20 shows the basic flowchart of the tilting platform. This figure shows the basic controls algorithm used for the tilting platform. However, there were a few challenges we faced along the way.

Tilt Sensor

Initially, a standard IMU was used to compute the tilting angle of the platform. However, a standard IMU does not output an angle, but rather an angular velocity. Thus, to determine the angle we had to integrate the angular velocity with respect to time. Unfortunately, digital data is never continuous but discrete. The following computation was done to determine the angle from a standard IMU.

$$\alpha = \sum \left(\frac{\omega_t + \omega_{t-1}}{2} \cdot dt \right) \tag{8}$$

α: Angle of the tilting platform from the horizontal.
ω: Angular velocity reading from the IMU.
dt: Time interval between each IMU reading.

Due to the summation of discrete values, as time increased the angle calculated became more and more erroneous, i.e. the tilting sensor had a large drift.

In order to fix this, we purchased a tilting sensor (WIT 901) that computed Euler angles onboard the sensor using a combination of acceleration values, gyroscope values, and magnetic field values. Moreover, we positioned the tilting sensor such that the pole of the sensor (global axis of the sensor) was aligned with the local axis of the sensor. This way we could take advantage of not having to compute the tilting platform angle from the Euler angles using a computationally intensive transformation matrix. Instead, we could read the raw values from the tilting sensor which provided the tilt angle of the platform.

Servo

During our integration phase, we noticed that the servo does not move as quickly as the tilting sensor refresh rate. For example, let's say that the servo motor moves from 0-20 degrees. Before the servo reaches the final position, the tilting sensor will read the angle assuming that the servo reached the 20-degree mark. This leads to misalignment in the data causing the tilting platform to react chaotically.

One solution was to simply lower the refresh rate of reading the tilting sensor values. Although this works, it will significantly slow down our system, i.e. the tilting platform will react very slowly to any changes in the slope.

The solution we opted for was to create a parallel task that solely moves the servo to the final position in increments rather than the absolute position. This way the servo is fast enough to reach its desired position and the data from the tilting sensor will no longer be misaligned. The algorithms below will show an overview of how the PID works (Algorithm 1) and how the servo task is running in parallel (Algorithm 2).

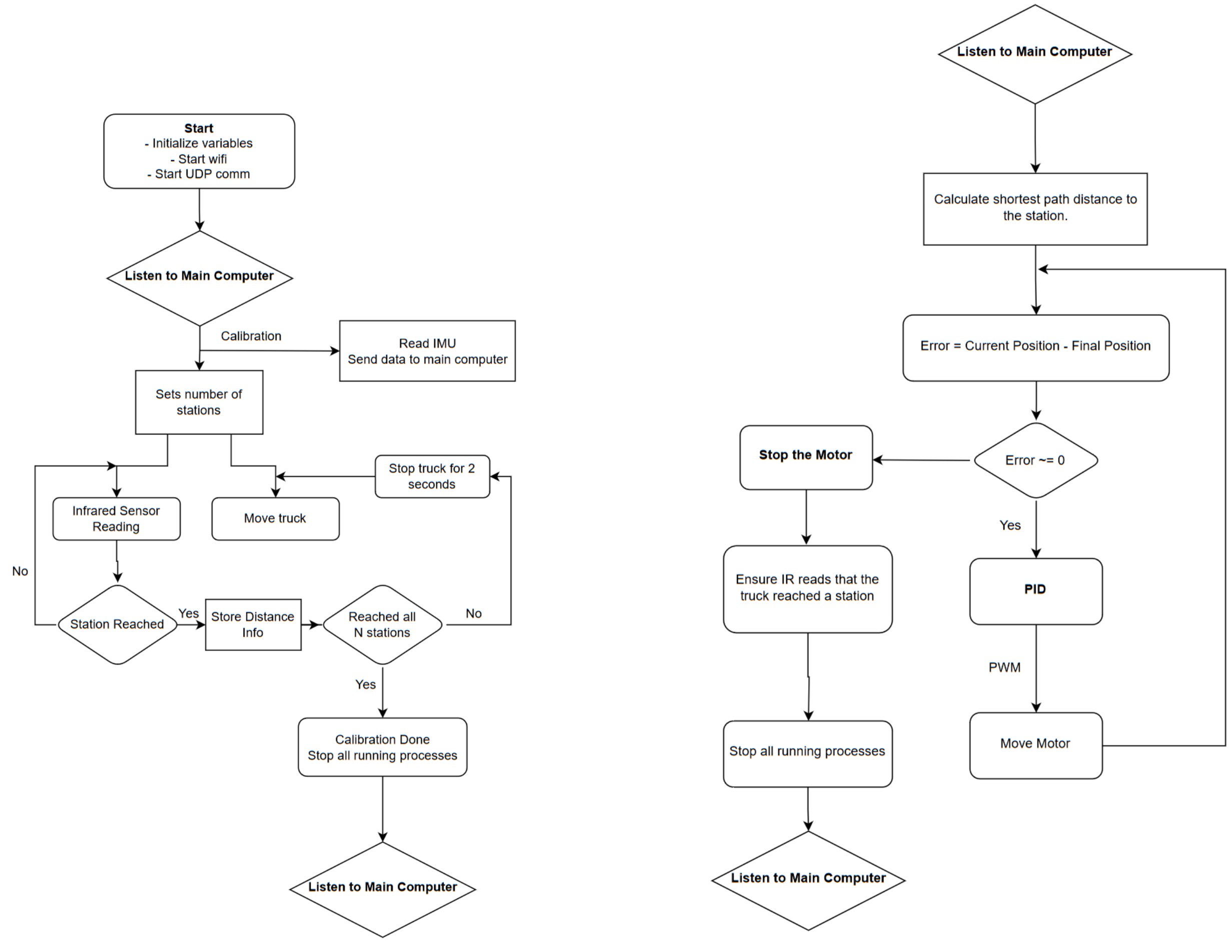
Algorithm 1 PID Task

```
SET_POINT = 0
prevError = 0
global angleChange = 0
while TRUE do
    error = TILTING_SENSOR_VAL - SET_POINT
    derror_dt = (error - prevError )/ dt
    serror_dt += ((error + prevError) · dt) / 2
    angleChange = Kp·error+ Kd·derror_dt + Ki·serror_dt
    timeDelay(50 ms)
end while
```

Algorithm 2 Servo Task

```
global angleChange
servoAngle = 0
while TRUE do
    servoAngle += (angleChange) · 0.2
    timeDelay(10ms)
end while
```

10.3 Truck



(a) Calibration Routine

(b) Move to Station

Figure 21: Controls flowchart for Truck

Figure 21 shows the control flow chart for the truck system. The only challenge we faced with the truck system was getting the IR sensor to consistently output the same values.

IR

While testing the IR sensor, we realized that the IR had to remain a fixed distance from the track. Any changes from this could lead to inconsistent results. This was mechanically fixed by creating an IR housing on the truck. Once this was done, we read the IR output for different materials and colors and finally chose black electric tape.

The output of the IR sensor ranges from 0-4095; the IR reading for the aluminum extrusion is 1500 and for the 3D printer parts is 2500. Over the black electric tape, the IR reading was consistently 4095. Algorithm 3 shows the pseudocode of how the IR detects a station.

Algorithm 3 IR Task

```
IR_THRESHOLD = 4000
servoAngle = 0
while TRUE do
    if IR_VAL > IR_THRESHOLD then
        STOP TRUCK, REACHED A STATION
    end if
    timeDelay(10ms)
end while
```

11 GUI

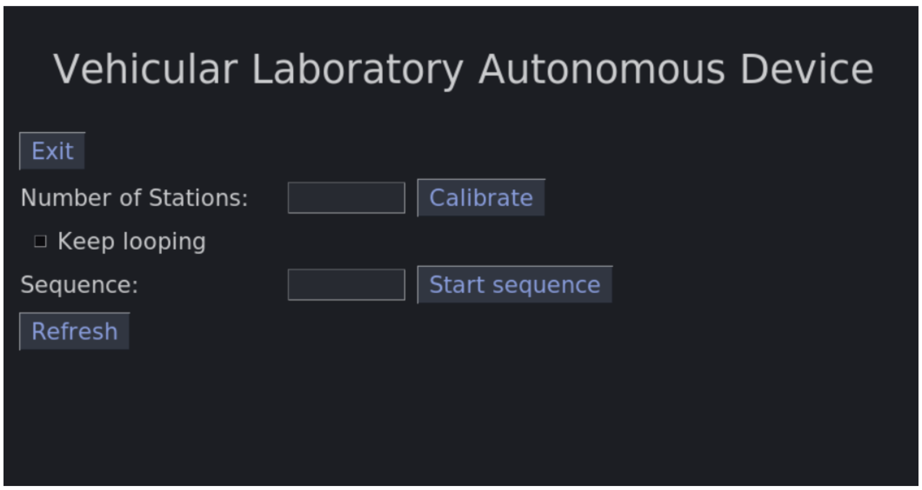


Figure 22: GUI

Figure 22 shows an image of the GUI that was created to demonstrate the functionality of VLAD. With the current GUI, the user can start calibration and move the truck to a specific station. During calibration, the user computer gets angle data from the truck via UDP. Using this data, the track can be mapped on the GUI. Currently, the main computer does get the data. The next step would be to simply map the track on a plot and embed the plot onto the GUI.

12 Power Distribution

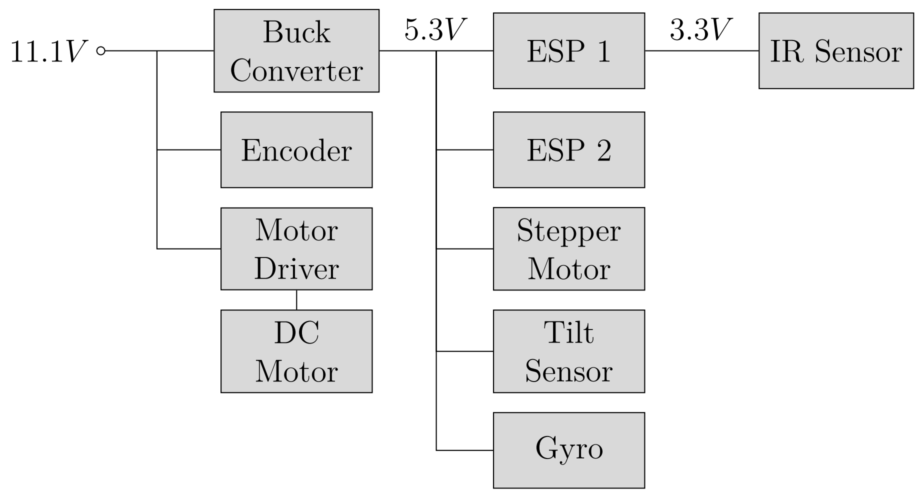


Figure 23: Power Distribution Diagram

Figure 23 shows how power is distributed among all the peripherals of the system. Initially, we thought using four 9V batteries would do the job. However, while testing, we realized that a standard 9V battery could not output the necessary current to run all the peripherals. We substituted all the 9V batteries for a single **11.1V LiPo battery** with an energy capacity of **3000mAH**.

The LiPo battery provided enough current draw to run all the peripherals for the system. Some peripherals such as the controllers (ESP32) required a voltage of 5.6V. We used a Buck converter (DC to DC step-down converter) to convert the 11.1 V to 5.6 V, which is then used to power a few peripherals such as the ESP32. In addition to this, the ESP32 controllers can also provide a 3.3V output, which is used to power the IR sensor.

Once all the peripherals were connected, we tested the current draw on the battery when the system was idle and when the system was running with high resistances being applied to the drive wheels. Table 2 shows the results of this test.

Table 2: Current Draw of the Whole System

State	**Current Draw**
IDLE	0.4 A
RUNNING	0.5 A

Using information from Table 2, we can compute the duration our system can run without charging. Assuming that the system is constantly running, equation 9 shows that the system can run for 6 hours straight.

$$\text{Max Time} = \frac{3\,\text{Ah}}{0.5\,\text{A}} = 6\,\text{h} \tag{9}$$

13 Results

Evidence of our results can be found in Appendix A in the form of links to our GitHub repository and videos of our system operating. When the system was manufactured and tested, it was able to meet all the design objectives set forth by our client, while maintaining a relatively low cost. We estimate that the cost to the user to set this system up in their lab would be $350 to create the tilting platform/truck ensemble, plus an additional $6 per foot of extrusion (including clamps, turn pieces, etc.) for the rest of the system.

Figure 24: VLAD

14 Future Work

While we have provided a good starting point for any laboratory to begin automating its experiments, there are still a few tasks that can be done to further advance this project.

- **PCB:**Design a PCB to mount all electronics neatly.
- **Battery Station:** Create a battery station such that the truck can charge itself autonomously.
- **Scheduler:** Create a scheduler to automate tasks running in parallel, i.e. if multiple samples are at different stages of the experimentation process.
- **Event Based Timeouts:** Truck will only move to the next station once a particular task is completed by a laboratory instrument.
- **User-Friendly GUI:** Current implementation of the GUI is fairly simple. Future work on the GUI could make it more visually appealing and give the user more control functionality.
- **T-junction:** Design a T-junction that permits the track to split. This way the truck can move in different paths which allows it to complete more autonomous laboratory work.
- **End-effectors:** Develop end-effectors, such as a robot arm, that can be placed on the tilting platform to perform a task at a station.

A Links

The main source code for this project can be found on [GitHub].

Video demonstrations of our project can be found on [Google Drive]