ECE 459 Final Project Report
Luke Andresen, George Fang
December 13th, 2024

Table of Contents

Introduction and Motivation:

The S.T.O.P. (Smart Technology Optimized for Protection) Biking project aims to enhance bicycle safety through an intelligent, automated braking system that integrates object detection, user feedback, and regenerative power management. As biking becomes an increasingly popular mode of transportation, both for health and environmental reasons, the risks associated with urban cycling, such as collisions with vehicles, pedestrians, or stationary obstacles, also rise. Addressing these risks motivated the development of an innovative system to augment traditional biking safety mechanisms.

The project combines hardware components like LiDAR for object detection, a brake actuator for automatic braking, and sensors for velocity and orientation. The system is designed to autonomously detect crash threats, calculate time-to-impact, and, if necessary, initiate braking to prevent accidents. User-centered design is also prioritized, with a comprehensive interface providing real-time feedback, including current speed, total travel distance, and alerts for imminent braking scenarios.

Motivation for the project stems from the critical need to improve safety in cycling without diminishing the user experience. By automating key aspects of crash prevention and offering intuitive user feedback, S.T.O.P. Biking seeks to reduce accidents and increase confidence for cyclists in complex environments. Furthermore, the incorporation of regenerative battery technology not only reflects a commitment to sustainability but also significantly enhances the user experience. By extending battery life and reducing or even eliminating the need for recharging or replacing the battery, the system becomes more convenient and reliable for the consumer. This longer operational life ensures that cyclists can focus on their rides without the frequent interruptions of power concerns, making the product both practical and user-friendly. This innovation has the potential to significantly impact the future of cycling, making it safer and more accessible to a broader audience.

System Specifications:

Based on our technical requirements, the S.T.O.P. Biking project can be broken down into four distinct subsystems: object detection, braking, user interface, and battery management.

1. Object Detection: For the system to effectively prevent collisions, it requires a reliable mechanism to detect obstacles and assess the time-to-impact. This subsystem has two core requirements:

A. Obstacle Sensing: The object detection system, powered by a LiDAR sensor, must **detect objects at a range of at least 30 meters**. This ensures the system has sufficient time to calculate braking scenarios, even at higher speeds.

2. Braking System: The braking system is central to ensuring user safety in high-risk scenarios. This subsystem involves both manual and automated functionalities with two main requirements:

A. Auto-Braking: The brake actuator must **engage the brakes automatically when the system detects that the rider is within 20% of the stopping distance** from an obstacle.

3. User Interface: Providing intuitive feedback to the rider is essential for usability. The user interface subsystem is composed of two primary requirements:

A. Data Display: The interface must continuously update and **display the speed (with ±5% error), distance (with ±1% error), and ride duration (with ±0.1% error).** This ensures the rider has accurate and actionable information during their journey.
B. Alerts and Warnings: **Visual and auditory signals** must notify the rider of imminent braking scenarios **1.5 seconds before the auto-braking system engages**. This alert system includes a flashing red display and a buzzer to grab the user's attention.

4. Battery Management: Battery performance directly impacts the usability of the S.T.O.P. system, and this subsystem has one critical requirement:

A. Longevity: The **battery must sustain the system for a minimum of 3 hours per charge**. By using regenerative charging technology powered by the rider's motion, the system reduces reliance on external recharging, making it more convenient for users and extending operational lifespan.

Hardware and Software Design:

Overall, our system is built upon a Teensy 4.1 microcontroller. As above, we can break our system down into 4 key subsystems.

1. Object Detection:

Our abject detection system is built around a Garmin LiDAR-Lite v3HP, which allows for up to 40 meters of range and supports polling rates over I2C far beyond our requirements for this project. We then designed a mounting bracket to secure the sensor to the front of the frame.

Figure 2: Garmin LiDAR-Lite v3HP

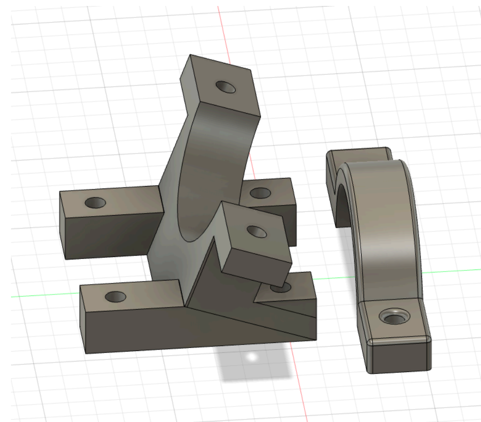

Figure 3:LiDAR Mount

Figure 4: Completed LiDAR System

The Teensy 4.1 polls the LiDAR sensor at a consistent rate and filters the data to provide a smooth output of the distance to the object ahead. We use multiple successive data points from the LiDAR to calculate both the speed of the object (relative to the bike) and the time to impact if no corrective action is taken.

2. Braking System:

First, to discern how fast the user is moving, we mounted 8 evenly spaced magnets along the spokes of the front wheel and placed an A3144 hall effect sensor on the frame of the bike. Each time the wheel complets 1/8th of a rotation, the hall effect triggers an interrupt on the Teensy, where the program discerns how long it has been since the last interrupt, and converts that time to the speed of the bike, as the circumference of the wheel is a fixed value.

We have a robust 3D printed bend pipe that allows for simultaneous braking of the system from both the user and the computer.

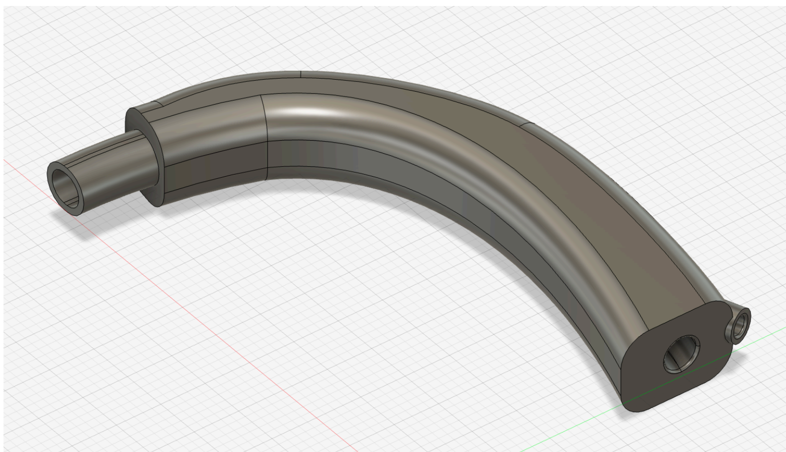

Figure 5: Bend Pipe

Figure 6: Dual Run Cables

Next, a mounting bracket holds a DS51150 12V 150 kg servo in place as it pulls the brake line. A 5200 mAh battery supplies the high power demands of the servo.

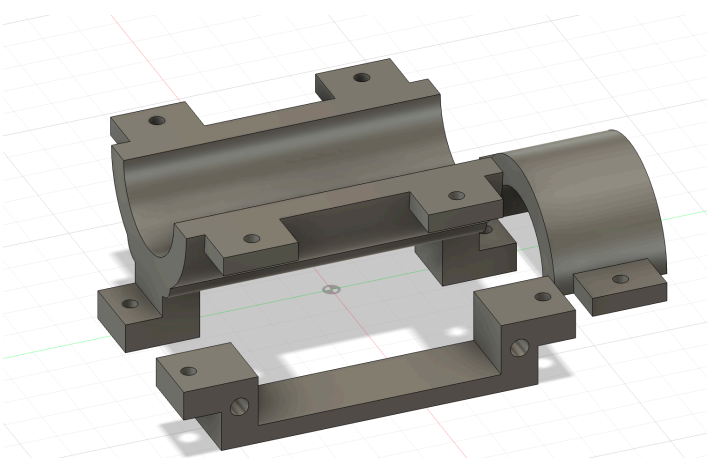

Figure 7: Servo Mounting Plate

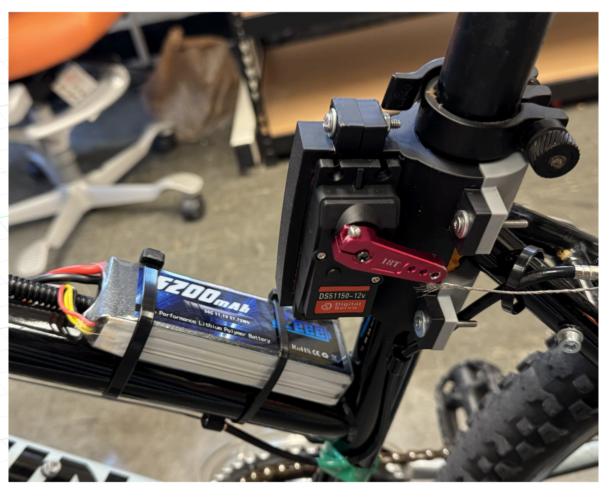

Figure 8: Mounted Servo and 3S Lipo

To decide when to actuate the servo, we did real world testing with the bike to come up with functions that convert the speed of the bike at the moment of braking to the distance and time it takes to stop. We then carefully tuned the functions and found thresholds that trigger the braking. Specifically, our program uses the two distance and time functions to come up with how long (in time and distance) a stop would take at the current speed. It then checks the current distance and time to impact of the object in front of the bike, and if both of these values are within the tuned thresholds of the calculated values above, then a stop is triggered.

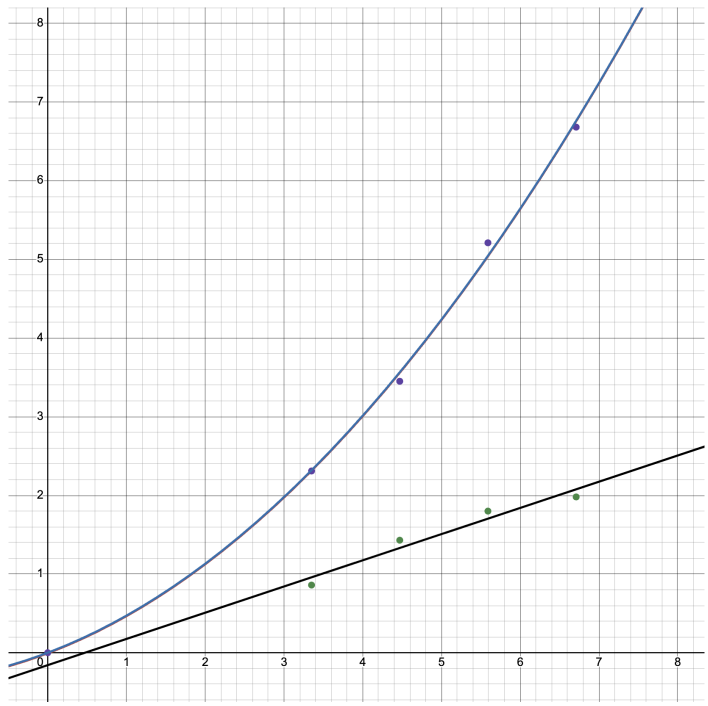

Figure 9: Example Brake Calculations of Speed (m/s) to Distance (m) in Blue and Speed (m/s) to Time (s) in Black

3. User Interface:

To interface with the user, we use an Adafruit Color TFT LCD display, which provides key metrics to the user, like bike speed, distance traveled, distance to the object in front, relative speed to the object in front, time to impact, and total ride time. The screen also flashes red when a brake is triggered. The Teensy interfaces with this screen using SPI and updates the screen each time new LiDAR data is polled. A basic buzzer also indicates to the user when the system is about to brake, so their attention is drawn to the road and they are prepped for the sudden deceleration.

All of this, plus an Adafruit ICM20948 IMU (ultimately unused due to its data going unneeded), is housed in a 3D printed casing on the handlebars.

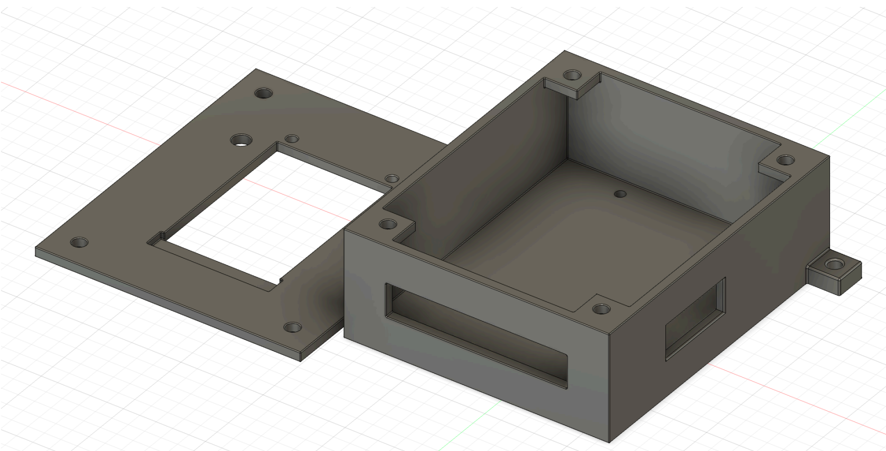

Figure 10: UI Mounting Box

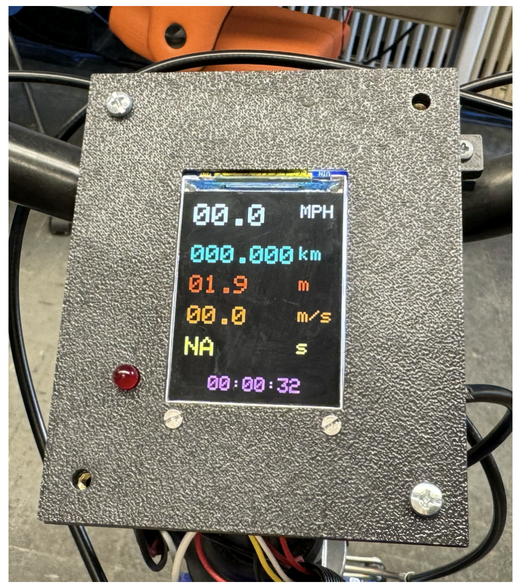

Figure 11: Active UI

4. Battery Management:

To power the system, we use an Adafruit 3.7V Lithium Ion Battery Pack rated to 6600mAh. However, to allow for dynamic charging of the battery while the user pedals, we implemented an Adafruit Powerboost 1000C that stepped the ~3.7V from the battery up to 5V to power the Teensy and all other sensors and components that required 5V. This chip also had a charging input, allowing for the battery to be charged in use, and the load to be powered using the charger when the battery is fully charged. We also included a switch that would enable and disable the load side of the Powerboost 1000C, acting as the system's power switch.

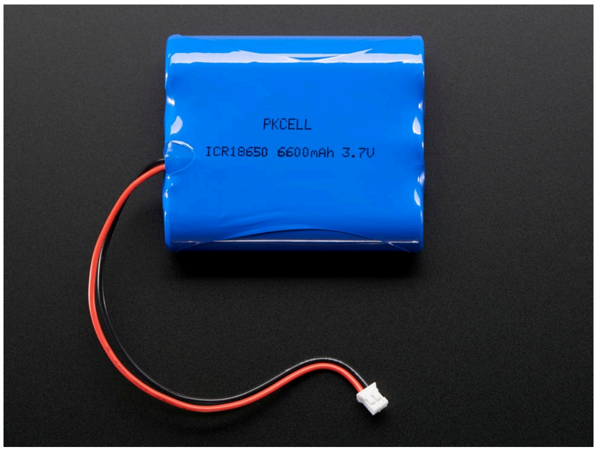

Figure 12: Adafruit 3.7V Lithium Ion Battery Pack

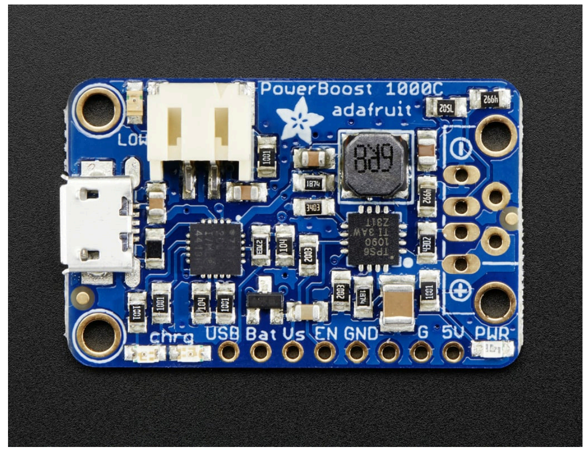

Figure 13: Powerboost 1000C

To provide an input to the charging side of the Powerboost 1000C, we bought an 12VAC bike dynamo, used an AC/DC converter, and then ran the 12VDC through an Adafruit MPM3610 buck converter to supply a steady 5V to the chip.

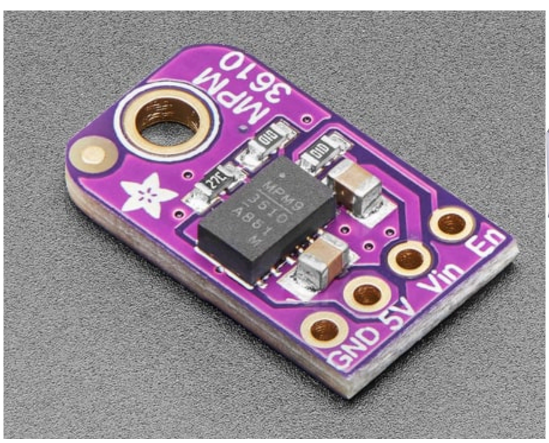

Figure 14: Buck Converter

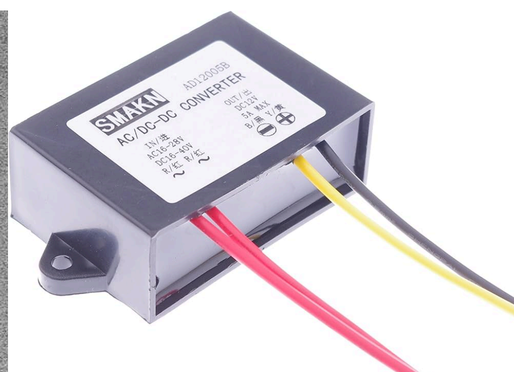

Figure 15: AC/DC Converter

Figure 16: AC Dynamo

All of these systems (apart from the dynamo) are housed in a 3D printed mount.

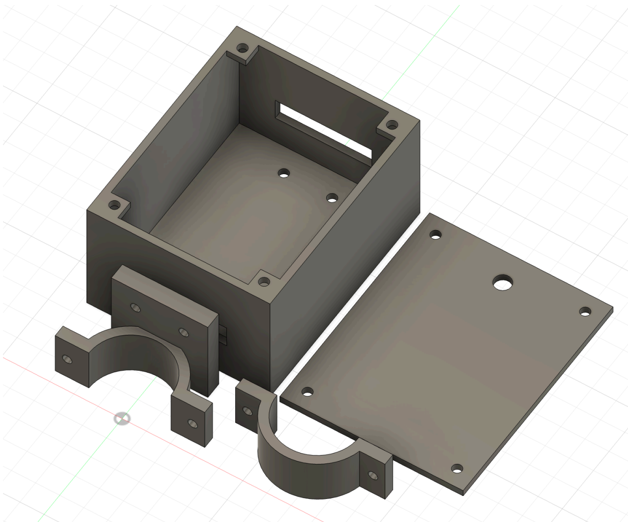

Figure 17: Battery Box CAD

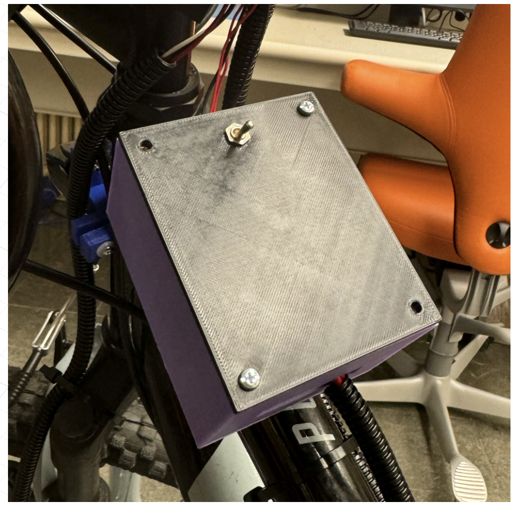

Figure 18: Assembled Battery Box

Lastly, as mentioned above, we included a 3S Lipo to power the servo, decreasing the high demand on these lower power systems.

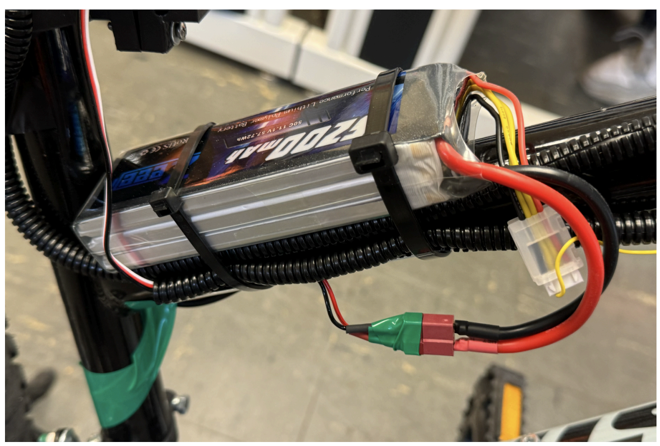

Figure 19: Mounted 3S Lipo

Below is our general hardware architecture.

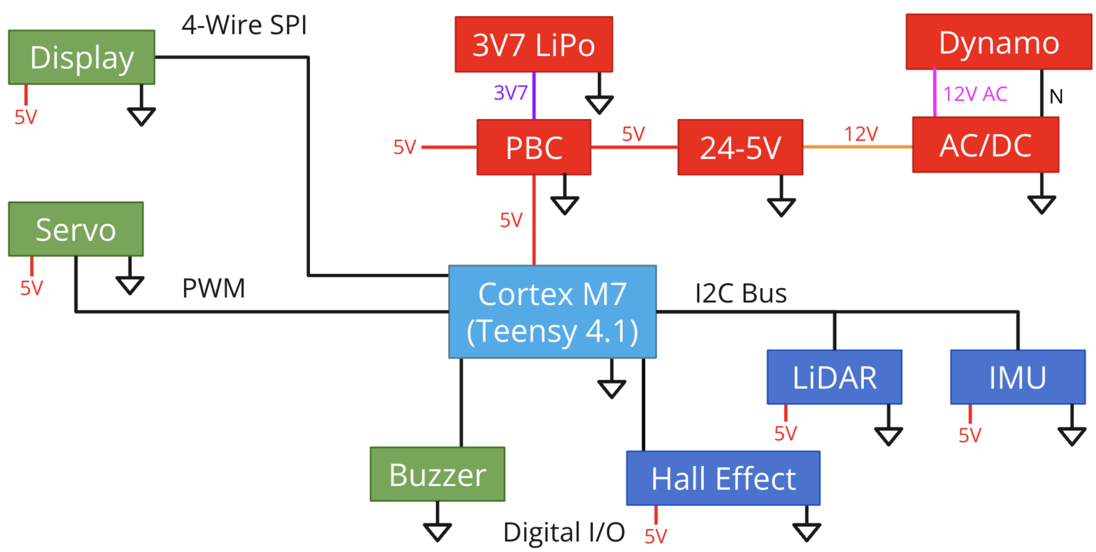

Figure 20: Hardware Architecture

Below is our general software control loop. For more information, visit the code repository linked at the end of the report

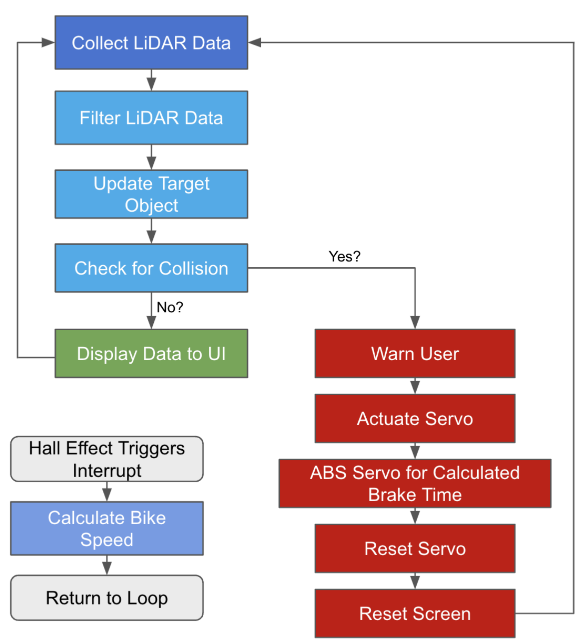

Figure 21: Software Architecture

Experimentation and Evaluation:

1. Object Detection:

- To evaluate our object detection system, we rigorously tested the capabilities and limitations of the LiDAR sensor. Our approach included the following:
  - *Field of Vision*: We measured the height and width of the sensor's detection cone and determined that its narrow field of vision was advantageous. As a bike is a relatively narrow object, this setup helped minimize false positives by ignoring objects that were not directly in the bike's path.
  - *Distance Measurement*: We developed a simple script to continuously output the distance (in meters) of the closest detected object. This allowed us to analyze the accuracy and consistency of the sensor in real-world scenarios.
  - *Mounting Optimization*: Using CAD software, we iterated through multiple designs for mounting the LiDAR sensor to ensure it was parallel to the ground and pointed straight ahead. This alignment was crucial for accurate detection of objects in the bike's path.
- Challenges Identified: One issue we encountered involved the bike's pre-existing brake cables, which hang in front of the handlebars. When turning, these cables would occasionally pass in front of the LiDAR sensor, causing it to detect the cables as an obstacle. This led the system to mistakenly calculate a sudden jump in the speed of the closest object, triggering an unnecessary emergency brake.
- Outcome: Ultimately we met our target specification of being able to detect objects at a range of at least 30 meters.

2. Braking System:

- Our braking system was perhaps the most mechanically complicated component of the project, and thus went through several iterations throughout the testing cycle.
- To actually actuate the brake, we first had to design a way to have two brake cables share the same braking mechanism on the bike, to allow for the user and the servo to control the brake as needed. We first simply ran the additional cable through a basic brake horn. However, this solution was not robust, as the two cables shared the same exit hole, meaning the entirety of the braking force was imparted on the smaller cable each time the user used the brake, eventually severing the additional cable. We then worked towards a system that had 2 exit holes, one for the existing line and one for the additional line. Originally this system was not strong enough to endure the high forces imparted on the system when braking, and the brake cables sliced clean through it as shown below. After a few failed prototypes, we landed on a reliable final design that was strong enough to

provide tension, yet small enough to fit in the braking mechanism

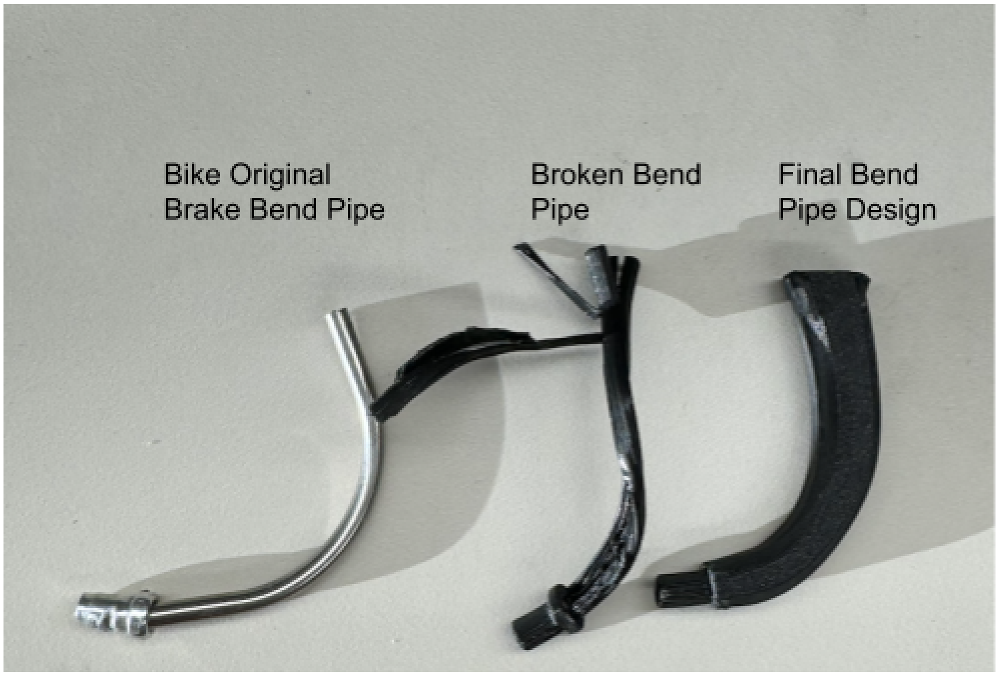

Figure 22: Bike Brake Bend Pipe Timeline

- Once we could run the cable, we needed something to pull the cable. Naturally, a servo proved the best option and we immediately began work to mount a servo to pull the brake lines. While lab testing showed that small 25 kg servos could stop the bike, they simply did not provide enough strength to stop the user at higher speeds. Also, since these smaller servos were on the general power system, their high current draw in stall would force the system to reboot each time a brake executed. Thus, we began using the 150 kg servo and purchased the 5200 mAh 3S lipo to provide the increased power to drive the servo, separate from the existing power frameworks. Mounting the servo did prove to be a challenge, as the high forces that the servo imparted on the brake line would often spin its mount or rupture it completely. We went through many design iterations before settling on an extremely dense mount that could truly hold the forces of the system. During the class demo, our brake performance worked great, but the bike brakes slightly slower than in ideal conditions. This is because the way we attached the brake line to the servo horn allowed for some slippage, loosening the cable. In the future, we would like to create a more robust connection so that tunning is only required once.
- Ultimately, we met our design specification of engaging the brakes automatically when the system detects that the rider is within 20% of the stopping distance. One piece to note is that this 20% mark was set somewhat arbitrarily at the start of the semester and can certainly be changed based on user preferences by altering our braking algorithm.

3. User Interface:

- While fairly straight forward, testing the user interface was very important for user friendliness and usability. We wanted to test three main usages:
  - Data Display: In order to make sure the display works in all environments, we brought the screen around to several different environments (hot/cold, light/dark) to ensure that the screen was functioning and easy to read.
  - Usability: To ensure our UI layout was intuitive, we asked a friend to look at our screen and try to identify what each element represented and why it was important.
  - Visual/Auditory Alerts: For visual and auditory alerts, we simply ensure that the screen turned red and the buzzer sounded when braking was signaled.
- Ultimately, we met our design specifications of displaying the speed (with ±5% error), distance (with ±1% error), and ride duration (with ±0.1% error), as well as providing visual and auditory signals in advance of the auto-braking system.

4. Battery Management:

- Battery management presented unexpected challenges during this project, particularly when we had to switch to a 150 kg servo. The servo's high current draw exceeded the capacity of our single battery to power both the servo and the rest of the system. To address this, we added a second battery dedicated solely to the servo. In this configuration, the bike's dynamo charges only the system battery, while the servo battery operates as a standalone power source.
- Dynamo System Challenges:
  - During testing, we encountered issues with the battery regeneration system using the bike dynamo. The dynamo, being inexpensively built, struggled to maintain consistent contact with the bike tire, resulting in an unstable flow of electricity to the battery. This inconsistency caused the Powerboost 1000C module to repeatedly toggle its internal charging switch, paradoxically draining the battery instead of recharging it.
- In our first test ride, which lasted 13.26 minutes, the system battery voltage dropped from **4.09 V** to **3.21 V**, a loss of **0.88 V**. Surprised by this unexpected drain, we conducted a second test ride, this time without activating the dynamo. Over a 15-minute ride, the battery voltage decreased from **4.15 V** to **4.09 V**, a loss of only **0.06 V**, which aligned more closely with our expectations.
- To further validate these results, we completed an extended ride lasting 3 hours and 45 minutes. During this period, the battery voltage dropped marginally from **4.15 V** to **4.08 V**, demonstrating that the system operated efficiently when the dynamo was not in use. Based on these results, we calculated that the battery loses **0.07 V** every 4 hours. Given a maximum charge of **4.2 V** and a kill voltage of **3.0 V**, we estimate the system can run for

approximately: $1.2\,V \div 0.07V/hour \times 4\,hours = 68.6\,hours$, nearly 3 straight days of usage.

- For the 5200mAH battery charging the servo, at operating voltage of 11 V, the idle current draw is 4.5mAH. For stall torque of 150 kg/cm, the stall current is 7.4A. If idle, the battery would last for 1155.56 hours (48 days). Assuming each brake is 2 seconds long, we can also calculate that the battery would be good for 1,268 brakes.

Discussions and Conclusions:

The S.T.O.P. Biking project successfully demonstrated the feasibility of integrating automated braking, object detection, and user feedback into a cohesive system designed to improve cycling safety. Over the course of the project, we developed a robust hardware and software framework that allowed our system to perform as intended, meeting most of our design specifications. However, there were several challenges encountered during the development process that provided valuable learning experiences and insights for future improvements.

Key Accomplishments

- Object Detection: The Garmin LiDAR sensor performed well, detecting objects accurately within the required range of 30 meters. The narrow cone of detection proved advantageous for a bike's slim profile, reducing false positives. This allowed the system to calculate accurate time-to-impact and reliably trigger braking.
- Braking System: Our dual braking mechanism allowed for both manual and automated braking without interfering with one another. The integration of a high-torque servo and carefully tuned braking algorithms ensured that the system could respond to collision threats effectively.
- User Interface: The display and auditory alert systems provided intuitive and real-time feedback to the rider. Testing in various environments ensured the display was legible and effective.
- Battery Management: Despite initial challenges with the dynamo system and power demands of the servo, we implemented a sustainable power system using dual batteries. The system achieved an impressive operational lifespan of nearly 69 hours of continuous use on a single charge and a fast recharge time of 45 minutes.

Challenges and Lessons Learned

- Dynamo System: The dynamo's inconsistency highlighted the importance of reliable hardware. A better-quality dynamo could resolve the issue of fluctuating power supply and make regenerative charging more effective.
- Mechanical Robustness: The servo and dual-cable braking mechanism required multiple iterations to withstand the forces involved in braking. This underscored the importance of robust and precise mechanical designs.
- Complexity of Real-World Scenarios: The system's sensitivity to pre-existing brake cables and road conditions showed that further refinements are needed to handle diverse riding environments.

Future Work

- While the project was successful, several enhancements can be made to improve the system:

- Weight and Road Condition Settings: Introducing settings for rider weight and road conditions (e.g., wet or dry) could dynamically adjust braking force and thresholds for improved safety and performance.
- Servo Disable Option: A setting to disable the automated braking system could be useful for scenarios like weaving through traffic, crowded pedestrian areas, or nature trails.
- IMU Integration: Utilizing the IMU to detect hill tilts could enable dynamic braking adjustments, applying more or less braking force depending on incline or decline.
- Refinement of Mounts: Improving the robustness of mounts for both the servo and LiDAR sensor will ensure consistent performance in high-stress conditions.

Conclusion

The S.T.O.P. Biking project showcased the potential of smart safety systems to enhance cycling experiences in urban environments. By combining advanced sensing technology, automated braking, and user-friendly feedback, the system offers a significant step forward in bicycle safety innovation. Despite some mechanical and hardware challenges, the project successfully met its primary objectives and laid a solid foundation for future development. With further refinements, this system has the potential to become a transformative safety feature for cyclists, making biking safer and more accessible for all.

Code Repository