University of Waterloo
Faculty of Engineering
Department of Electrical and Computer Engineering

BSAS (Bicycle Safety Assistance System)

Group: 2020.08

Prepared by
Xi Fan: 20593033
Tiancheng Gao: 20586084
Ran Chen: 20580257
Zaoli Zhang: 20553143

Consultant:
Daniel Davison

February 13, 2020

Abstract

Most people can notice the speed limit of the automobile, but very few can realize there is also a speed limit for bicycles. Many countries regulate the speed limit of cycling under 15 mph considering the potential injuries due to over-speed riding. This project is intended to design an integrated system built on bicycle brakes, and it is supposed to provide the speed control to the bike as the bike speeding up while going down a ramp or unintentional high-speed riding. This system can prevent potential accidents and injuries from happening by limiting the speed, which will leave the rider with more time of reaction as unexpected obstacles come up. The system mainly consists of a microcontroller, a motor, and a speed sensor. The data from the speed sensor is processed as an input by the microcontroller and then fed to the motor, and the motor controls the brake system based on the embedded algorithm. The BSAS is developed with a variety of theories, including control theory, embedded system design and mechanical knowledge. The main advantage of the BSAS is to prevent over-speed riding and ensures that the driver is always under protection.

Acknowledgements

We would like to thank Professor Daniel Davison for being our project consultant and his helpful advice in guiding the design of the project. Professor Davison also gave us valuable feedback and helpful suggestions on our control system design.

We would also like to thank Professor Sagar Naik, David Bell, Paul Ludwig, and Bill Jolley for their work for the course ECE 498A and ECE 498B.

We also confirm that this report has not been previously submitted for academic credit at this or any other academic institution.

Contents

List of Figures

List of Tables

1 High-Level Description of Project

1.1 Motivation

Most people can notice the speed limit of the automobile, but very few can realize there is also a speed limit for bicycles. In 2014, a 31-year-old cyclist in Marin County struck a 9-year-old child because of overspeed. Due to the accident, the child suffered from severe head and internal injuries [1]. The incident captured the attention of many people about the speed limit of bike. In fact, many countries regulate the speed limit of cycling under 15 mph (24km/h) or 20 km/h. In Calgary, overspeed cyclist can be fined for $50 [2]. Considering the potential speed tickets and injuries due to over-speed riding, a project related to the bicycle brake assistance system was proposed.

1.2 Project Objective

The objective of the project is to design an integrated system built on bicycle brakes, and it is supposed to provide the speed control to the bike as the bike speeding up while going down a ramp or unintentional high-speed riding. This system can prevent potential accidents and injuries from happening by limiting the speed, which leave the rider with more time of reaction as unexpected obstacles come up.

1.3 Block Diagram

As shown in Figure 1 below, the system mainly consists of a front panel and a back-wheel control system. To be more specific, the front panel includes a displayer and several buttons, which allows the user to interact with the system. The back-wheel control system consists of a power supply system, a microcontroller, a motor, a brake and a sensor. The power supply system supplies the power for all electronic devices. Once the power supply is turned on, the speed data from the sensor is processed as an input by the microcontroller and then fed to the motor, and the motor controls the brake system based on the embedded algorithm. The BSAS is developed with a variety of theories, including control theory, embedded system design and mechanical knowledge.

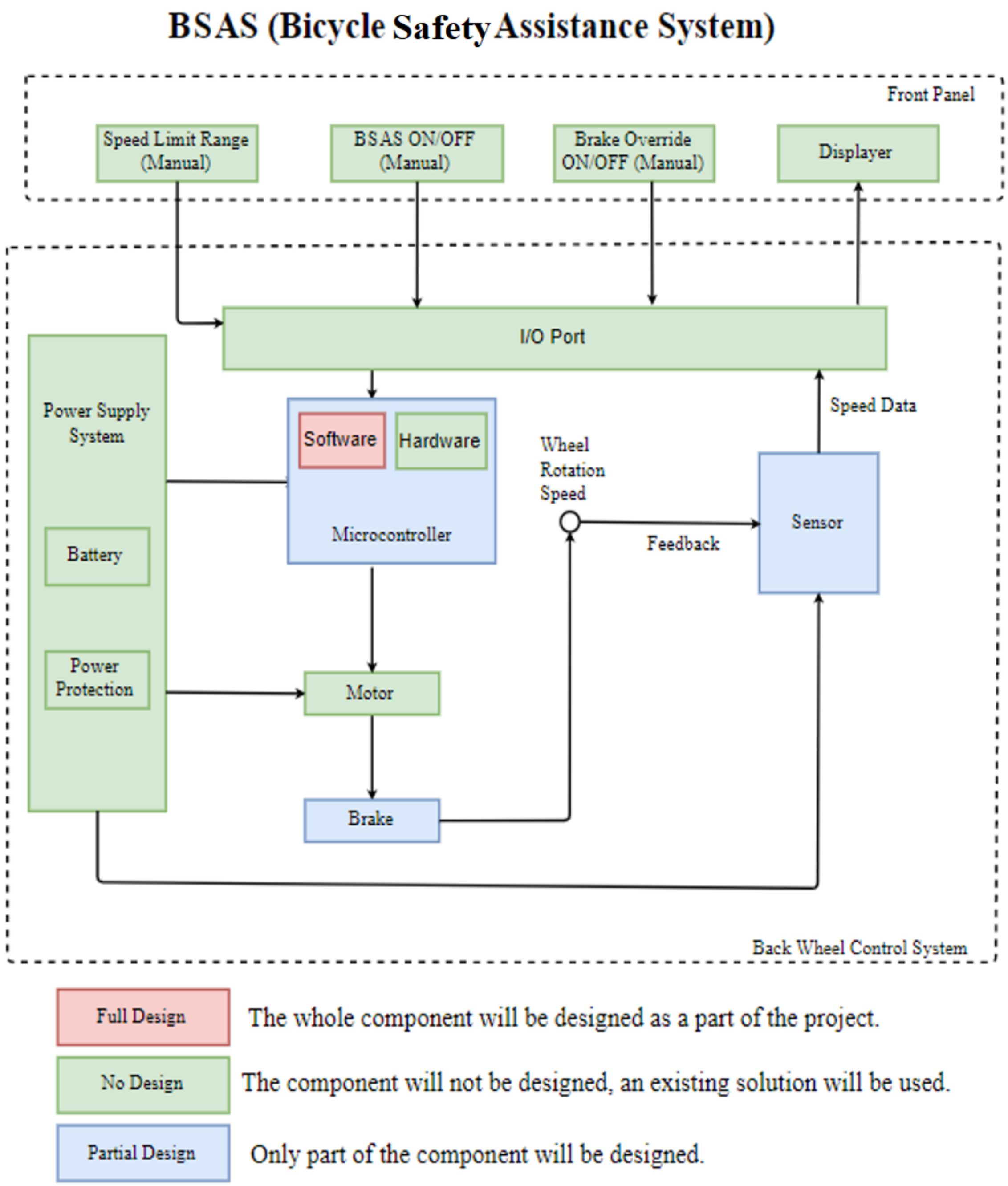

Figure 1: High-Level Block Diagram

Each subsystem is described in detail in the following Table 1:

Table 1: Description of Block Diagram

Subsystem	**To be designed**	**Description**
Speed Limit Range	No	Allows user to set the speed limit range. Usually, it is under 25 km/h
Speed Limit ON/OFF	No	Allows user to turn on or off the BSAS system.
Brake Override ON/OFF	No	Allows user to stop the bicycle immediately. This is a backup system for the brake lever of the bike.
Displayer	No	Allows user to read the current speed.
I/O system	No	An interface between input/output information and the microcontroller.
Power Supply	No	Provides power for microcontroller, sensors and motor. The power supply system contains two batteries: one battery for the motor and another battery for other devices. The motor requires a separate battery as a power supply protection so the current cannot flow back to other equipment.
Microcontroller	Partial Design	Used to take the input data to control the motor based on the embedded algorithm. The hardware part of the microcontroller is a non-design component, but the algorithm is developed in this project.
Motor	No	Used to pull the bike brake cable to reduce the speed.
Brake	Partial Design	Cooperate with the motor to reduce the speed.
Sensor	Partial Design	Used to detect the current speed of the bike.

According to Figure 1, there is a node named wheel rotation speed. This is a state that represents the real running speed of the bike, which can be considered as the plant in the control theory.

2 Project Specification

2.1 Functional Specification

The following Table 2 shows all the functional specifications involved in this project.

Table 2: Functional Specification

Subsystem	**Specification**	**Classification**	**Description**
BSAS ON/OFF	Speed Control Switch	Essential	The entire speed control system can be switched ON and OFF using one of the buttons on the front panel.
Speed Limit Range	Speed Limitation Function	Essential	Speed limit range should be adjustable via the pushbuttons.
Brake Override ON/OFF	Override Function	Non-Essential	With the override button pressed, the override function should take place. The motor should start tightening up the brake cable and slow the bicycle down to stop.
Displayer	Speed Display	Essential	The speed and speed limitation displayed on the screen needs to be updated every 1 second.
Microcontroller	Feedback function	Essential	The microcontroller must be able to adjust the bicycle speed based on feedback from the speed sensor.
	Data transmit and Receive	Essential	The microcontroller must be able to receive the speed data from the sensor, send the override request from users to the motor, and send current bicycle speed to the front panel.
Motor	Brake Function	Essential	As the running speed exceeds the pre-set limit, within 0.2 seconds delay, the motor should start tightening up the brake cable and slow the bicycle down to the expected speed range.

2.2 Non-functional Specification

The following Table 3 shows all the non-functional specifications involved in this project.

Table 3: Non-Functional Specification

Subsystem	Specification	Classification	Description
Overall System	Overall Cost	Essential	The overall cost of the system must not exceed $1000.
	Water Resistance	Non-Essential	The system is anticipated to operate under various riding conditions, including rainy days.
	Operating Temperature	Non-Essential	The system should be able to operate in a temperature range from -10 °C to 50 °C to accommodate different riding needs.
Power Supply	Battery Protection	Essential	Power protection techniques must be used while delivering power to the motor to avoid potential system damages
	Battery Life	Essential	The battery must be able to support at least 2 hours of regular operation of the entire system.
	Power Supply Dimensions	Essential	The size of the power supply set must not exceed 25 cm*25 cm*25 cm to ensures easy installation
Front Panel	Display Dimensions	Essential	The front panel size must not exceed 10 cm*10 cm to avoid eyesight blocking
	Front Panel Button	Essential	The front panel must contain 4 push buttons, which provides accessibility to 3 main control inputs, namely, speed range setting, speed limit ON/OFF and brake override ON/OFF.
Motor	Motor Dimensions	Essential	The motor size must not exceed 25 cm*25 cm*25 cm to ensures easy installation
	Motor Strength	Essential	The motor must be able to provide enough torque in order to drive the brake wire to stop the bike.
Overall System	Steady-state Accuracy	Essential	The system must control the bicycle speed to a steady state value within 20% difference compared to the desired rate.
	Settling time	Essential	The settling time of the system should not exceed 5 seconds, which means the output speed of the system should enter and remain in the desired speed range within 5 seconds.

3. Detailed Design

3.1 Subsystem Front Panel Design

The front panel, including the BSAS ON/OFF button, the brake override ON/OFF button, the speed limit range + button, the speed limit range – button and the displayer, is an interface subsystem of BSAS which allows the user to interact with the system. There are two solutions, touch screen and LCD (Liquid-Crystal Display) Keypad shield, that can achieve the gold of interacting with users.

3.1.1 Touch Screen

Touch Screen, usually TFT (Thin Film Transistor) touch screen, are widely used for Arduino applications. The TFT touch screen is a combination device that includes TFT LCD and a touch technology overlay on the screen, and can both display content and act as an interface device for the BSAS. The button and display content are customized by programming, which requires higher performance for the processing unit. Also, the touch screen is operated at 5V, 300mA [3], so the power consumption can be calculated by using the following formula:

$$P=V*I \qquad \text{(Equation 1)}$$

Where

P is the power in Watt

V is the voltage in V

I is the current in A

Therefore:

$$P=5V*300mA=1.5W$$

3.1.2 LCD Keypad shield

The LCD keypad shield is a basic shield which features an LCD module and six buttons. The buttons can be used for "speed limit range +", "speed limit range -", "BSAS ON/OFF" and "brake override ON/OFF". At the same time, the displayer is used to indicate the current speed and the pre-set speed limitation in km/h. The input from the button can be simply read by creating a read_LCD_button function, and the output to the displayer is implemented by using lcd.print(). The LCD keypad shield operates during 4.5V to 5.5V, 0.6mA [4], so the maximum power consumption is as following by using Equation 1:

$$P=5.5V*0.6mA=3.3*10^{-3}W$$

3.1.3 Decision Matrix on Front Panel Selection

Based on the above analysis, a decision matrix is created to find the best design for front panel subsystem. As shown in table 4, the scale 0-1 for each product are applied based on their complexity of implementation, flexibility, the cost and the power consumption.

Table 4: Decision Matrix

Criteria	Weight	Touch Screen		LCD Keypad shield	
Complexity	25%		0.5		1
Flexibility	25%		1		0
Cost	25%	\$13	0.92	\$12	1
Power Consumption	25%	1.5W	0.0022	$3.3*10^{-3}$W	1
Total Mark	100%	0.61		0.75	

As shown in table 4, the best solution is to use the LCD keypad shield, as its total score is higher. The complexity of the LCD keypad shield is given the grade of 1 since it is easier to implement in the programming aspect. The flexibility of the touch screen is 1 because the button and the display content can be customized. The score of cost and power consumption are given based on the price and the spec sheet of the products. As a result, LCD Keypad shield is a better choice since it is less complicated, cheaper and power efficient.

3.2 Subsystem Sensor Design

Considering the cost, the size, and the compatibility with Arduino, the laser transmitter and receiver are determined to use as the speed sensor. Moreover, three possible approaches to speed measurement are discussed and compared in the following section.

The first approach uses a pair of the Arduino laser transmitter and laser sensor receiver module. The laser transmitter and receiver can be used as a speed sensor via measurement and mathematical calculations while ensuring all the specifications of the project are satisfied.

As shown in Figure 2, The laser transmitter labelled in red and receiver labelled in yellow are fixed and aligned on the opposite seat stays, and a long curved black plastic slice labelled in black is placed on the spokes of the back wheel and rotating with the tire. The black plastic slice should be long enough to cover 5 to 8 spokes for the sake of high accuracy. The transmitter continuously sends infrared ray to the receiver. The receiver activates the high mode and sends the high signal to the Arduino while receiver receives the infrared ray, deactivates low mode and sends a low signal while the infrared ray is blocked by the black plastic slice. When the black plastic slice passes through the gap between the transmitter and receiver, the infrared ray would be blocked during this period, the receiver works at low mode and continuously sends sufficient low signals to the Arduino. Although it is possible that the bicycle spokes block the infrared ray while the transmitter continuously sends infrared ray to the receiver diode, the low signal caused by spokes is discontinuous and insufficient. An if statement on software algorithm is used to filter the spoke blocking effect.

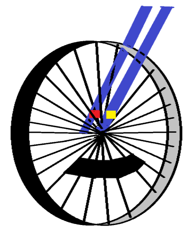

Figure 2: First Approach of Speed Sensor

When the bicycle moves, the Arduino can detect sufficient continuous low signals. T1 is the time when the first continuous low signal string ends, T2 is the time when the second continuous low signal string ends, and T_n is the time when the n-th continuous low signal string ends. In addition, the perimeter of the wheel can be calculated using the following equation:

$$L = \pi * D \qquad \text{(Equation 2)}$$

Where

L is the perimeter of the bicycle tire

D is the outer diameter of the bicycle tire, which is 67 centimeters in the BSAS project.

Therefore, the current speed of the bicycle can be calculated according to the following equation:

$$V = \frac{L}{\Delta T} \qquad \text{(Equation 3)}$$

Where

V is the speed of the bicycle

L is the perimeter of the bicycle tire

ΔT is the time difference between two continuous low signal string, calculated by $T_n - T_{n-1}$

The second approach is a theoretically advanced version of the first approach, and it uses two pairs of laser transmitters and receivers to build two speed sensors. As shown in Figure 3, the two pairs of Laser transmitters and receivers are fixed and aligned on the additional external metal bar labelled in grey which soldered on the opposite seat stays. Two Laser transmitters labeled in red are placed on one metal bar and two Laser receivers labeled in yellow are fixed on another metal bar.

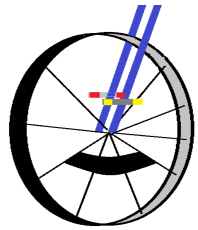

Figure 3: Second Approach of Speed Sensor

The distance between these two pairs of transmitter and receiver is much smaller than the minimum distance of two spokes (5 centimeters) and much larger than the spoke diameter (0.5 centimeters). There are four types of possible results of these two speed sensors, as shown in Table 5 below:

Table 5: Truth Table of The Sensor

	Result from Sensor A	**Result from Sensor B**	**Meaning of the Result**
Case A	Receive Infrared Ray	Receive Infrared Ray	None of the sensor is blocked.
Case B	Receive Infrared Ray	No Infrared Ray	Sensor B is blocked by a spoke.
Case C	No Infrared Ray	Receive Infrared Ray	Sensor A is blocked by a spoke.
Case D	No Infrared Ray	No Infrared Ray	All sensors are blocked by the slice.

Therefore, the spoke blocking effect can be eliminated because the spoke cannot block both sensors at the same time, and case D only occur when these two pairs of sensors are blocked by the black plastic slice. Similar to approach one, the perimeter of the wheel can be easily calculated by equation 4. T1 is the time when the first case D low signal string ends, T2 is the time when the case D low signal string ends, and Tn is the time when the n-th case D low signal string ends. The speed of the bicycle can be calculated by equation 4:

$$V = \frac{L}{\Delta T} \qquad \text{(Equation 4)}$$

Where

V is the speed of the bicycle

L is the perimeter of the bicycle tire

ΔT is the time difference between two continuous case D signal string, calculated by T_n - T_{n-1}

The third approach has the same principle to calculate the speed of the bicycle. However, a Hall effect magnetic sensor and a magnet are used instead of a laser transmitter-receiver pair and a black plastic slice. The magnet is placed at the location of the plastic slice and the magnetic sensor is placed at the location of the laser transmitter or receiver. When the bicycle wheel rotates, each time the magnet passing by the magnetic sensor, the sensor detects the magnetic field change and sends out a signal, just like what the laser receiver does. The time that the signal is detected can be used to calculate the bicycle's moving speed, as what is done in the first approach.

As shown in Table 6, the third approach is the final choice for the project as it has the highest total score. Compare these three approaches, the third approach has the lowest cost, and lowest power consumption, which is chosen to measure the bicycle speed for the project.

Table 6: Decision Matrix of Sensor Implementation

	Cost	**Complexity**	**Power Consumption**	**Feasibility**	**Total**
Weight	25%	25%	25%	25%	100%
Approach one	10	8	8	8	8.5
Approach two	5	6	6	6	5.75
Approach three	10	7	10	9	9

3.3 Subsystem Motor and Brake Design

The purpose of having a motor in this project is to use it as a source of torque. In other words, it is acting as the engine of the system, to provide a program-wise controllable force to the bike braking system. To achieve the goal of running bike speed manipulation, a precise rotation angle of the motor and the timing of the rotation are the keys. The motor and brake system mainly consist of five parts, namely motor itself, gearbox, brake cable and brake caliper. The brake cable and brake caliper are coming from the existing bike braking system. In this project, the brake cable is attached to a motor shaft via a mounting hub and a pulley. The mounting hub is basically a piece of metal which has some screw holes across from top to bottom. The center hole is designed to work with motor shafts. The mounting hub can be fixed to the motor shaft by having the shaft inserted into the center hole and tighten up the side screw. Various types of modules can be attached to the hub using screws to serve different purposes. For example, gears, fans, servo pulleys and servo horns are all valid module to attach. In this project, a servo pulley is used and stacked with the mounting hub since the goal is to transfer the motor torque to a force which can pull the brake cable. Below is a figure with demonstrating the dimension and appearance of a typical servo pulley.

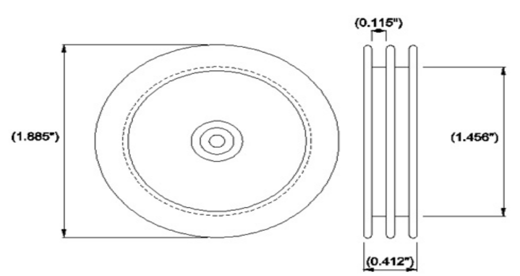

Figure 4: Servo winch pulleys [5]

Servo winch pulleys are used to gather string and are commonly found on R/C sailboats. A demonstration of where to put the motor and how it connects to the pulley is shown below.

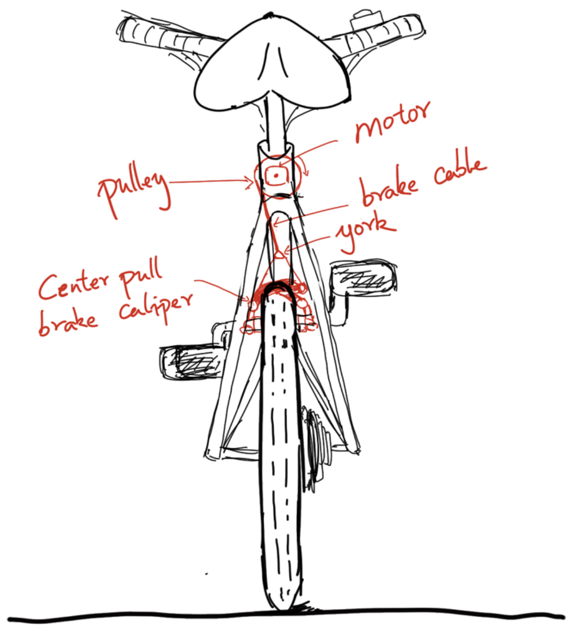

Figure 5: Detailed Motor and Brake Design

A motor is mounted under the bike set with the shaft pointing horizontally towards the back wheel. A pulley, which has brake cable securely wind around, is attached to the motor shaft via the mounting hub. The brake cable is routed through a yoke, and in which the cable gets separated into two yoke cables. The brake cable is then connected to the caliper which has two brake pads on. As the tension in the brake cable increases, the caliper gets pulled in both directions around the pivot point, which creates a motion moving the brake pads towards the center and touching the rim. The friction between the rim and the pad causes the bike to slow down.

Therefore, once the motor shaft and brake cables are successfully bonded, the next step is simply to apply adequate amount voltage to the motor so that it drives the cable to pull the brake caliper. In order to know how much voltage and torque is needed for each specific case, knowing the minimum torque required to fully stop the bike can be helpful when choosing which type of motor to use. According to the result of research, the typical braking force needed to slow a bike with a deceleration of 1m/s^2 is roughly 50-Newton. The measurement assumes that the total weight is 90 kg, including a driver weight of 75 kg [6]. Moreover, this measurement is taken by applying the force to the rear wheel only. This is a reasonable assumption for this project as well since the project is intended to control the speed smoothly and never about suddenly stopping the bike. Therefore, it is very unlikely that the force needed from the motor in this project exceed 50-Newton. Thus, with the force specified, the minimum torque required from the motor can be calculated using the following equation:

$$\tau = F \cdot r \sin(\theta) \qquad \text{(Equation 5)}$$

where

τ is the torque vector

F is the force vector

r is the length of the moment arm, which is essentially the radius of the pulley given in the previous section

θ is the angle between the force vector and the moment arm. This is approximately 90 degree since the project is aiming to have the brake cable tangentially pulled by the pulley.

Thus, by using each value:

$$\tau = 50\ N * (1.885/2)\ inch * 0.0254\ meter/inch * \sin(90)$$
$$= 1.1969\ newton\text{-}meter$$
$$= 12.20\ kg\text{-}cm$$

Various kinds of motors can be considered after fixing the torque requirement. For instances, simple DC motor, stepper motor and servo motor are all valid solutions. The following table compares each type of motor by listing out their pros and cons.

Table 7: Comparison Between Different Types of Motors

Motor Types	Simple DC Motor	Stepper Motor	Servo Motor
Cost	Lower, roughly 10 -30 CAD	Lower, roughly 10 -30 CAD	Slightly Higher, around 50 CAD
Size	Wide range of size choices	Stepper motor does not have as many size selections compared to the other two, mostly small	Wide range of size choices
Overload Safety/ Reliability	Unlikely to be damaged if overload protection is implemented correctly	Stepper motor is unlikely to be damaged by mechanical overload	Servo motor may malfunction if overloaded mechanically
Efficiency	Depending on types	Not as efficient, consuming a lot of power, and yields roughly 60% to 70% efficiency	Efficient, yielding 80% to 90% efficiency
Torque	High torque	Low torque	High torque
Rotation Precision	No control of rotation position unless a specific control circuit has to be implemented	High	High

Compatibility	Hard to set up, everything has to be designed from scratch. For example, gear ratio, feedback loop, microcontroller interfacing.	Plug-and-play	Easy to set up but still requires some effort, existing libraries working on Arduino. Built-in gearbox and feedback loop

It turns out that the servo motor is the best fit for this project for the following reasons. Firstly, it comes with high torque and provides best rotation precision without needing any additional circuitry or gear ratio design. This is also the most important aspect of the design. A typical high torque servo motor can provide torque for approximately 32.4 kg*cm to 53.1 kg*cm, and it has the dimension of 43x32x32.5 mm^3 [7]. The project specification states that the motor dimension has to be less than 25x25x25 cm^2, and based on calculation, the motor has to be able to provide at least 12.2 kg*cm torque. Therefore, the servo motor chosen is certainly satisfying the project non-functional specifications. Moreover, it has the best compatibility with Arduino since existing libraries and sample codes can be found easily. The operating frequency of the motor is 1520us/333Hz, thus, it can respond to the command from Arduino within 0.2s. This suggests that the brake function specification is also satisfied. Furthermore, according to the non-functional specification in the previous section, the system is supposed to be able to run continuously for at least two hours. The efficiency of the servo motor is relatively high, which minimizes energy wastes of the system. Therefore, all the specifications can be met by using the servo motor.

3.4 Subsystem Microcontroller Design

3.4.1 Arduino with front panel

Four components are involved in the front panel design: the LCD displayer design, the speed limit range design of the bicycle, the BSAS ON/OFF design of the inner function, and the Brake Override ON/OFF design of the inner function. The first component, the LCD design, requires output from the Arduino, while the last three components, which are the speed limit range design, the BSAS ON/OFF design, and the Brake Override ON/OFF design, provide input to the Arduino. Before coding the input and output, the LCD screen first needs to be set up by including the corresponding library and coding in the Arduino-given setup() function:

```
void setup() {
    limit = 24; // set the speed limit initially to be 24 km/h
    lcd.begin(cl, ln); // start the LCD with ln+1 lines and cl characters of capacity in each line
    lcd.setCursor(0, 0);
    lcd.print(“BSAS”);
    lcd.setCursor(1, 0); // set cursor to (0, 0) position
```

```
    lcd.print(“Limit: ”); // print message
    lcd.setCursor(2,0); // set cursor to (1, 0) position
    lcd.print(“Speed: ”); //print message
}
```

For the LCD design, two lines of information are displayed on the LCD screen, which is the speed limit setting of the BSAS and the current speed of the bicycle. The message “Limit: ” and “Speed: ” is printed in the setup() function. The numbers are dynamically printed in the Arduino-given loop() function to keep those values up-to-date:

```
void loop() {
    lcd.setCursor(1, 7); // set cursor to (1, 7) position after “Limit: ”
    lcd.print(limit); // print value
    lcd.setCursor(2, 7); // set cursor to (2, 7) position after “Speed: ”
    lcd.print(speed): // print value
}
```

The speed limit range design, the BSAS ON/OFF design, and the Brake Override ON/OFF design are triggered by buttons, which provide input to the Arduino. The speed limit range design controls the speed limit value and involves two buttons, which are speed-up and speed-down. The BSAS ON/OFF design involves one button and is used to turn the functionality of the BSAS on and off. The Brake Override ON/OFF design involves one button and works in the way such that the bicycle brakes to stop once the button is pushed. The task of hearing the button activities is also conducted in the given loop() function, which makes it a polling process that can consistently monitor the button states.

```
void loop() {
    lcd_key = read_LCD_buttons();
    switch (lcd_key)
        case speedUp:
            if (limit < 30) -> limit++;
        case speedDown:
            if (limit > 20) -> limit--;
        case BSASOnOff:
            if (BSAS is on) -> turn BSAS off;
            if (BSAS is off) -> turn BSAS on;
        case brakeOverride:
            limit = 0;
```

```
}
```

3.4.2 Arduino with the speed sensor

As the previous section discusses, the speed sensor functions based on the use of a pair of laser sensors. The laser transmitter is connected to a 5V Arduino pin to be powered, and the laser receiver is connected to an Arduino GPIO pins, for example, pin 2. The laser sensor pair is coded in the following way to set up:

```
#define DETECT 2
int counter = 0;
double time_start = -1;
double time_end = -1;
void setup() {
    Serial.begin(9600); // initialize the serial connection at 9600 bps
    pinMode(DETECT, INPUT); // define DETECT (pin 2) to be an input pin
}
```

The laser receiver receives laser from the transmitter and keeps sending 1's (logical high) to the Arduino. When the laser is blocked, 0's (logical low) are sent. When the black plastic slice blocks the laser, since the blocking time is short enough, only a single 0 is sent in the form such as "…1110111…". When the laser is blocked by the black plastic slice, since it is blocked for a period of time, a series of 0's (no less than three) are sent in the form such as "…1110000111…". Each time a series of 0's is received, the algorithm calculates the time span from the last time when a series of 0's being received. The time is treated as the time it takes for the wheel to rotate a full rotation. The pseudocode is shown below:

```
void loop() {
    int t = 0;
    int detected = digitalRead(DETECT); // take the input from DETECT (pin 2)
    if (detected == HIGH)  -> counter++;
    else {
        if (counter >=2) {
            if (time_start != -1) {
                set time_end to be current time;
                t = time_start – time_end; // the time it takes for to do a full rotation
            }
                set time_start to be current time;
        }
```

```
        counter = 0;
    }
    If (t != 0) -> speed = perimeter of the wheel / t;
}
```

3.4.3 Arduino with the motor

A servo motor is used in this project to function the brake system. It is connected to an external 5V power supply and the Arduino by a GND pin, and a control pin. Only the control pin requires coding to set up. Let the control pin be connected to pin 9 of the Arduino:

```
#define servoPin 9
void setup() {
    pinMode(servoPin, OUTPUT); // define servoPin (pin 9) to be an output pin
}
```

The control signal is a series of pulses with 50Hz frequency. The width of each pulse determines the angular position of the servo motor. The pulse width is controlled using the Arduino given delayMicroseconds() function, as shown below.

```
void loop() {
    digitalWrite(servoPin, HIGH);
    delayMicroseconds (1500); // 1.5ms pulse width refers to 90 degree angular position
    digitalWrite(servoPin, LOW);
    delayMicroseconds (18500); // 18.5ms + 1.5ms = 20ms which is 50Hz
}
```

3.4.4 Control System Design

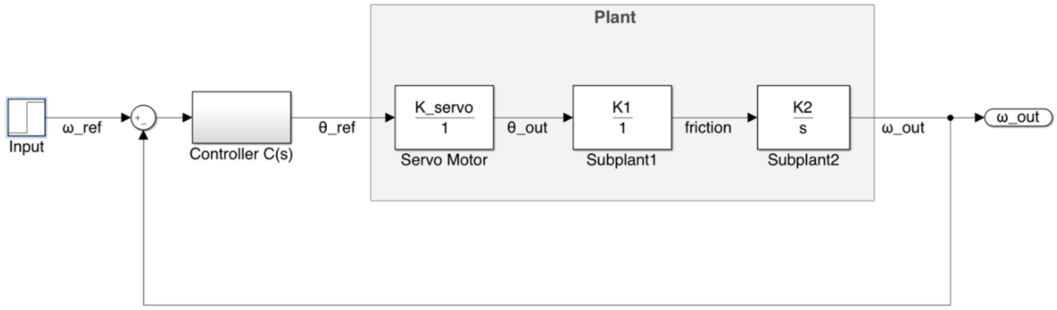

Figure 6: Block Diagram of Control System

The block diagram above depicts the BSAS from the perspective of a control system. The plant is composed of three subsystems: the servo motor, the "Center Pull Brake Caliper" (subplant1) that is described in the previous section, and the bicycle wheel (subplant2) that the brake system functions on.

The servo motor, as the previous section describes, is a linear system that controls the angular position of the motor shaft. As a result, if the input is θ_ref and the output is θ_out, the transfer function should be a constant K_servo.

The subplant1, which is the "Center Pull Brake Caliper" system, takes the angular position of the motor shaft θ_out as the input and transform that into a pair of pushing forces F_{push} over the bicycle wheel through the brake pads. Since the components of the "Center Pull Brake Caliper" system, such as the servo winch pulley and the yoke cables, can be modelled with linear relationships between their inputs and outputs, it is reasonable to model the whole thing with a linear relationship between θ_out and F_{push}. Also, since $F_{friction} = \mu F_{push}$, where $F_{friction}$ is the friction that the bicycle wheel encounters due to the pushing force from the pair of braking pads and μ is called the friction coefficient which is a constant for a specific bicycle wheel, there is a linear relationship between θ_out and $F_{friction}$. As a result, the transfer function of the subplant1 is a constant K_1.

The subplant2, a bicycle wheel, has two torques summing on it: $\tau_{friction} = -r_{wheel}F_{friction}$ that is due to the braking friction and τ_{driver} that is due to a driver stepping on the pedals and is treated as the disturbance to the control system. The disturbance can be theatrically ignored such that $\tau_{driver} = 0$ since stepping on the pedals usually does not have any effect when a bicycle runs in a high speed, and a driver normally does not slow down the bike and step on the pedals at the same time. As a result, the total torque on the bike wheel is $\tau_{total} = \tau_{friction} + \tau_{driver} = \tau_{friction}$. The relationship between $F_{friction}$ and the angular speed ω can be bridged by the angular momentum $L = rmv$ and the moment of inertia $I = \frac{L}{\omega}$, such that

$F_{friction} = -\frac{\tau_{total}}{r} = -\frac{rF_{total}}{r} = -\frac{rma}{r} = -\frac{rm\dot{v}}{r} = -\frac{\dot{L}}{r} = -\frac{I\dot{\omega}}{r}$. By doing Laplace transform to both sides, the transfer function from $F_{friction}(s)$ to $\Omega(s)$ is calculated as $\frac{\Omega(s)}{F_{friction}(s)} = -\frac{r}{I} \cdot \frac{1}{s} = \frac{K_2}{s}$.

Thus, the whole plant can be modelled with a transfer function $\frac{\Omega\_out(s)}{\Theta\_ref(s)} = \frac{K\_servo \cdot K_1 \cdot K_2}{s} = \frac{K_{plant}}{s}$. Since it has one pole at $s = 0$, which is not in the open-left-half plane, the plant as a system is not stable.

To stabilize the control system, a proportional (P) control can be added to the closed-loop system. By setting the controller $C(s) = K_c$, the transfer function can be derived as $T_{ry}(s) = \frac{PC}{1+PC} = \frac{K_c * \frac{K_{plant}}{s}}{1+K_c * \frac{K_{plant}}{s}} = \frac{K_c * K_{plant}}{S + K_c * K_{plant}}$. With the transfer function, one single pole of the closed-loop system can be found at s = -$K_c * K_{plant}$. Since K_{plant}=$-\frac{r}{I}$, it is essentially a negative gain to the system. Therefore, to shift the pole to the OLHP (open left half plan), -$K_c * K_{plant}$ need to be less than zero, which means K_c has to be positive to stabilize the system.

3.5 Subsystem Power Supply Design

To implement the power supply system, two steps, circuits design and battery selection, need to be taken. Firstly, the circuit design is based on the input voltage and output voltage (if applicable) of every component, as shown in table 8.

Table 8: Input/output Voltage

Device	input voltage	Output voltage
Arduino	10V (in range 7V to 12V)	5V
Sensor	5V	
Servo Motor	7.4V to 12V	
Front Panel	4.5V to 5.5V	

According to Table 8 and the block diagram, the circuits are designed, as shown in figure 7. The power supply 1 is a 10V battery that can actuate the Arduino. The output voltage of Arduino is 5V, which can be connected to the sensor and front panel as a source of power. The power supply 2 is a 7V battery connected to the servo motor driver as an offset voltage of the motor. At the same time, the voltage supply from Arduino to servo motor is adjustable from 0V to 5V by using a driver. Therefore, the power source of the servo motor is 7V to 12V.

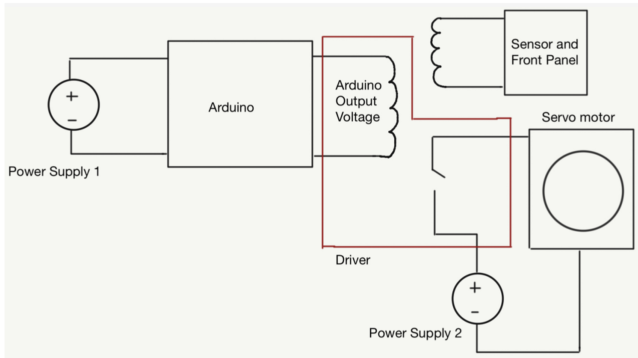

Figure 7: Circuits Design

Secondly, batteries are selected depending on their capacity and voltage. The capacity can be calculated by using table 9 combined equation 6.

$$C=\frac{T*I}{0.7} \qquad \text{(Equation 6)}$$

Where

C is the battery capacity in mA*h,

T is the expected battery life in h, I is the load current in mA

0.7 is the allowances for external factors which can affect battery life

Table 9: Capacity Calculation

	Expected battery life T in h	Device	Current I in mA	Maximum Battery Capacity in mA*h ($C=\frac{T*I}{0.7}$)
Power supply 1	2	Arduino	50	≈6000mA*h
		Sensor	30	
		Front Panel	0.6	
		Servo motor	100 to 2000	
Power supply 2	2	Servo motor	100 to 2000	≈6000mA*h

Noticeable, the servo motor is counted twice when the capacity is calculated. The reason is that when the maximum battery capacity is calculated, another voltage source should be disconnected and all power of the circuits need to be drawn from a single power supply, which is the worst-case condition.

In conclusion, the 10V 6000mA*h battery is selected as power supply 1, and the 7V 6000mA*h battery is selected as power supply 2.

4. Prototype Data

4.1 Prototype Installation

Figure 8 shows how the subsystems are connected. The front panel, including the Arduino and the LCD, is enclosed by a bike handlebar bag and fixed in the stem. This subsystem is the interactive system that takes input and sends output to other components. To be more specific, on the one hand, the Arduino gains the speed limitation from users and calculates the current speed. On the other hand, the microcontroller controls the motor to adjust the speed of the bike. The servo motor is connected to the Arduino so the rotation degree of the servo horns, which is connected to the bike brake cable, can be manipulated. The Hall effect sensor is installed in the seat stay, and it can be used to measure the speed of the bike by collaborating with the magnet in the back wheel.

Figure 8: System Installation

Notable, all components of the BSAS are enclosed in waterproof cases, so the system is able to run under unfavorable weather conditions.

4.2 Experimental Data

To evaluate the accuracy of the system, many tests are conducted to gain the response of the BSAS. The following Table 10 shows the experimental data, and these experiments are taken when going downhill.

Table 10: Experimental Data

time (s)	Rider Weight: 59 KG Test1	Test2	Test3	Test4	Test5	Test6	Rider Wight: 70 KG Test1	Test2	Test3	Test4	Test5	Test6
0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	2	1	2	2	2	3
2	4	2	2	1	2	2	3	3	5	6	5	3
3	6	4	3	3	4	3	7	7	10	6	8	7
4	7	7	6	6	8	6	9	11	14	12	13	12
5	11	10	8	9	10	12	15	15	18	17	13	15
6	14	14	12	13	13	18	20	19	20	20	17	20
7	19	19	17	17	16	20	22	21	22	20	20	21
8	22	21	22	20	20	21	24	22	23	20	21	22
9	24	22	24	23	24	22	25	23	23	23	24	23
10	22	21	21	22	23	22	24	22	22	22	23	22
11	20	21	20	22	23	21	23	22	22	22	22	21
12	19	21	20	21	21	21	22	21	21	22	21	20
13	19	20	21	21	20	21	20	21	21	21	20	19
14	19	21	20	20	20	20	20	21	20	21	20	19
15	20	21	21	21	20	19	20	20	19	20	19	18

time (s)	Rider Weight: 75 KG Test1	Test2	Test3	Test4	Test5	Test6	Rider Weight: 80 KG Test1	Test2	Test3	Test4	Test5	Test6
0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	3	2	4	2	1	3	3	3	4	4
2	3	4	4	6	6	5	3	6	7	6	6	8
3	5	5	6	10	7	6	5	8	9	8	9	10
4	7	10	10	14	12	9	9	8	10	10	12	12
5	12	14	13	17	16	12	13	9	11	11	14	14
6	16	17	19	19	22	17	16	10	14	15	16	16
7	20	20	20	20	23	20	19	15	16	19	17	19
8	23	21	22	23	23	24	21	20	21	20	22	21
9	25	24	24	26	24	27	24	22	23	22	23	23
10	23	23	23	25	23	25	23	21	22	21	22	22
11	21	22	21	24	22	23	22	20	20	20	20	21
12	20	22	20	22	22	23	21	20	20	20	19	20
13	20	21	20	21	21	22	20	19	20	18	19	20
14	20	21	19	21	21	21	20	19	19	18	18	20
15	20	21	19	20	21	21	19	18	19	18	17	19

The tests start when the riders are in flats, and the start time is set to be 0s. Riders are required to reach the maximum speed of the trial when t=9s. At this time, the rider is going downhill. After t=9s, riders are not allowed to pedal, and the bike starts to reduce the speed due to the BSAS. Figure 9 to 12 shows the time response of the BASA system for different riders.

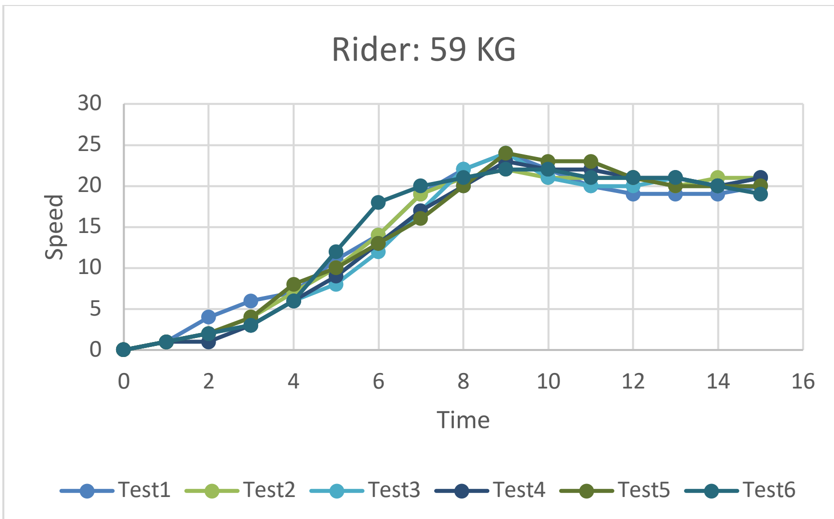

Figure 9: Time Response for Rider 1

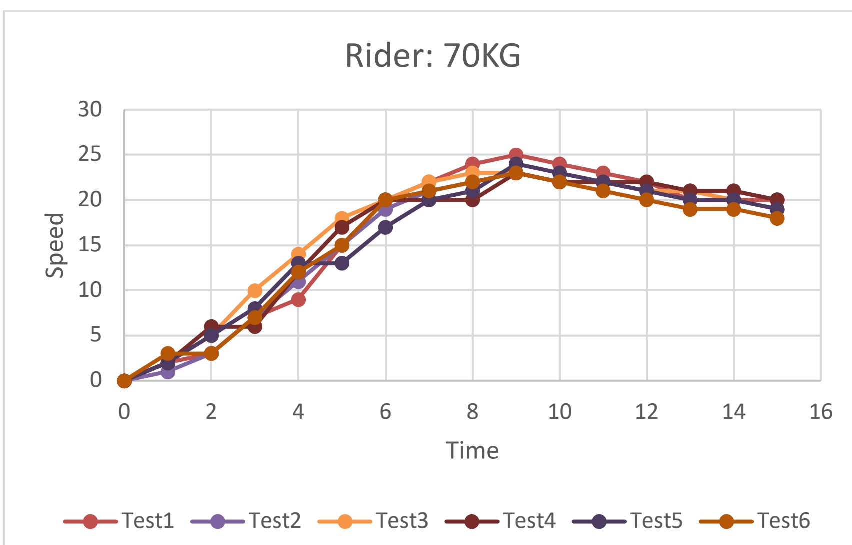

Figure 10: Time Response for Rider 2

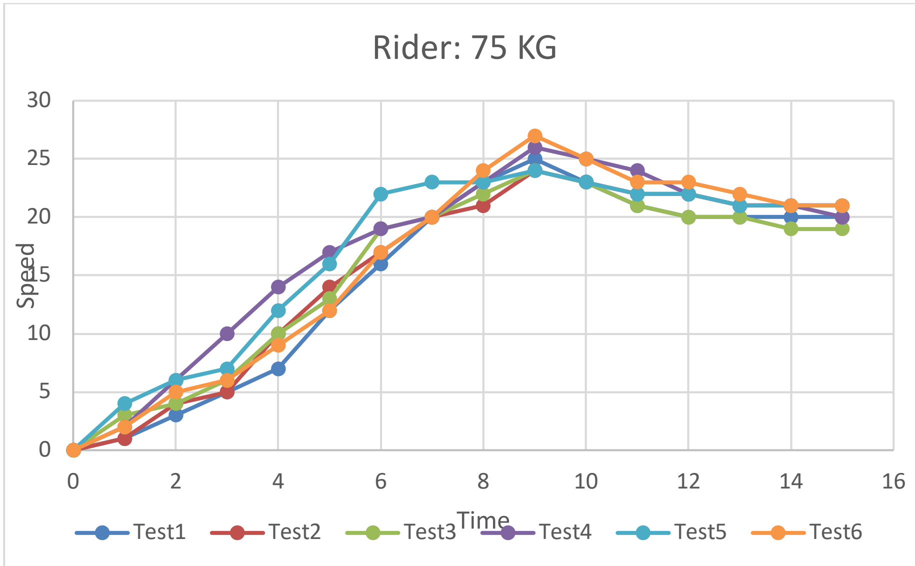

Figure 11: Time Response for Rider 3

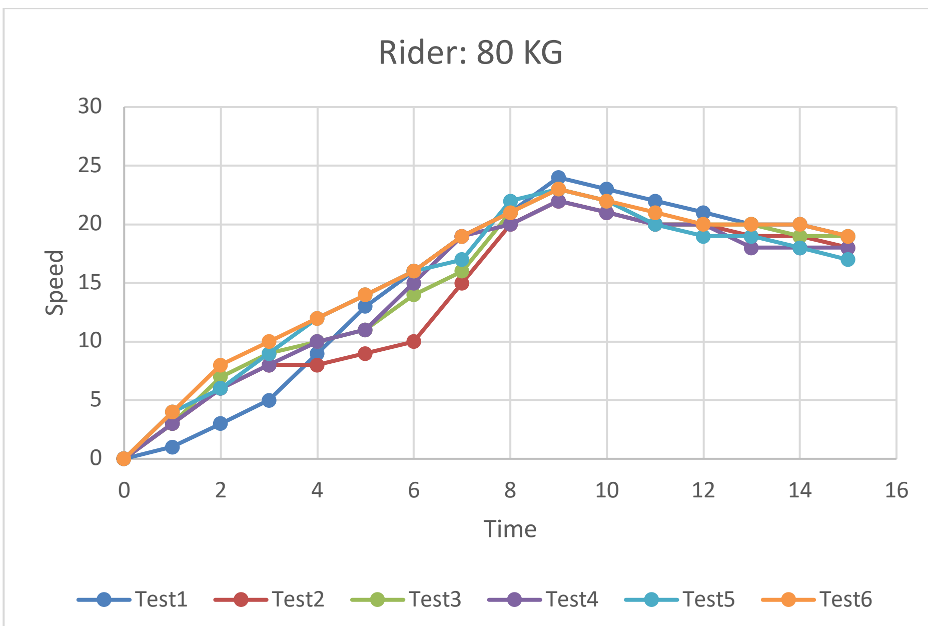

Figure 12: Time Response for Rider 4

4.3 Discussion

As shown in Figure 9 to 12, the BSAS is able to control the speed and reach the steady state finally. By using the data from Table 10 above, the average steady-state value and the settling time for each person can be calculated, as shown in table 11. From this table, it is clear that the steady-state value is around 3% and the settling time is less than 4 seconds, which satisfy the statement in non-functional specification. Also, from the first three riders of Table 11, it is clear that the settling time is increased when the weight of the rider increases, which is reasonable since the heavier the person is, the larger the acceleration is. However, the 80 KG rider has a lower settling time. This is because his speed at t=9s is not high and the required settling time is shorter.

Table 11: Steady State Value and Settling Time for Each Rider

Rider	Steady-state Value (km/h)	Settling time (s)
59 KG	20.33	2.83
70 KG	20.5	3.5
75 KG	20.67	3.83
80 KG	19.83	2.67

5. Discussion and Project Timeline

5.1 Evaluation of Final Design

Our design satisfies all of the essential functional and non-functional specifications in Table 1 and Table 2. With careful consideration and engineering analysis, the decisions are made for the component's selection. As a result, the functions of the components can be optimized and fully taken advantage of, and as a result, the components can perform their tasks as desired. All of the specifications are under consideration, and corresponding solutions are proposed to accommodate them. Moreover, the unpredictable difficulties in reality also have been taken into account such that the solutions can be adjusted and modified later.

5.2 Use of Advanced Knowledge

Knowledge of embedded system language from ECE 327 is used to write button functions and software algorithm for Arduino. For the sensor module, knowledge of digital and analog signal analysis from ECE 318 is used to obtain sensor signals. Feedback control theory from ECE 380 and nonlinear to linear system analysis from ECE 481 are used to design the control system of the bicycle speed. For power module, knowledge of power consumption theory from ECE 361 is eliminated to choose power supply type and power protection. In the end, to analyze the PCB circuit of Arduino, driver, and laser transmitter and receiver, circuit knowledge from ECE 432 is also required.

5.3 Creativity, Novelty, Elegance

The novelty of the project design is that the BSAS can detect the over-speed situation of the bicycle and automatically decreases the speed to the pre-set desired range, which provides a vital feature that can prevent potential accidents and injuries. It is creative and elegant since the feedback control system operates by the cooperation of these three main parts: speed sensor for speed measurement, controller for speed decrease and front panel for the user interface. The infrared ray transmitter and receiver are used for speed sensor and, PID (Proportional, Integral, Derivative) control loop is used for speed decrease to achieve accurate speed, acceleration, and distance. Moreover, LCD screen and buttons are used for the front panel such that the users can easily adjust the desired speed limit based on the rules of different countries, use the BSAS to slow down when the brake is broken and turn on and off the BSAS whenever they want.

5.4 Quality of Risk Assessment

The primary safety hazard relates to the speed data collection, decrease and testing. The test volunteer needs to perform over-speed riding to test the speed decrease function, and it is possible that the volunteer is physically injured during the testing process. This hazard was avoided by letting the tester wear protection suit during the over-speed test.

In addition, the BSAS is implemented on a bicycle for outdoor activities and potentially exposed to rain and snow, and the required components are energized by 7V to 12V DC voltage. There is such potential hazard that the tester gets an electric shock while testing. To avoid the hazard, tests were intentionally operated when the weather was good. Also, power insulation design was also included in the project to best ensure the tester's safety.

As a result, the risks were mitigated to the most extent such that nothing went wrong.

5.5 Student Workload

Table 12 summarizes the total number of hours for each student worked on this project so far, and Figure 13 and 14 include the detailed timeline for BSAS.

Table 12: Total Student Hours Worked

Student	Hours (498A / 498B / Total)
Ran Chen	73 / 70 / 143
Xi Fan	70 / 65 / 135
Tiancheng Gao	69 / 68 / 137
Zaoli Zhang	76 / 63 / 139

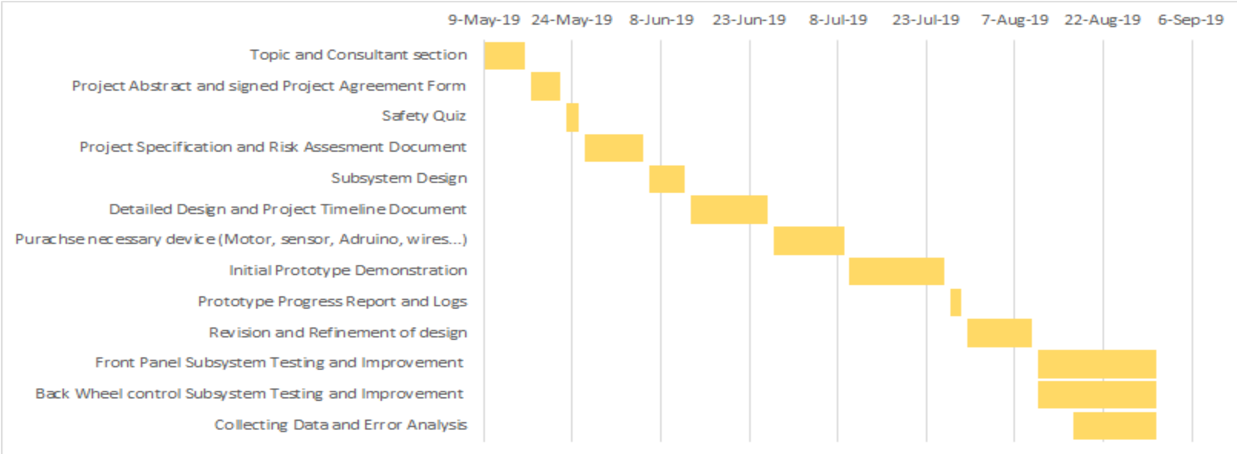

Figure 13: Project Timeline for 498A

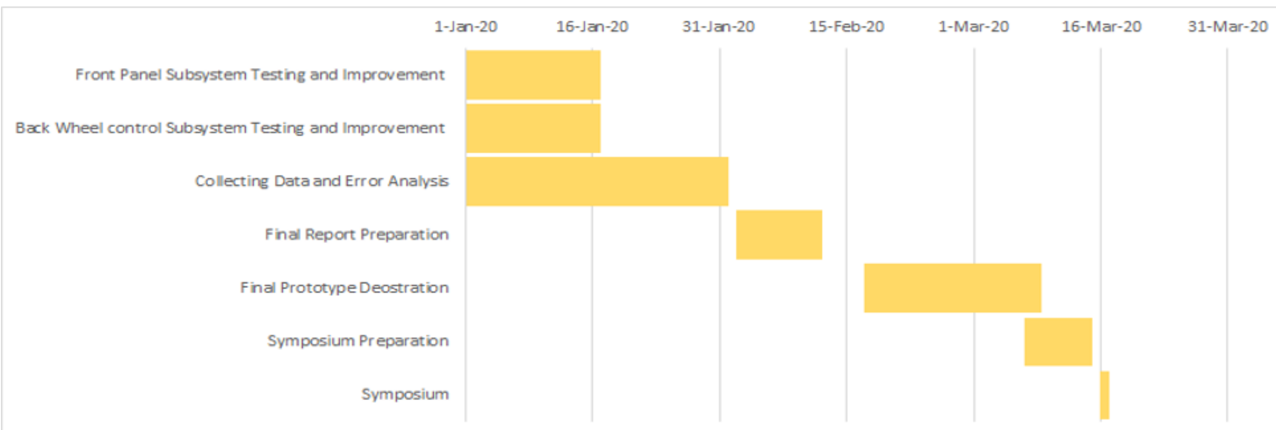

Figure 14: Project Timeline for 498B

During June 2019, the subsystem design and the detailed design and project timeline document are finalized as well as the implementation of the front panel and back wheel control system. The initial prototype demonstration and the prototype progress report will be finished during July 2019. The front panel subsystem and the back-wheel control subsystem testing and improvement will start simultaneously at the end of August 2019 and last for about four months until the middle of January 2020. The data collection and error analysis will start from the end of August 2019 and end at the start of February 2020. Finally, the final report preparation, the final prototype demonstration and the symposium will occur at the end of the 4B term during February and March 2020

References

[1] A. Ruggiero, "Bicycle Speed Limit: Marin County 'Common Sense' Crackdown," 6 April 2016. [Online]. Available: https://gearjunkie.com/15-mph-bike-speed-limit.

[2] K. Smith, "Calgary bike shop questions 20 km/h speed limit as cyclists warned to slow down," 15 June 2016. [Online]. Available: https://globalnews.ca/news/2765207/calgary-bike-shop-questions-20-kmh-speed-limit-as-cyclists-warned-to-slow-down/.

[3] [Online]. Available: https://www.canadarobotix.com/products/2316?variant=14423846879281. [Accessed 23 June 2019].

[4] "Dot Matrix Liquid Crystal Display Controller/Driver," [Online]. Available: https://www.elecrow.com/download/HD44780.pdf. [Accessed 23 June 2019].

[5] "ServoCity," [Online]. Available: https://www.servocity.com/hitec-pulley#.VDP3lucUbRc. [Accessed June 2019].

[6] G. Olsson, "Brake Performance and Stability for Bicycles," LenaNOVA, 15 May 2013. [Online]. Available: https://www.sveafordon.com/media/36782/Olsson_Brake-PerfStab-for-Bicycles_130515_public.pdf. [Accessed June 2019].

[7] "ROBOT US," [Online]. Available: https://www.robot-r-us.com/vmchk/motor-robot-servo/digital-metal-gear-ultra-high-torque-robot-servo-53-kg.cm-/-280.html.