Alpha Robo-Shifter

MAE 172: Electronic Bicycle Transmission

Advisor: Professor Bobrow
Team: Andre Aranez - Controls
       Edmond Kwok - Structure
       Kushal Shah - Electronics
       Ryan Wong - Controls/Electronics
March 21 2013
Executive Summary

The human body is most efficient at around 80 rpm. The purpose of this project is to create an automatic shifting system for the rear derailleur of the bicycle based on measuring the cadence of the cyclist. An automatic gear shift will target the set optimal cadence, and will shift gears up or down if cadence is too high or too low respectively. A potentiometer is added to adjust the target cadence for training purposes, high cadence training 110 rpm or higher or 70 rpm for beginner cyclists/ commuters. This project opens a new world for bicyclist and lets the rider enjoy true fun ride.

Table of Contents

Executive Summary ........................................................................................................................ 1

1. Motivation ................................................................................................................................. 2

2. Competition in the Market........................................................................................................ 2

3. List of the Parts for the Project ................................................................................................. 4

Hall Effect Sensor ....................................................................................................................... 4

4. Description of the Design ......................................................................................................... 6

4.1 SolidWorks (CAD) Model of Alpha Robo-Shifter ................................................................. 6

4.2 SolidWorks Drawing of Alpha Robo-Shifter ......................................................................... 7

4.3 Assembly of the Device: ......................................................................................................... 9

4.4 Electronics of Alpha EBT- Building Circuits ...................................................................... 14

4.5 Control of Alpha EBT: ........................................................................................................... 16

4.6 Mathematical Model of Alpha EBT ...................................................................................... 16

a. Control Law: .......................................................................................................................... 16

b. Code: (In the Appendix) ......................................................................................................... 17

c. Power Calculations ................................................................................................................ 18

5. Future Plans ............................................................................................................................. 20

6. Team Contribution .................................................................................................................... 20

7. Appendix: ................................................................................................................................. 21
1. Motivation

The human body is most efficient at around 80 rpm. However, sometimes cyclists do not realize what the optimal cadence speed is for their body and continue wasting energy when the gear could have been shifted up or down to save the energy put into the paddling. For most bicycles, the rider must shift the gears manually based on the cadence. If the cadence is too high or low, most notably when going down or up a hill, the rider must manually shift the gears to be able to pedal more easily. However, it is possible for this system to be automatized which can save the trouble and energy while achieving desired or optimal cadence for the rider at all time. The motivation of this project is the desire to ride the bicycle at optimal/desired cadence speed at all times. This project is the answer to the question why ride like the rest of humanity when you can turn your bike into a Robo-shifting super machine?

The main objective of this project is to have an automatic gear shift to relieve the rider of this responsibility, giving him/her more freedom while riding. This will give the bicycle world a whole new look at the bicycles. This is taught to make bicycles faster, smarter and easier to use than a regular shifter, since the electronic version requires no inputs and is optimal for the bicyclists. Even an amateur bike rider will be able to feel the difference between electronic gear transmission and those on a standard road bike. There's no sticking or bumpy shifting, making the action smooth and effortless -- getting the shift right every time.

2. Competition in the Market

i. Browning Automatic Bicycle Transmission

The earliest automatic bicycle transmission is the Browning automatic bicycle transmission. This design consists of 5 major components: the onboard computer, the chain length compensator, the front and rear sprocket clusters, the gear selectors, and two communications
buttons on the handlebar. It is powered by a 9 volt battery. Each component provides a major advantage to this design. The computer shifts gears to maintain the preferred cadence. The chain length compensator requires no force or power. The sprocket clusters allow the shift to be smooth. The gear selector allows the rider to provide all of the necessary force to complete the shift, so that the gear selector requires minimal power. The communication buttons allow the rider to turn off automatic transmission, in case the rider would like manual transmission for a while. A disadvantage of this system is that it requires battery power. However, the rider should be able to travel long distances without needing to change the battery.

ii. Landrider

For the Landrider, The derailleur changes gears automatically. As the wheel turns and begins to speed up, the two weights on the derailleur spin faster moving the chain to a higher gear. As you slow down or brake, the wheel slows down and the derailleur weights slow down moving the chain to a lower gear. This feature tells the bike when to shift gears so riders don’t have to. This mechanism moves the transmission into the optimal position to maintain rider choice.
## 3. List of the Parts for the Project

<table>
<thead>
<tr>
<th>Part</th>
<th>Picture</th>
<th>Cost</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Uno</td>
<td><img src="image" alt="Arduino Uno" /></td>
<td>$25</td>
<td>SparksFun</td>
</tr>
<tr>
<td>20x4 Character LCD</td>
<td><img src="image" alt="20x4 Character LCD" /></td>
<td>$20</td>
<td>SparksFun</td>
</tr>
<tr>
<td>Slider Pot</td>
<td><img src="image" alt="Slider Pot" /></td>
<td>$2.50</td>
<td>SparksFun</td>
</tr>
<tr>
<td>Rotary Pot</td>
<td><img src="image" alt="Rotary Pot" /></td>
<td>$1</td>
<td>SparksFun</td>
</tr>
<tr>
<td>Hall Effect Sensor</td>
<td><img src="image" alt="Hall Effect Sensor" /></td>
<td>$1</td>
<td>SparksFun</td>
</tr>
<tr>
<td>Motor</td>
<td><img src="image" alt="Motor" /></td>
<td>N/A</td>
<td>Previous Year’s Class MAE 106</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
<td>Cost</td>
<td>Source</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------</td>
<td>-------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Amplifier</td>
<td>N/A</td>
<td>Developed</td>
<td></td>
</tr>
<tr>
<td>11.1 V &amp; 2200 mah</td>
<td>Battery</td>
<td>$12</td>
<td>Hobby People</td>
</tr>
<tr>
<td>Steel Plates</td>
<td></td>
<td>$15</td>
<td>Irvine</td>
</tr>
<tr>
<td>Mounting Screws</td>
<td>As Desired</td>
<td>$5-$10</td>
<td>Ace Hardware Store</td>
</tr>
<tr>
<td>Weld Nut</td>
<td>$</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>$100-$110</td>
<td></td>
</tr>
</tbody>
</table>
4. Description of the Design

We created a module that adapted to most bicycles using traditional cable tension, friction style shifting. Our design consists of a servo controlled by an Arduino that connects to the traditional shift cable to operate the rear derailleur. The Hall Effect sensor is used to measure the cadence, depending on the desired/optimal cadence speed, Arduino will send signal to the amp for shifting gear up or down. This system eliminates manual gear shifting, which can make the ride more fun.

The design begun with the SolidWorks model, then the parts were ordered and the manufacturing of the design was done. Then, the soldering and connection of wires were developed. Finally, the code was implanted and the Robo-Shifter was tested. During testing, major failure was the meeting the power requirements. However, the issue was correcting by using alternative power supply and rewiring of some connection. Finally, the Robo-shifter was born and the ride of lifetime begun.

4.1 SolidWorks (CAD) Model of Alpha Robo-Shifter
4.2 SolidWorks Drawing of Alpha Robo-Shifter
## 4.3 Assembly of the Device:

### Shifter Assembly

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 1 | Obtain a 3” x 6” piece of aluminum of ¼” thickness. | 1. Obtain 3” x 6” piece of aluminum of ¼” thickness.  
2. Drill holes indicated in drawing.  
Note: Picture on the left shows more holes than necessary. Please ignore those. |
| 2 | Obtain a 2x L bracket measuring 1” horizontal and 1.5” vertical. | 1. Obtain 2x L bracket measuring 1” horizontal and 1.5” vertical.  
2. Drill holes indicated in drawing. |
| 3 | Obtain a L bracket measuring 1” horizontal, 1.6” vertical, and 3” wide. | 1. Obtain L bracket measuring 1” horizontal, 1.6” vertical, and 3” wide.  
2. Drill holes indicated in drawing. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 4    | 1. Obtain Delrin piece measuring 1” x 0.5” x 0.25”.  
2. Drill holes indicated in drawing.  
3. Obtain M2 bolt, washer, and nut. |
| 5    | 1. Align holes in Delrin and bracket from step 3.  
2. Bolt through the bottom hole, as the top will be left for the shifter cable. |
| 6    | 1. Obtain two strips of 0.35” wide 1/8” thick aluminum.  
2. Drill bottom hole (1/4”) to fit on the plate.  
3. Bending pieces making sure bottom is 1” and entire piece is 1.5” in height. The vertical piece should be perpendicular to the bottom piece.  
4. Drill hole (1/4”) on vertical face to fit on L bracket. This piece serves as a gusset. |
<table>
<thead>
<tr>
<th>Page</th>
<th>Image</th>
<th>Instructions</th>
</tr>
</thead>
</table>
| 7    | ![Image](image1.png) | 1. Obtain 1 ¼” steel shaft collar and 3” x 0.5” steel strip.  
2. ¼” holes are drilled on the ends of the strip 0.5” away from the edge.  
3. Use a mill to cut the bottom of the shaft collar to create flat service.  
4. Making sure it is centered, weld bottom piece of shaft collar onto the steel strip. |
| 8    | ![Image](image2.png) | 1. Obtain flexible spider shaft coupling hub, ¼” and 3/16” bore, 5/8” OD |
| 9    | ![Image](image3.png) | 1. Assemble weld nut and bolt together.  
2. A small piece of aluminum is added on the bottom to prevent the weld nut to rotate. M3 bolt and nuts are used to secure it onto the weld nut.  
3. Attach springs to the side with the head of the bolt. One on the bolt and one through the cable.  
   Note: Springs are not shown in the pictures. |
1. Assemble everything together according to the picture.
2. Despite the coupling’s ability to deal with misalignments, it is still important to align everything to prevent binding and excess friction.

3. Brackets and mounts were made for Arduino and amplifier. Brackets attach on the bottom through the same four bolts used for the motor mount and brackets.

1. Mounts are bent according to the bike frame to secure the shifter assembly. Rubber was superglued onto the mount to prevent slipping. This is the last piece that is attached with the four bolts.

2. Tighten four nuts in alternating pattern. Make sure to not over tighten to where plate will be bent.
### 13

1. Thread shifter cable through starting from the weld nut.
2. To index correctly, start with the shifter at the lowest position.
3. Making sure there is not too much slack, tighten bolt on the derailleur to secure shifter cable.
4. Use barrel adjuster to optimize shifting.

### 14

1. Obtain Delrin and create slit for Hall-effect sensor.
2. Attach sensor assembly to chain-stay of bike using zip ties.

### 15

1. Attach magnet to the crank using zip ties.
2. Make sure sensor and magnet is close to each other.
4.4 Electronics of Alpha EBT- Building Circuits

This section shows how to wire up the circuit used in the Alpha Robo-Shifter. This part of the robot shows the schematic where it arranges all the electric supplies of the robot such as input and out voltages for the system to perform. In order for securing the strength of wires, the soldering was performed. By using the soldering method, all the loosen wires were secured. This was an advantage because it held wires together with a strong tie during the high vibrational motion caused during the hopping motion across the table. This part contains the electric component diagram and the connection table.
<table>
<thead>
<tr>
<th>Connection A</th>
<th>Connection B</th>
<th>Connection A</th>
<th>Connection B</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCD Pin Vss</td>
<td>Common GND</td>
<td>Linear Pot Wiper</td>
<td>Arduino Pin- A0</td>
</tr>
<tr>
<td>LCD Pin Vcc</td>
<td>+5V Source from Arduino</td>
<td>Linear Pot Power</td>
<td>+5V from Arduino and Common GND</td>
</tr>
<tr>
<td>LCD Pin R/S</td>
<td>Arduino Pin- D9</td>
<td>Rotatory Pot Wiper (Target Cadence Pot)</td>
<td>Arduino Pin- A2</td>
</tr>
<tr>
<td>LCD Pin R/W</td>
<td>Common GND</td>
<td>Rotatory Pot Power</td>
<td>+5V from Arduino and Common GND</td>
</tr>
<tr>
<td>LCD Pin E</td>
<td>Arduino Pin- D8</td>
<td>Amp Vcc</td>
<td>11.1 V Power Supply (+)</td>
</tr>
<tr>
<td>LCD Pin DB4</td>
<td>Arduino Pin- D6</td>
<td>Amp Vss</td>
<td>11.1 V Power Supply (-) (Common Ground)</td>
</tr>
<tr>
<td>LCD Pin DB5</td>
<td>Arduino Pin- D5</td>
<td>Amp GND</td>
<td>(Common Ground)</td>
</tr>
<tr>
<td>LCD Pin DB6</td>
<td>Arduino Pin- D4</td>
<td>Amp Analog In (AIN)</td>
<td>Arduino Pin- D10 – PWM Signal</td>
</tr>
<tr>
<td>LCD Pin DB7</td>
<td>Arduino Pin- D3</td>
<td>Amp Direction (Dir)</td>
<td>Arduino Pin- D12</td>
</tr>
<tr>
<td>Hall Effect Sensor Pin3</td>
<td>Arduino Pin- D2</td>
<td>Amp M1</td>
<td>Motor (+)</td>
</tr>
<tr>
<td>Hall Effect Sensor Pin2</td>
<td>Common GND</td>
<td>Amp M2</td>
<td>Motor (-)</td>
</tr>
<tr>
<td>Hall Effect Sensor Pin1</td>
<td>+5V from Arduino</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.5 Control of Alpha EBT:

4.6 Mathematical Model of Alpha EBT

a. Control Law:

In this control law Theta Desired is linearly related to the position of the weldnut attached to the ball screw.

1. Block Diagram:

(Figure 1.5 – The Control Law)

The information needed for mathematical computations:

Error Measured in the Diagram is  \[ e = \Theta - \Theta_d \]

The Motor Torque Equation is  \[ T = J\dot{\Theta} \]
The torque that applied was proportional to supplied voltage => \( V = (\text{Alpha})(\text{Torque}(T)) \)

Explanation of Block Diagram:

The diagram above shows the feedback controlled block diagram of the robot.

2. The Control Law :-

\[
V = -K_p(\Theta - \Theta_d)
\]

where, \( T = \text{Torque applied by motor} \)

\( \Theta = \text{Actual Angle} \)

\( \Theta_d = \text{Desired Angle} \)

\( K_p = \text{Proportional Gain} \)

3. Transfer Function Derivation:

\[
\frac{J\ddot{\Theta}}{\alpha} = -K_p(\Theta - \Theta_d)
\]

\[
\frac{\ddot{\Theta}J}{\alpha} + K_p\Theta = K_p\Theta_d
\]

\[
\Theta(s)\left(\frac{J}{\alpha s^2} + K_p\right) = \Theta_d(s)K_p
\]

4. Transfer Function = \( H(s) \) :-

\[
H(S) = \frac{\Theta(s)}{\Theta_d(s)} = \frac{K_p}{\frac{J}{\alpha s^2} + K_p}
\]

b. Code: (In the Appendix)
c. Power Calculations

Motor Specs:

<table>
<thead>
<tr>
<th>Speed</th>
<th>Current</th>
<th>Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>163</td>
<td>0</td>
</tr>
<tr>
<td>52</td>
<td>1000</td>
<td>260</td>
</tr>
<tr>
<td>0</td>
<td>2000</td>
<td>600</td>
</tr>
</tbody>
</table>

From the Data above, we created the following graph for Load vs Current at 24VDC.

![Load vs Current at 24VDC Graph](image)
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Equations</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force needed</td>
<td>-</td>
<td>15 lb</td>
</tr>
<tr>
<td>Travel Distance</td>
<td>-</td>
<td>0.75 in</td>
</tr>
<tr>
<td>Linear Work</td>
<td>Force * Distance</td>
<td>11.2</td>
</tr>
<tr>
<td>TPI</td>
<td>Rotaion Per Inch</td>
<td>18</td>
</tr>
<tr>
<td>Rotational Work</td>
<td>Torque * Θ</td>
<td>-</td>
</tr>
<tr>
<td>Equating Equations</td>
<td>Linear Work= Rotational Work</td>
<td>-</td>
</tr>
<tr>
<td>Solving for Torque needed</td>
<td>T = F / (2π*TPI)</td>
<td>2.12 (oz-in)</td>
</tr>
<tr>
<td>Stall Torque</td>
<td>2 * Torque Needed</td>
<td>4.24(oz – in)</td>
</tr>
<tr>
<td>Solving for Current from the Plot of Torque vs. current</td>
<td>Current = 3.0556(T_{need}) + 178.39</td>
<td>184.87 mA</td>
</tr>
<tr>
<td>Power at 24 V</td>
<td>I * V</td>
<td>4.43 W</td>
</tr>
<tr>
<td>Amp Draw at 12.6 V</td>
<td>P / (V=12.6V)</td>
<td>352 mA</td>
</tr>
</tbody>
</table>
5. Future Plans

After redesign for manufacture the power requirements will be lower and the battery life will be extended to ride with Robo-Shifter for Longer time. Furthermore, the alert signal or flashing light will be implanted in the display to alert the rider before the gear change is performed by the Arduino controls. Furthermore, the manual override of the gear shift will also be implanted in the system. Finally, the system can be minimized to be compact system.

6. Team Contribution

Andre Aranez
• Lead of the Controls for the Robot
  o Developed the controls for the robot using Arduino
  o Done the Calibration of the potentiometer voltage values during the test and competition by Arduino

Edmond Kwok
• Lead for the Mechanical Design and Mechanical Fabrication
  o Worked on the design from conceptual to its final phase.
  o Assisted with fabrication and conducted the major fabrication.
  o Mounted and Fabricated the Assembly.

Kushal Shah
• Lead for the Electronics
  o Developed the Electrical Components including the current amplifier
  o Developed the solid connection between the different electrical components of the hopper using the soldering process.
  o Assisted with testing the hopper and repaired/adjusted any physical components as required.

Ryan Wong
• Lead for Controls and Electronics
  o Researched linear servos
  o Researched mounting automatic transmission on bike
  o contributed to motor control code
  o wrote rough draft of report
7. Appendix:

```c
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(9, 8, 6, 5, 4, 3);

#define numberGears 7
// pins
int potOut = A0;//motor Pot
int potIn = A2;//In Pot
int dirPin = 12;//direction pin
int motorPin = 10;

//variables
int perror = 0;
int kp = 50;

int potMin = 30;
int potMax = 250;
//int cassette[numberGears];//Pot positions of each cog
int cassette[] = {
    250, 190, 165, 120, 95, 65, 30
};

//Cadence
int cadenceSpeed = 0;
int previousCadence = 0;
volatile long previousTime = 0;
volatile long currentTime = 0;
long lastShiftTime = 0;
long currentShiftTime = 0;
long shiftInterval = 2000;
int threshold = 10;
int targetCadence = 40;
int cog = 4;
```
void setup() {
    attachInterrupt(0, cadence, RISING); // pin 2, 1 on pin 3
    Serial.begin(9600);
    pinMode(dirPin, OUTPUT);
    pinMode(motorPin, OUTPUT);
    
    for (int count = 0; count < numberGears; count++) {
        cassette[count] = increment * (numberGears - count) + potMin;
        Serial.println(cassette[count]);
    }
}

TCCR1B = 0x01; // Timer 1: PWM pins 9 & 10 @ 32 kHz

void loop() {

    //LCD
    lcd.setCursor(0, 1);
    // print the number of seconds since reset:
    lcd.print("Cadence");
    lcd.setCursor(10, 1);
    // print the number of seconds since reset:
    lcd.print(cadenceSpeed);
    lcd.setCursor(0, 2);
    // print the number of seconds since reset:
    lcd.print("Target Cadence");
    lcd.setCursor(16, 2);
    // print the number of seconds since reset:
    lcd.print(targetCadence);
    lcd.setCursor(0, 3);
// print the number of seconds since reset:
lcd.print("Cog");

cursor(5, 3);
// print the number of seconds since reset:
lcd.print(cog + 1);

//set targetCadence
targetCadence = map(analogRead(A2), 0, 1023, 40, 140);

currentShiftTime = millis();
long timeDifference = currentShiftTime - lastShiftTime;

//Switch Gears
if (abs((float)cadenceSpeed - (float)targetCadence) >= threshold && previousCadence != cadenceSpeed && timeDifference > shiftInterval) {
lastShiftTime = millis();
previousCadence = cadenceSpeed;
//Shift Up
if (cadenceSpeed - targetCadence > 0 ) {
cog += 1;
if (cog == numberGears) {
cog = numberGears - 1 ;
}
}
//Shift Down
else {
cog -= 1;
if (cog <= 0) {
cog = 0;
}
}
}

//Motor Control
static int dir = 1;
int out = analogRead(potOut);
int in = analogRead(potIn);
int inVal = cassette[cog];
int outVal = map(out, 0, 1023, 0, 255);
if (inVal > 250) {
    inVal = 250;
}
else if (inVal < 30 ) {
    inVal = 30;
}

int error = inVal - outVal;

if (error < 0 ) {
    dir = HIGH;
}
else {
    dir = LOW;
}
int U = kp * abs(error);
if (U > 254) {
    U=254;
}
analogWrite(motorPin, U);
digitalWrite(dirPin, dir);

void cadence() {
    currentTime = millis();
    cadenceSpeed = 60000 / (currentTime - previousTime); //rev per min
    previousTime = currentTime;
}