ERSAB: Electronic Response System for Assisted Braking

Neil Stimpson (nts2)
Vassily Petrov (vassily2)
Prerak Sanghvi (psangh5)

Final Report for ECE 445: Senior Design Laboratory, Fall 2019
TA: Christopher Horn
11 December, 2019
Project No. 09

Abstract

To avoid the error-prone, human-reliant braking systems on bicycles, the Electronic System for Assisted Braking (ERSAB) allows for ease of use of its brakes via servo motors that are connected to the brake for more controlled, precise, and powerful braking that is available on current bicycles today. The bulk of the design consists of a metallic rectangular box placed in the center of the bicycle that is able to house a load cell to measure the force applied to the brake, a Li-ion battery to power the electronics, and a compact printed circuit board (PCB) that has a microcontroller for communication protocols with the servo motors and speed sensors, which are mounted on the front and rear tires via 3D-printed mounts. With all of these electronic devices working in tandem, the bicycle is able to brake smoothly and provide various data points for further modification. Specifically, characteristics such as wheel rotation velocity and acceleration, applied braking force, and incline/decline biking, ERSAB can be further modified to the user's pleasure to account for phenomena such as bike flipping and fishtailing.

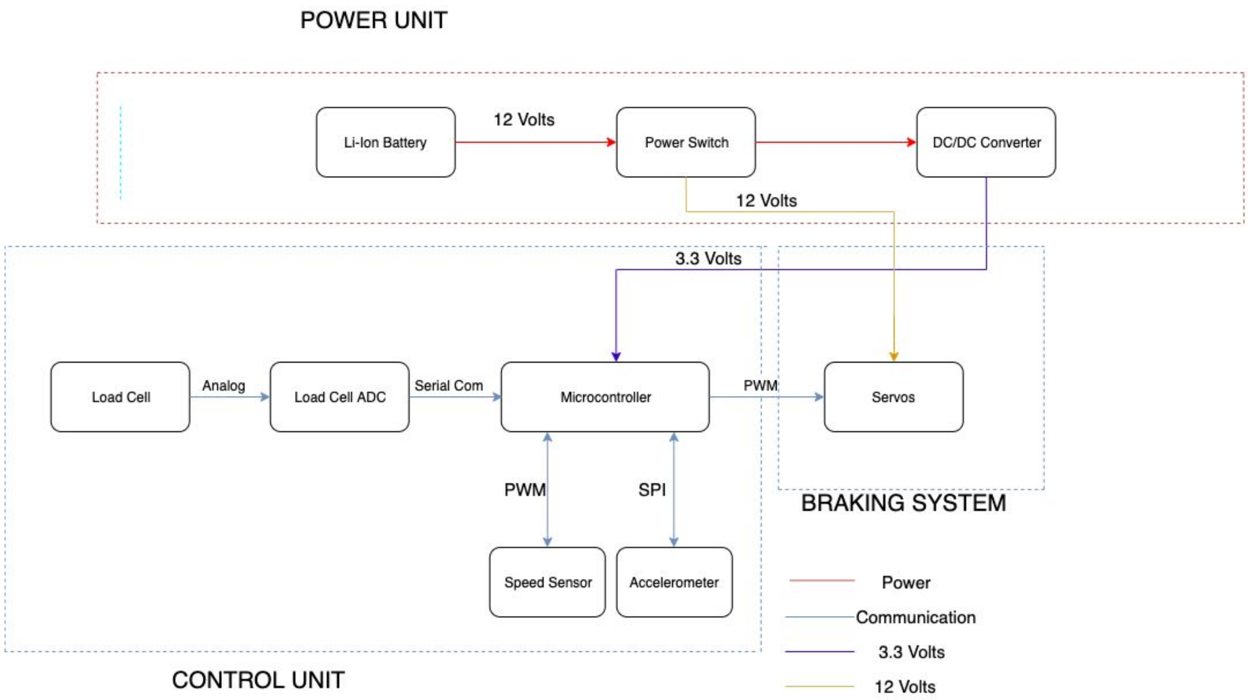
Contents

1. Introduction

Braking on a bicycle is seldom consistent when relying on human reaction, whether the brakes are being applied too fast or too slow. In both cases, unpleasant bike rides and sometimes even serious injury can occur. Since electronically motorized movements are more precise than humans by their very nature, an electronic response system that aids in the braking process can reduce these aforementioned risks. Also, the advantage of ERSAB is that it can maximize braking performance without having the wheels lock. It removes the hesitation that riders face if they must brake suddenly -- it also makes it easier to apply the brakes for those that have weak hand strength and reduces wrist and hand strain from having to enable the brakes frequently. In 2015 there were 818 cycling fatalities according to the NHTSA[2] in the United States. 25.7% of these fatalities are due to no fault of the cyclist and 34.9% are due to failure to yield to the right of way. In both cases, more cyclists may have survived if they were able to brake faster or take evasive action without locking up their brakes.

ERSAB uses one PCB which processes sensor data and actuates servo motors that activate the brakes more seamlessly. It can be broken up into seven modules that encompass the entire design: *Power Delivery & Storage*, *Braking System Interface*, *Microcontroller*, *Motor*, *Wheel Rotation Sensor*, *User Feedback*, and *Accelerometer*. Naturally, *Power Delivery & Storage* refers to the Li-ion battery and charger that power our entire system, the *Braking System Interface* that consists of the brake line, a modified handle brake, and load cell that reads the force applied, the *Microcontroller* that communicates the data necessary for the optical sensors, the *Motors* that applies sufficient torque to apply the brakes, the *Wheel Rotation Sensor* with its 3D-printed optical ring to register rotation speed of the front and rear wheels, the *User Feedback* system that displays battery life on an LCD monitor, and the *Accelerometer* that detects the incline of the bicycle. Notably, with such an elaborate foundation of data processing, ERSAB is ready for future software algorithmic modification, specifically to address dangerous, but common biking circumstances such as flipping and fishtailing.

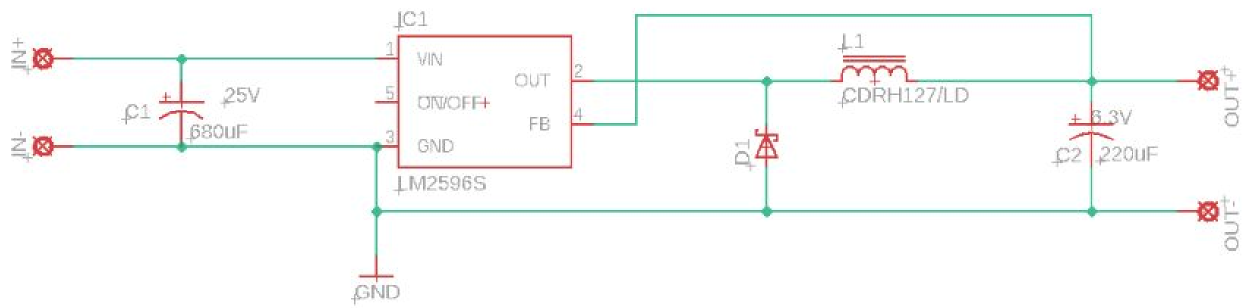
2. Design



2.1 Power Delivery & Storage

We used a 3S 12-V lithium-ion battery to run six sensors and two servo motors. We powered the servo motors directly from the battery at 12 V and used buck switching voltage regulating circuits to get 3.3 V and 5 V for the sensors. We used the buck topology to improve the overall efficiency of the power system. We were able to achieve 90% efficiency using the buck converter topology which would have been close to 50% if we went with the more traditional voltage divider. We used Texas Instruments TPS563240 buck converter chip which has an operating voltage of 4.5V - 17V and runs at a frequency of 1.4 MHz thus providing a very steady supply of DC. We also used voltage monitors in order to collect data on the amount of power pulled by the servo motors. We used the Texas Instruments INA260 chip which is a digital out power monitor with overcurrent protection.

We provided a 5V bus and attached it to header pins thus enabling us to expand the project in the future if we decide to add more sensors.



2.2 Microcontroller

We used an ARM Cortex-M0+ chip and attached a 24 MHz crystal. We used an EFM32HG222F64 provided by Silicon Labs. This chip has 64 kB of flash memory and 8 kB of RAM. We wrote around 1000 lines of code which used 24.76 kB of flash memory and 0.66 kB of RAM. It will need to be sufficiently energy efficient to last for hours running from the battery. The chip is engineered to be low powered and also supplies all of the inputs to handle the connections from all of the sensors.

The chip uses the I2C bus to communicate with the INA260 voltage monitors. We supplied both monitors different IP addresses and assigned them as slaves and the chip being the master. We used the Pulse Width Modulation (PWM) pins on the chip to send signal pulses to the servo, thus controlling the servo rotation angles precisely. We used the analog pins to get the data from the optical sensors. This data was processed in the chip by using a dataset of eight points at once and running a rectangular window to average out the data points. Using this method, we were able to clean some of the noise from the sensor but still had some inconsistencies. Some future work includes having a better filter to average the data and a sturdier optical disk.

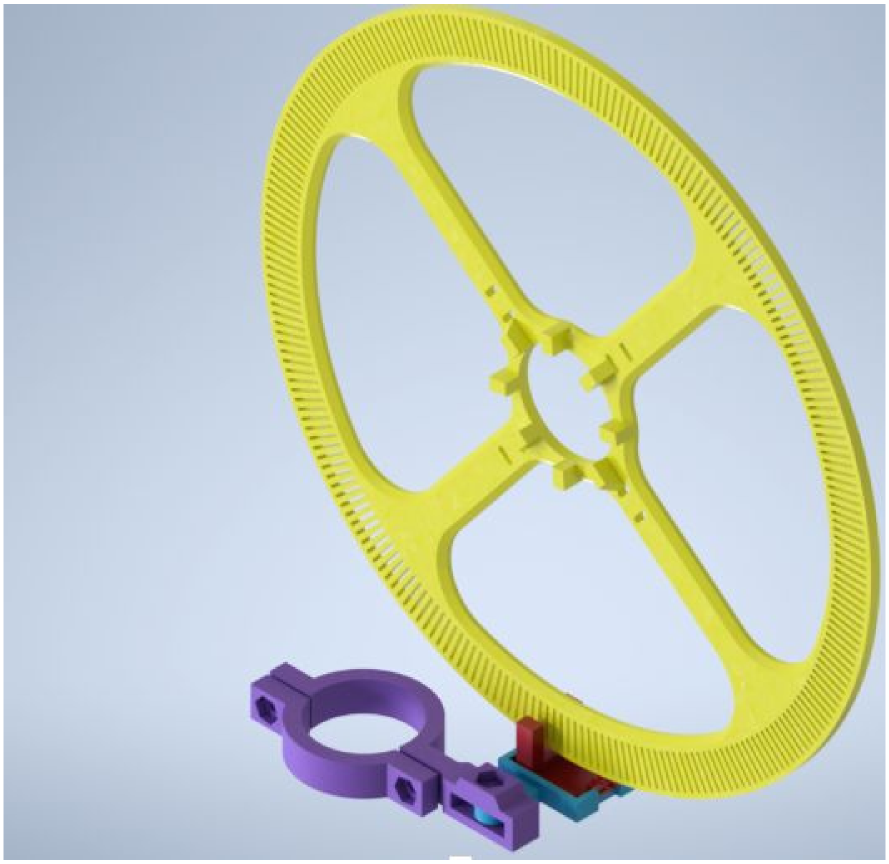
We used the Serial Interface to communicate with the HX711 chip and the raw data was used to calibrate the brakes with the servo motor. The chip uses about 120 uA/MHz which is very small; thus, were able to save a lot of battery life due to the high efficiency of the chip. We programmed this chip using the Simplicity Studio IDE and uploaded the code using the SWD-JTAG interface on an EFM32 microcontroller board.

There were a few unused pins on the chip which we routed to header pins thus enabling future expansion of the project. We had the foresight to add a 5-V power supply, thereby providing us with endless possibilities for the future.

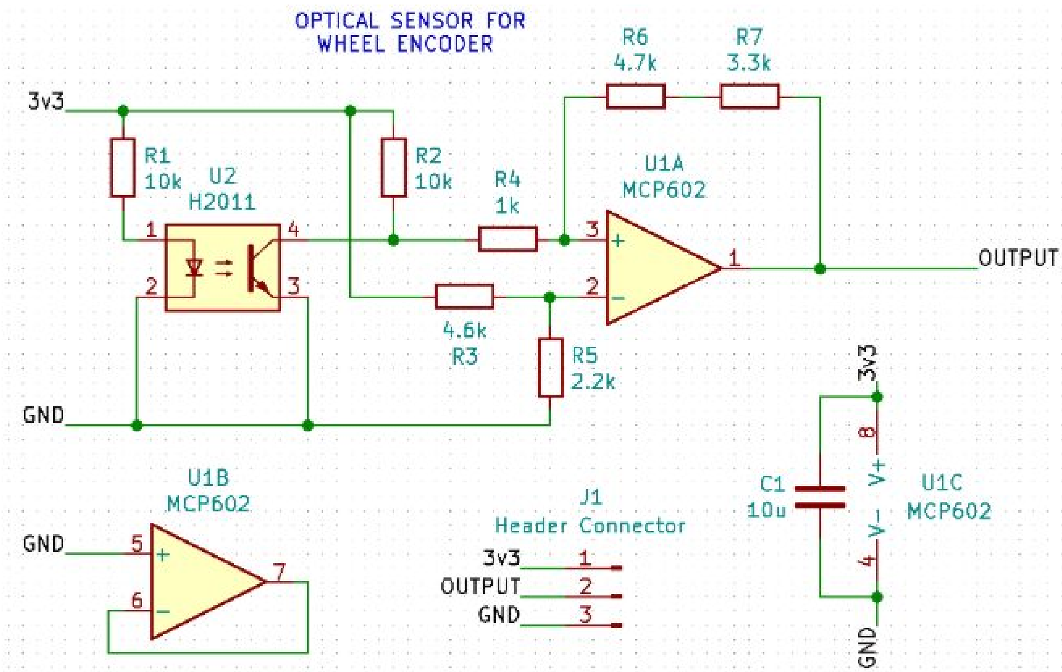
2.3 Motors

The type of servo motor used is the RoboStar SBRS-5314HTG. This motor was chosen based on its speed and torque so that it activates with sufficient strength to slow down the wheel. The motor has a high torque of 54 kg cm which is equivalent to 3.9 ft lbs of force. The motors run on 12 V which was supplied from the battery. The motors supplied more than enough torque to pull on the brake lines. Specifically, the motors would consume an average of 45 W each when activated. We had designed our system to handle 5 A of current to be drawn by each servo. In practice, the servo drew about 2.8 A of current under full tension. The motor would rotate 90 degrees in less than 0.4 s, which is a quick response to stop the bicycle in time.

2.4 Wheel Rotation Sensor



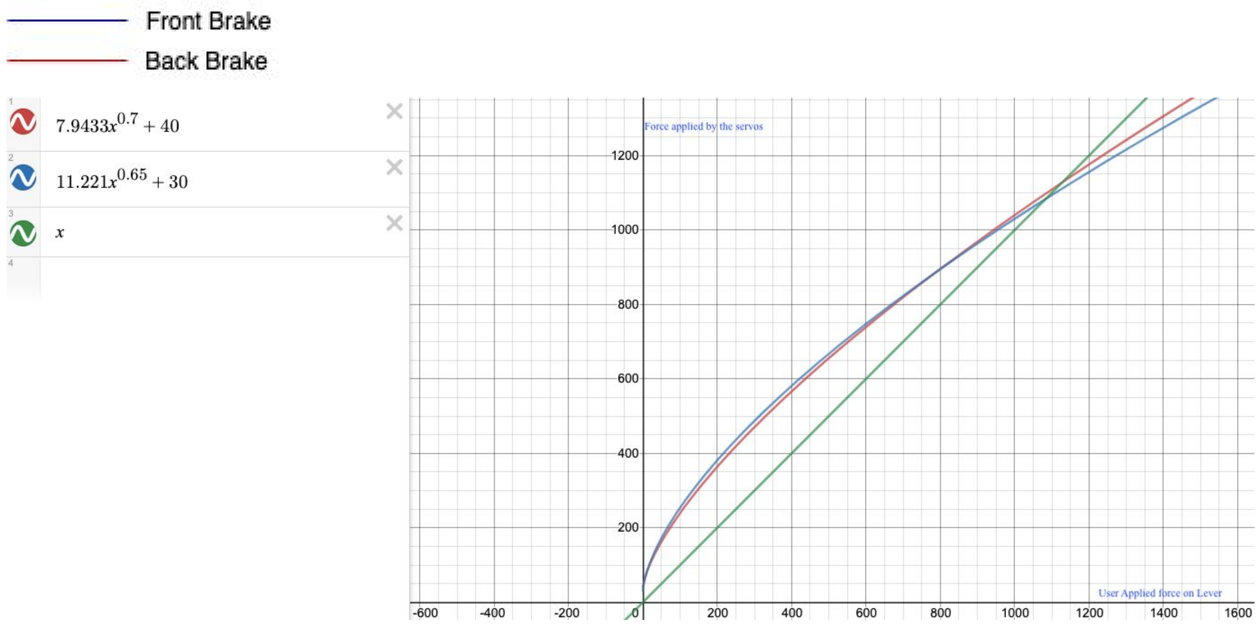
The wheel rotation sensor module consists of a plastic 3D-printed ring with 200 radially spaced slits to allow for the passage of light. An infrared optical sensor is mounted to the fork tangent to the ring. As the ring rotates, the slits will pass through the optical sensor and will intermittently block the passage of infrared light. A custom built optical sensor is attached directly to an infrared detector and outputs a signal when it detects light on the photoresistor attached to the other side. This sensor is sampled using an interrupt routine on the microcontroller and will handle speeds of up to 40 MPH. The output signal is routed to the microcontroller within the Master Housing Unit.



2.5 Braking System Interface

The braking system interface consists of the original hand brake attached to a short segment of the brake cable routed to the master housing unit. The end of the brake cable is attached to a load cell which converts mechanical force exerted by the hand to an electronic analog value. The analog value will be converted to a digital signal by the HX711 ADC chip. This digital value is communicated to the microcontroller which converts to a PWM signal thus controlling the servo motors. We used logarithmic braking algorithms for the front and the back brake in order to provide a clean braking experience while ensuring the user safety.

The front brake servo uses the equation $11.221\,x^{0.65} + 30$ to apply tension on the brake cable. The back brake servo uses the equation $7.9433\,x^{0.7} + 40$ to apply tension on the brake cable.



As depicted in the figure above, the blue line depicts the front brake, the red line depicts the rear brake, and the green line depicts a linear braking system. From this graph, we understand that the front brake is working a bit harder than the back brake due to the additional momentum. If a user presses the brake to its maximum distance, the system recognizes it and softly starts applying the brakes to prevent locking of the wheel and thus incorporating a higher sense of safety.

2.6 Accelerometer

An accelerometer mounted on our PCB via a breakout board detects sudden jolts and shifts which would aid in determining when the bike loses control or begins to flip. With the accelerometer, the inclination of the ground the bike is riding on can be determined. We found that the RX and TX on the accelerometer board were flipped thus hindering our goal. As this is a simple fix, we are optimistic to get this to work on our ERSAB version 3.0 PCB board.

3. Design Verification

3.1 Power Delivery & Storage

3.1.1 Battery Charging

The requirements of the project were changed to charge the battery externally. However, even with the external battery charger, it charges in about 1 hour.

3.1.2 Battery Actuations

The estimated power consumption was consistent with the observed data. Applied power while braking was measured to be 42.6 W (both servos). Measured braking time was 3.2s. This gives a total of (3*60*60*3.6*3/(42.6*3.2)) = 857 actuations. This means hours of biking.

3.1.3 Dimensionality

The entire Master Housing Unit fits within the frame of the bike. The bike can be ridden with the unit operating entirely on the bike.

3.2 Braking System Interface

3.2.1 Applied Force

The load cell we chose is rated to 250 N. This well exceeds the target of 100 N.

3.2.2 Microcontroller Readings

We probed the serial lines of the HX711 and measured the period of the serial data to see how often data is sent to the microcontroller. The measured period was 40 ms corresponding to a sampling rate of 25 Hz.

3.2.3 Ride Comfort

Each member of the ERSAB team rode the bike and found the brakes to be easy to press.

3.3 Microcontroller

3.3.1 Velocity Calculation

In order to increase precision, we went with an interrupt-based system rather than polling the wheel rotation sensor. Therefore, the only limitation to the measurement performance is the execution speed of the interrupts. The original 5000 samples/sec figure was based on the wheel rotation sensor speed of 40 MPH. As we met that requirement without overrun in out interrupts or loss of precision, this velocity calculation is met as well.

3.4 Motor

3.4.1 Applied Force

The pull radius was set to be 12.7 mm. With a 280-degree rotation we are thus able to achieve a pull distance of 62 mm -- significantly better than the target of 10 mm. The servos functionally locked the brakes while riding, exceeding the 100 N requirement.

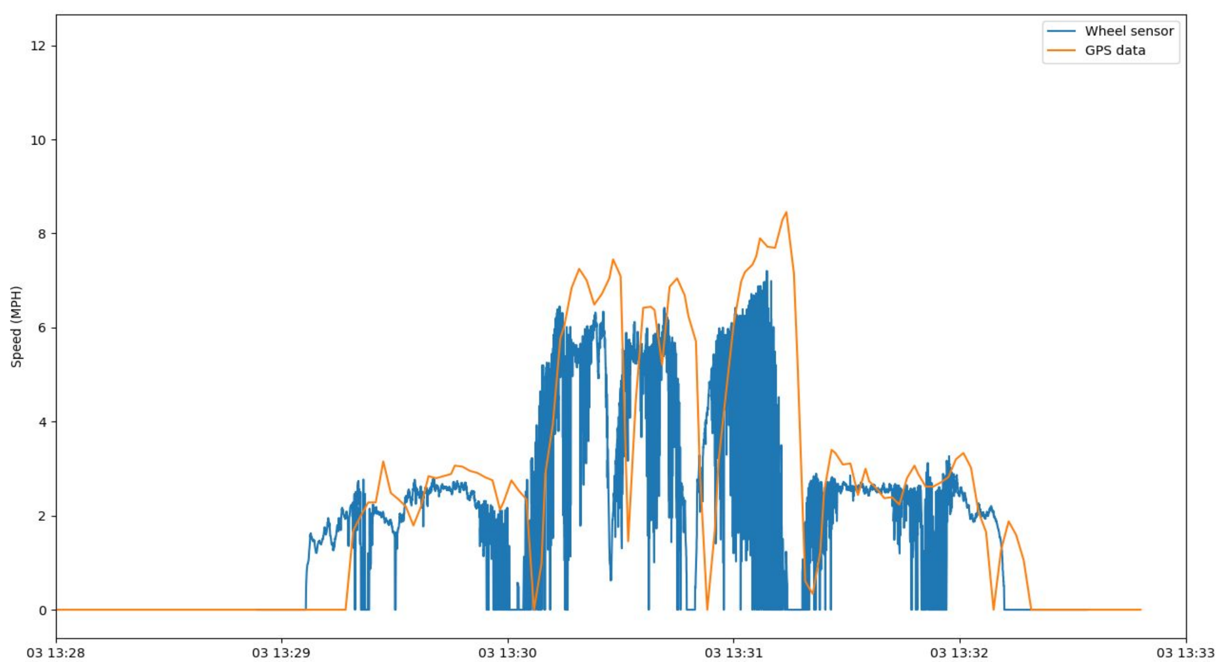
3.2.2 Brake Actuation

Measurement with a stopwatch in accordance with the R.V. table procedure yielded a brake actuation time of 0.605 seconds. While this is slightly higher than our target of 0.5 seconds, in practice this does not make a difference while riding.

3.5 Wheel Rotation Sensor

3.5.1 Wheel Rolling Accuracy

Rather than measuring with a tape measure, we opted to measure sensor accuracy compared to a GPS. We performed a test by recording GPS data from a phone mounted to the bike and recorded the optical sensor readings at the same time.



The results show that the GPS sensor closely follows the optical sensor data, meaning that we are getting accurate measurements of speed.

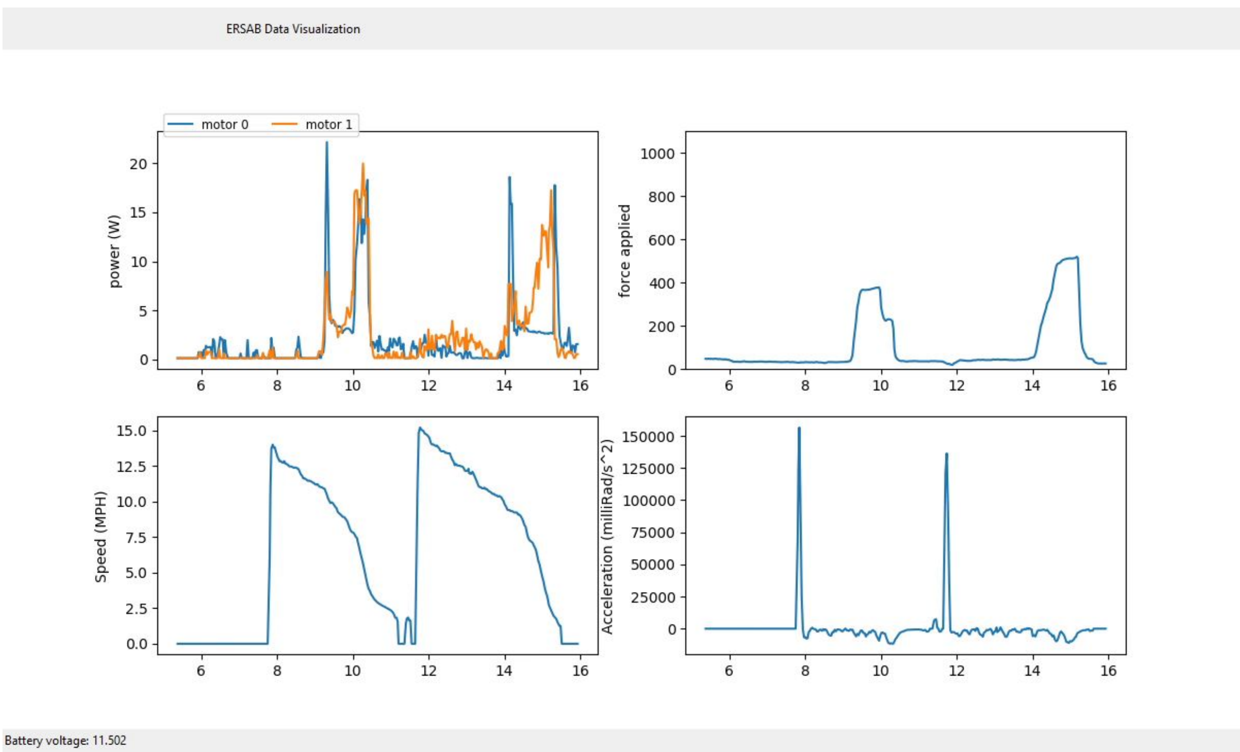
3.5.2 Wheel Speed Maximum

While we were unable to actually ride the bike at 40 MPH, we collected data by standing the bike on the test bench and spinning the wheel as hard as possible. This resulted in a measured speed of 30mph and no gaps or jumps in the data were observed.

3.6 User Feedback

3.6.1 Data Display

We changed the project requirements to display data on a computer over a serial link instead of an LCD display. This PC-based visualizer is able to successfully show power, hand force, speed, acceleration, and battery voltage.



3.7 Accelerometer

We were unable to make the accelerometer function because of problems soldering a QFP chip as well as problems communicating over SPI due to flipped RX and TX pins. This module does not function as we predicted in our R&V table. See section 5.2 & 5.4 for a detailed analysis of the shortcomings of the accelerometer.

4. Costs

Part	Part #	Manufacturer	Quantity	Total Cost
Load Cell Amplifier	HX711	AVIA Semiconductor	1	$6.00
Luggage Scale	8541976354	AlepTau	2	$14.00
3-Cell LiPo Battery	9067000452	ZIPPY Compact	1	$16.00
Optical Sensor	H2010	Generic	2	$4.00
Microcontroller	EFM32HG2222	Silicon Labs	1	$2.31
Power Monitors	INA260	Texas Instruments	2	$10.60
Miscellaneous PCB Components				$15.00**
Accelerometer	SEN-11446	SparkFun	1	$16.00
PCB	N/A	PCBWay	2	$10.00***
3D-Print Optical Encoder Ring	N/A	Champaign-Urbana Community FabLab	2	$12.00
Servo Motor	SBRS-5314HTG	RoboStar	2	$40.39
Brushless Motor Controller	16BL30	Hobbywing	1	$27.79
3D-Print Mounts	N/A	Champaign-Urbana Community FabLab	3	$11.58
Master Housing Unit	N/A	ECEB Machine Shop	1	$907.00*

* *We expect that machinists earn on average $45.35/hour*[4], and it would take them around 20 hours of work to complete the project.
** Approximated
*** Based on PCBWay cost estimate. Actual cost was covered through the ECE445 Project Budget.

Excluding the cost of the Master Housing Unit, we paid $299.82 for our project.

4.2 Labor

According to the University of Illinois at Urbana-Champaign[3], the average starting salary for a UIUC Electrical Engineer with a bachelor's degree is $67,000. This comes out to $32.21/hour. Based on this number as well as our experience and contribution to the project, we decided on labor costs of $33.65/hour, $36.06/hour, and $38.46/hour respectively.

Labor costs for each group member is calculated below:

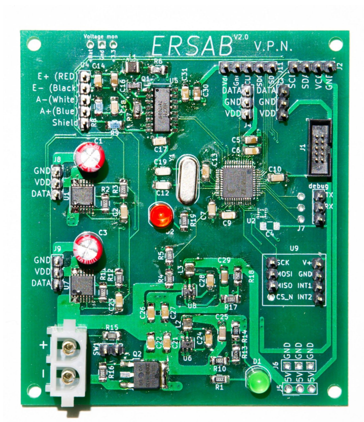
1. Neil Stimpson: ($33.65/hr)*2.5*(16 weeks of the semester - first 2 weeks - 1 week for Fall break - 1 week for finals)*(7 hrs/wk) = $7066.50
2. Prerak Sanghvi: ($36.06/hour)*2.5*(16 weeks of the semester - first 2 weeks - 1 week for Fall break - 1 week for finals)*(7 hrs/wk) = $7572.60
3. Vassily Petrov: ($38.46/hr)*2.5*(16 weeks of the semester - first 2 weeks - 1 week for Fall break - 1 week for finals)*(7 hrs/wk) = $8076.60

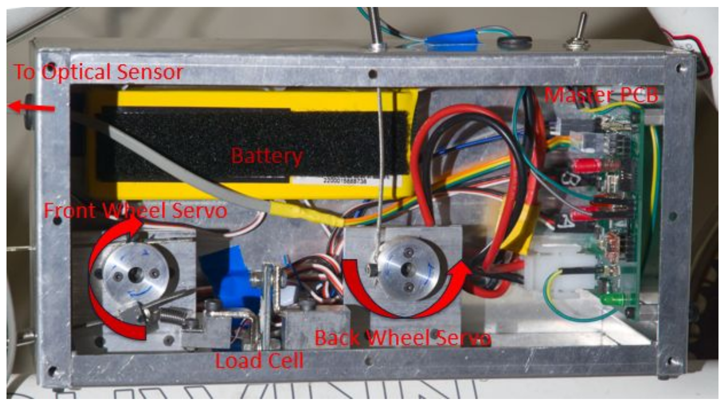
Therefore the total cost of labor for the project is $7066.50 + $7572.60 + $8076.60 = $22715.70

5. Conclusion

5.1 Accomplishments

Our project became a nearly fully-functional electronic braking system that can elicit a controlled braking force via servo motors when manually applying the brakes, communicate and measure relevant data between modules, such as wheel speed and acceleration, force applied on the brakes, and power output from the servo motors. It also utilizes sufficient bicycle real estate to mount the Master Housing Unit and external sensors to maintain a similar riding experience. Lastly, the build is a rigid mechanical design that can withstand average vibrations and other perturbations experienced during riding.





5.2 Challenges

We encountered a plethora of challenges during the duration of our project. Firstly, we went through a couple of iterations of the 3D-printed optical ring. Changes that were made include altering the orientation of the slits from vertical to horizontal and increasing rigidity, as the initial optical ring was fragile. We also made some rudimentary PCB modifications such as increasing the space between the header pins, added accelerometer port, an LED for indicating status, serial debugging interface, and a voltage monitoring port. We encountered another challenge with soldering a miniscule quad-flat package with no leads (QFN) accelerometer. In the end, the accelerometer was not functional, so we added a breakout board with a functioning accelerometer to our PCB. Lastly, we had an issue with nonlinear braking, where the user would apply the brakes and the slack in the brake line prevented a gradual application of the brakes. Therefore, this was accounted for with a logarithmic braking algorithm depicted section 2.5.

5.3 Ethical Considerations

There are a few safety hazards that are associated with our project. Lithium-ion batteries for our power supply, which contains microscopic metal particles that may come into contact with other parts of the battery cell, leading to a short circuit within the cell. In unfortunate but likely situations, a thermal runaway can occur as a result of this short circuit -- a process where temperature rises uncontrollably until a violent chemical reaction erupts.

Another safety hazard arose during the testing portion of our project: riding the bike to analyze the functionality of the braking system may contain bugs, risk braking too hard, flipping over and hurting ourselves in a variety of ways. Unfortunately, this particular safety hazard is the most prominent, because the objective is to prevent potential injury, so the threshold of safety will have to be towed very carefully to make sure accurate data is received as well as keep ourselves and others safe from physical harm. In order to address this potential hazard, an emergency mechanical brake was added in parallel to the original brake line so that if ERSAB fails, the cyclist will still be able to brake.

With regards to the IEEE Code of Ethics[1], the first and ninth codes seek to preserve the safety of the public, and if testing our project risks crashing our bike, testing must be conducted in an area that is devoid of people in order to not endanger anyone. After scouring other ethical codes like the University of Illinois' campus safety policy, most ethical guidelines do not pertain to our project except for the aforementioned ones that discuss personal and public safety.

Parallel Braking Configuration

5.4 Future Work

There are a few considerations that would add more functionality to ERSAB given more time to work on it. Instead of relying on a servo-controlled braking system, we would have liked to have developed an algorithm to directly detect when the bike is able to flip forwards or experience fishtailing. Also, with the correct transmit and receive accelerometer inputs, we would be able to have a functional accelerometer that would be able to supply the current degree of inclination of the bicycle. Furthermore, improving our noisy data from our speed data would be sensical improvement. This would be achieved in a few different ways: enlarging our optical rings to the size of the wheels, increasing the size of the slits on the optical ring, and increasing the rigidity of the optical rings. Since we added extra header pins to our PCB, more sensors could be integrated into our design. Most importantly, with the sheer amount of data processing and available electrical connectivity in ERSAB, we have thus built an infrastructure for any algorithm the user desires.

We would also like to work on the algorithm to prevent locking up of the wheels. Due to the noisy data received from the speed sensor we were not quite able to achieve this part. Our plan includes cleaning up the noise from the speed sensor with a tolerance of about 10% . Once that is achieved we can train the system to understand when the wheel is about to lock up by calculating the deceleration of the bike. If the bike decelerates at a higher rate than standard expected value (to be determined), then we can ease off the front and back brake to reduce the deceleration and thus ensuring the safety of the user.

References

[1] IEEE.org. (2019). *IEEE Code of Ethics*. [online] Available at: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed 19 Sep. 2019].

[2] NHTSA.gov. (2018). *Traffic Safety Facts*. [online] Available at: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/812502_pedestrian-and-bicyclist-data-analysis-tsf-research-note.pdf?fbclid=IwAR2haj6i7xpE9nyCyEeqgjgRTxs1GmiGcvj2ojDA53sZ9f4XOfNWzsCEgZk [Accessed 19 Sep. 2019].

[3] "Salary Averages," *Salary Averages*. [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages.asp. [Accessed: 02-Oct-2019].

[4] "Prevailing Wage rates for Champaign County," 01-Sep-2017. [Online]. Available: https://www2.illinois.gov/idol/Laws-Rules/CONMED/Documents/2017 Rates/Champaign.pdf.

[5] "Bike Brake Cable Testing," *Titan Arm*, 27-Jan-2013. [Online]. Available: https://titanarm.com/post/48680663848/bike-brake-cable-testing. [Accessed: 03-Oct-2019].

Appendix A

Requirement and Verification Table

Module	Requirements	Verification	Verification Status (Y or N)
Power Delivery & Storage	1. Able to fully charge the battery within two hours. (similar to many consumer electronics such as smartphones) 2. Has sufficient battery life to fully actuate the brakes at least 500 times, which would be enough for several days of commuting 3. Dimensions are small enough to fit within the frame of the bicycle	1. Fully discharge the battery. Plug it in to charge and record how long it takes to charge with a stopwatch 2. Fully charge the battery. Actuate the brakes fully and repeatedly. Record the number of actuations until it indicates that the battery is discharged. 3. Visual inspection and test ride to ensure all parts of the system are enclosed and do not hang off the bicycle.	Y
Braking System Interface	1. Able to handle an applied force of at least 100 N 2. Able to send a reading to the microcontroller over the serial interface at a rate of at least 10 samples/second. 3. Comfortable to press for an average person.	1. Attach a force sensor in-line with the hand brake. Apply a force until the force sensor reads 100N. Observe that readings from the system sensor continue to increase until the limit. 2. Connect oscilloscope probes to the serial communication lines of the sensor. Observe the periodicity of the signal. If the period between. 3. Ask 3 casual riders to test the brakes. Verify that all of the riders consider the brakes to be easy to use.	Y
Microcontroller	1. Able to calculate velocity and acceleration from the wheel rotation sensor at least	1. Using a logic analyzer, connect to the serial output of the microcontroller. Verify	Y

	at the expected rate of 5000 samples/second.	that at least 5000 samples are returned every second.	
Motor	1. Able to apply a force of at least 100 N across a pull distance of at least 10mm. 2. Actuate the brakes fully within 500 ms.	1. Use another luggage scale to measure the force of braking. 2. Run 10 trials of braking and releasing with a stopwatch and divide by 10 to see how long it takes for an individual trial.	Y
Wheel Rotation Sensor	1. Able to measure the wheel rolling forward with an accuracy of at least 20 mm. 2. Able to measure rotation up to a wheel speed of at least 1500 rpm (equivalent 40mph on the bike)	1. Roll the bike forward by 10 meters. Observe the number of readings from the rotation sensor. Divide 10meters/number of readings and verify that this is less than 20mm. 2. With the bike upside down, spin the wheel at 1500 rpm verifying with a tachometer. Verify that the value on the display matches the tachometer reading.	Y
User Feedback	1. LCD display must show a 100% battery when the lithium-ion battery reaches a full charge of 4.2V/cell. It must show 0% when the battery is fully discharged at 3.4V/cell.	1. Fully charge the battery and confirm cell voltage with a multimeter. Verify value on the display. Discharge the battery until multimeter indicates 3.4V/cell. Verify that the display reads 0%.	Y
Accelerometer	1. Get the gradient of the bike to show if it is going uphill or downhill. 2. Actively measure the x-axis to measure abrupt changes to prevent fishtailing.	1. Go uphill and calibrate the bike using the data from the accelerometer. 2. Calibrate data using bike under normal and slipping conditions.	N