Modeling Abrasive Wear of a 3D Printer Extruder Drive Mechanism

by

Joseph Sandoval

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Bachelor of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author .... Signature redacted
Department of Mechanical Engineering
May 17, 2016

Certified by..... Signature redacted ....
David Hardt
Professor of Mechanical Engineering: Ralph E. and Eloise F. Cross
Professor in Manufacturing
Thesis Supervisor

Accepted by ...... Signature redacted ......
Anette Hosoi
Associate Professor of Mechanical Engineering, Undergraduate Officer

Modeling Abrasive Wear of a 3D Printer Extruder Drive Mechanism

by

Joseph Sandoval

Submitted to the Department of Mechanical Engineering
on May 17, 2016, in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Mechanical Engineering

Abstract

Additive manufacturing, more specifically 3D printing using fused filament fabrication (FFF), is a valuable technique; however, little has been done in developing a new mechanism for driving filament through the hot end of the 3D printer. This investigation focuses on a prototype extruder mechanism that utilizes two counter rotating motors to force filament through the nozzle. The plastic passes through the armateurs of the two motors and the oppositely-threaded shafts drive the filament while balancing each others torque. This design provides better protection against jamming of the nozzle. A bench top prototype was able to extrude filament at a rate much greater than traditional mechanisms allow, but the threads on the aluminum shafts wore down with very little use.

An abrasive wear model of the extruder shafts was developed in order to determine the theoretical lifetime of the shafts given a certain material hardness. The wear rate of the shafts is proportional to the hardness of the material and the square of the velocity of the extruded filament. Based on this model as well as experiments done with various materials, it is estimated that steel shafts will have a lifetime that is 3 to 10 times longer than aluminum shafts. Still, this lifetime is still far too short for a part meant to be used 24/7. The wear on the threads of the shafts is so severe that no feasible material could survive an adequate lifetime before failure.

Thesis Supervisor: David Hardt
Title: Professor of Mechanical Engineering: Ralph E. and Eloise F. Cross Professor in Manufacturing

Acknowledgments

I would like to thank Professor Hardt for being my advisor and the guidance I received throughout the process as well as the entire NVBots team for the help, advice, and opportunity to work with them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The focus of this thesis is a novel 3D printer extruder drive mechanism. The mechanism features two counter-rotating, oppositely-threaded shafts that drive the filament through the hot end of the extruder while balancing each others torque. A prototype of the mechanism was built and was able to successfully extrude filament; however, the threads on the shafts wore extremely quickly. The goal of this investigation is to develop a model for the wear of the threads and to determine if this design is viable for incorporation into a consumer product. Chapter 1 focuses on the design of the fused filament fabrication printer and the traditional extruder mechanism. This chapter also contains a model of the flow through the nozzle of the extruder. Chapter 2 contains a more thorough explanation of the design of the counter rotating extruder as well as an explanation of its advantages over other mechanisms. Chapter 3 explains the mechanical design and control system of the benchtop prototype that was developed. A model of the abrasive wear of the extruder drive shafts that was developed after testing is presented in Chapter 4. Conclusions and comments on the feasability of the design are presented in the final chapter.

1.1 Fused Filament Fabrication

Fused filament fabrication, or FFF, is the most commonly used and cost effective type of additive manufacturing and 3D printing for use in rapid prototyping. In this process, a thin piece of filament is fed into a hot nozzle via an extruder. The most common extruder mechanism consists of a stepper motor that spins a gear against the filament, driving it downward. This hot nozzle then melts the filament, and deposits it on top of or alongside more of the same material. By repeating this process, the 3D printer is capable of building an object layer by layer. The extruder and the plate the filament is being deposited on are attached to high precision stepper motors. This allows for high precision depositing of material, and the creation of high quality parts. Figure 1-1 shows a simple diagram of the various parts of the typical FFF 3D printer.

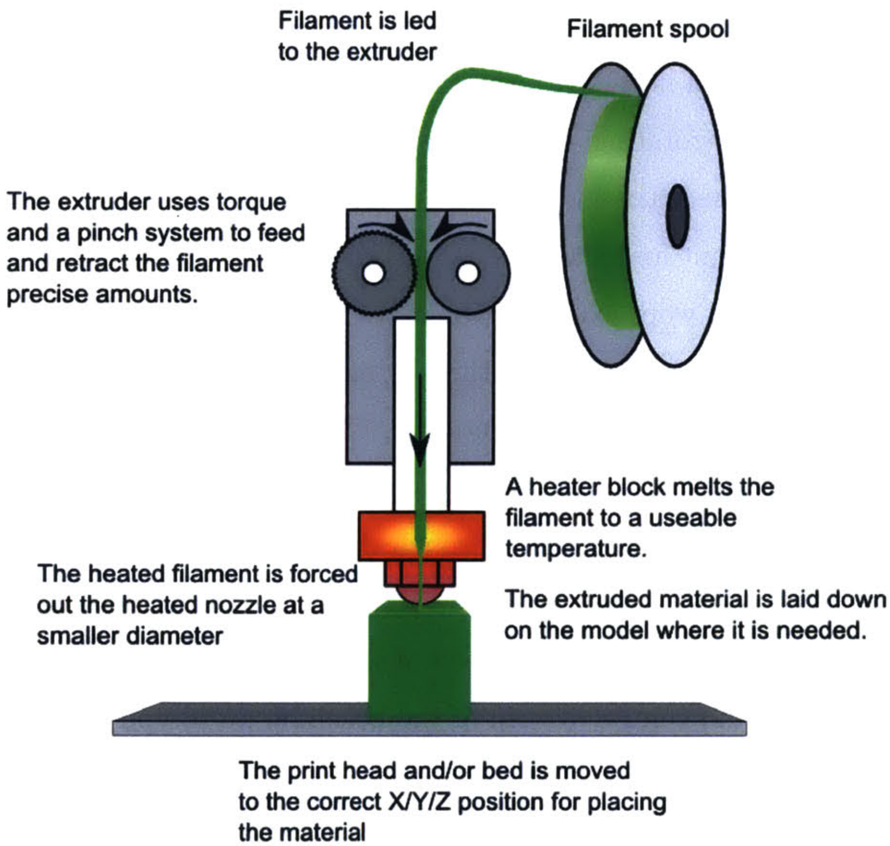

Figure 1-1: Diagram of a standard FFM setup. Reprinted from RepRap.org.[6]

In order to control the position of the extruder, computer software is necessary. The specific type of software is known as a slicer. The slicer software scans the computer model to be printed; it then produces the specific motor movements and speeds that are necessary to print the model. There are several parameters that may be changed to produce varying levels of printer quality and speed. The parameter this paper focuses on is the feed rate or the velocity of the filament as it moves through the extruder. The feed rate determines not only the potential speed of the 3D print, but also affects the force required from the extruder in order to move the filament through the nozzle due to an increase in back pressure as velocity increases.

1.2 Extrusion and Back Pressure

In order to model the wear of the extruder, the force exerted by the extruder shafts must first be determined. In order to extrude at a constant velocity, the net force on the filament must be zero. In order for this to occur, the extruder must exert a force equivalent to that generated by the back pressure of the filament as it moves through the nozzle. In order to determine the pressure drop across the filament, the velocity profile of the liquid filament must be determined. The following derivation of the velocity profile and subsequently the extrusion force required is adapted from An Integrated Approach to Thermodynamics. [1] In order to model the flow, the following assumptions were made:

1. The flow is steady and uniform
2. Gravitational effects can be ignored
3. The flow is axisymetric and fully developed

Not all of these assumptions hold true for the typical FFM nozzle; however, the model should still provide a good order of magnitude estimate for the force required to extrude. With these assumptions in mind, we are able to use the Navier-Stokes equation to model the flow of an incompressible fluid through a long, narrow, pipe. After the previous assumptions are taken into account, the Navier-Stokes momentum equation for the axial direction reduces to only the pressure term and a single viscous term. If the distance from the center of the pipe is r and the viscosity of the fluid is u, then the momentum equation reduces to:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) = \frac{1}{u}\frac{\partial P}{\partial z} \tag{1.1}$$

This is a linear differencial equation and can be solved quite easily. This yields the solution:

$$v = \frac{1}{4u}\frac{\partial P}{\partial z}r^2 + c_1\, ln\, r + c_2 \tag{1.2}$$

In order to solve for the constants in equation 1.2 we must define certain boundary conditions of the flow. At r=0 the velocity cannot be infinity so the first constant must be equal to 0. Additionally, at r=R, or at the wall of the nozzle, the velocity, v=0. This is known as the no slip condition. By plugging in v=0 and r=R we obtain the following for the second constant:

$$-\frac{1}{4u}\frac{\partial P}{\partial z}R^2 \tag{1.3}$$

Combining equations 1.2 and 1.3 yield an equation relating the velocity of the fluid to the pressure gradient.

$$v = -\frac{1}{4u}\frac{\partial P}{\partial z}(R^2 - r^2) \tag{1.4}$$

If we assume the change in pressure is linear across the length, x, of the nozzle we can remove the differential from equation 1.4.

$$v = -\frac{1}{4u}\frac{\Delta P}{x}(R^2 - r^2) \tag{1.5}$$

We are attempting to find the maximum force the extruder must exert on the filament to overcome the back pressure created in the nozzle. The maximum velocity, and therefore maximum pressure drop occurs at the center of the filament, r=0. After substituting and rearranging we arrive at an equation relating the velocity of the filament to the pressure difference in the nozzle.

$$\Delta P = \frac{4uxv}{R^2} \tag{1.6}$$

In order to find the force on the filament we simply need to multiply the pressure by the area of the nozzle. This provides an equation that relates extrusion force, F, required to filament velocity.

$$F = 4\pi uxv \tag{1.7}$$

Equation 1.7 shows that the extrusion force necessary is proportional to the viscosity of the melted plastic, the length of the nozzle, and most importantly the velocity or feed rate of the filament. The required force was calculated at various velocities using a value of 1500 Pa s for the viscosity of polylactic acid, 1.75 mm for the diameter of the filament, 0.4 mm for the diameter of the nozzle, and a nozzle length of 2 cm. Figure 1-2 shows the extrusion force required for commonly used filament feed rates. At 1000 mm/min approximately 6 N or force is required to extrude filament and approximately 1 N of force is required for each 160 mm/min increase in feed rate. It should be noted that the FFF nozzle does not exactly satisfy all of the previous assumptions; however, the model should still be adequate for an order of magnitude estimation and subsequent analysis.

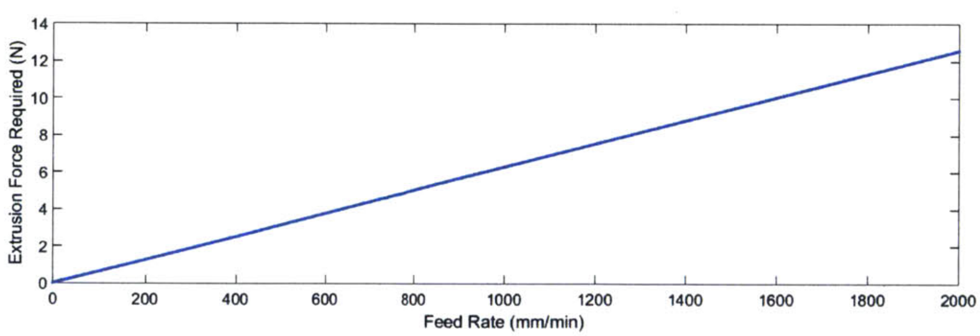

Figure 1-2: Linear graph of extrusion force vs. feed rate.

1.3 Typical Extruder Design and Limitations

The typical FFF extruder is devided into two parts, the hot end and the cold end. The hot end contains the nozzle and is responsible for melting the plastic, while the cold end contains the extruder drive mechanism. The standard FFF printer utilizes a pinch wheel drive mechanism. Figure 1-3 shows a more detailed sketch of a typical pinch wheel mechanism. In this case the stepper motor drives the filament directly, and the filament is pinched between the toothed wheel of the stepper, and a free spinning bearing.

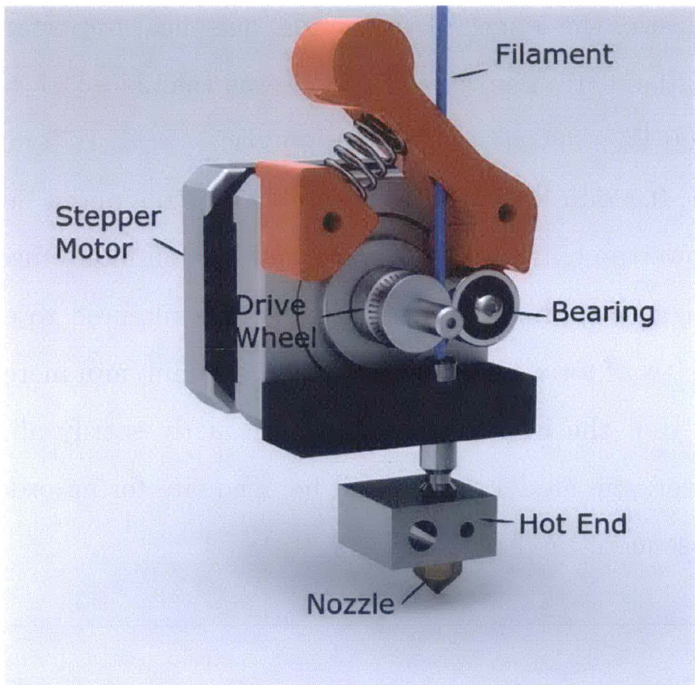

Figure 1-3: Sketch of typical FFF extruder showing pinch wheel mechanism. Reprinted from RepRap.org [5]

The typical mechanism features a stepper motor that is either geared, in order to create more driving torque, or, more commonly, directly drives a wheel that rotates against a bearing, with the filament pinched between. The bearing is spring loaded to ensure that the filament remains pinched between the two rotating surfaces. As the motor rotates, the two surfaces pinch the filament on either side and drive the filament downward. This design has several shortcomings. Firstly, the contact area between the driving wheel and the filament is very small, meaning the motor has to generate a high contact pressure in order to drive the filament. Secondly, if the nozzle jams, the motor continues to rotate against and wear down the filament. These leads to particles in the extruder and possibly more jams. Finally, this mechanism is not extremely precise, as the drive wheel may slip against the filament, furthermore, each rotation of the motor results in approximately 3 cm of filament extruded in the direct drive case.

Chapter 2

The Counter Rotating Extruder

2.1 Design

The counter rotating extruder (CRE) is a novel for an extruder drive mechanism that is designed to combat many of the problems with typical extruder desings that were explained earlier. The design is extremely simple and uses minimal components. The CRE was designed in order to maximize contact with the filament and therefore increase the amount of force that can be exerted on the filament. This will allow a higher feed rate and lead to less jams of the nozzle. The mechanism consists of two, oppositely spinning, oppositely threaded, through holes that are aligned. The filament passes axially through the shafts of the motors; the design is very robust. Two motors are required so that the torque on the filament is balanced and it does not spin as it moves through the extruder. Each through hole passes through the central axis of the armature of a motor, and the two motors spin in opposite directions.

This system is superior to the current standard. The CRE has a much larger area of contact and can exert more force on the filament without the risk of slipping. Additionally, the counter rotating extruder exerts force around the entire area of the filament rather than just a single side. This also allows for a greater distribution of the extrusion force which helps to prevent filament buckling. Furthermore, this mechanism is capable of much more precise filament deposition. The amount of filament desposited is directly related to the lead of the thread within the shafts. A single rotation of the motors results in only 0.4 mm of filament extruded. Each of these characteristics contribute to an extruder that should be able to feed filament faster and prevent jams better than the current designs in use. Figure 2-1 shows a cutaway model of the prototype of the counter rotating extruder. Table 2.1 shows a comparisson of the typical extruder and the counter rotating extruder. A plus symbol indicates that the particular extruder design variant has an advantage in that category. While the counter rotating extruder is larger and heavier than the typical extruder, it makes up for this in its precision, its simplicity of design and minimal components, and ability to exert a high extrusion force on the filament. Its large area of contact helps to prevent jams and is able to exert more force on the filament; however, this also creates problems due to a massive sliding distance that causes severe wear. By comparison, the traditional extruder design has a very small area of contact, and virtually no sliding distance, yielding virtually no wear on the wheel that drives the filament in the case of the typical FFF extruder.

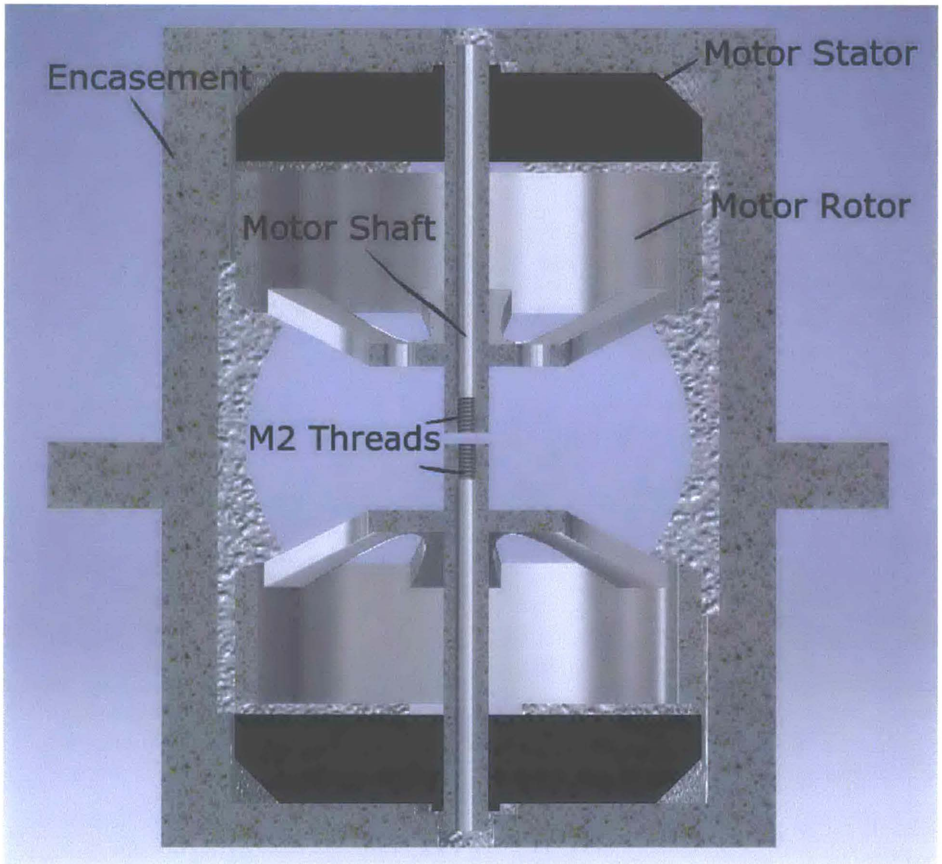

Figure 2-1: CAD model of the counter rotating extruder prototype

Attribute	Typical Extruder	Counter Rotating Extruder
Volume	+	-
Mass	+	-
Simplicity of Design	-	+
Number of Components	-	+
Ease of Manufacturing	+	-
Area of Contact	-	+
Sliding Distance	+	-
Possible Extrusion Force	-	+
Extrusion Precision	-	+

Table 2.1: Length of Filament and Feed Rate Data

Chapter 3

The Prototype

3.1 Mechanical Design

The mechanical components of the counter extruder prototype consisted of two brushless DC motors. The motors were designed for use on multi-rotor copters and were chosen for their ability to spin at very high speeds while still maintaining sufficient torque. The specifications for the particular motor used may be found in Appendix A.1. The motors shafts were replaced with custom shafts that were machined with through holes and tapped both right and left handed in order to drive the filament. The specifications for these shafts are located in Appendix A.2. The motors sit within a housing that was 3D printed. The housing ensures that the motors are secure as well as provides the correct spacing between the motors. It also contains mounting features and cable holes for sensors that are necessary for the operation of the motors. The housings specifications are located in Appendix A.3 and a photo is shown in figure 3-1. The extruder housing sits in a third 3D printed which acts as an adaptor for the hot end as well as a platform for the rest of the extruder. This part is shown in figure 3-2.

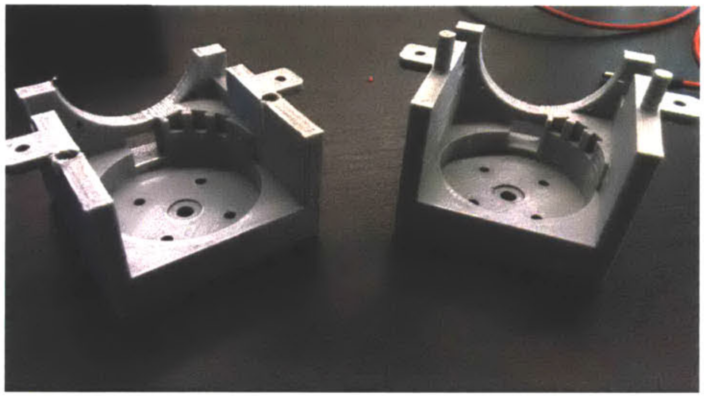

Figure 3-1: The two halves of the motor encasement.

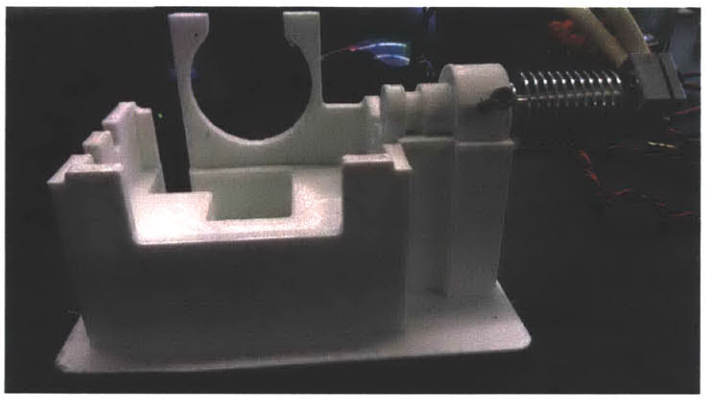

Figure 3-2: Encasement base and hot end mount.

3.2 Control System

In order for a brushless DC motor to properly rotate, it must receive information on where each of the magnets in the rotor are in order to electrify the right sections of the inner coils at the right time. One way to detect the location of the magnets is through the use of hall effect sensors, a digital sensor that switches on or off in the presence of a magnetic field. These hall effect sensors were already incorporated into the design of the prototype in order to accommodate for this. They were also used in order to sense the rotational velocity of the motor and implement a PID speed controller on the extruder. Simply controlling the duty cycle of the PWM sent to the motor controller was insufficient. Once filament once introduced, the motor would slow down from its target speed given the same duty cycle, therefore, a PID controller had to be implemented in order to ensure that not only was each motor rotating at the set speed, but that they were rotating at the same speeds. After a calibration process that included modeling in Matlab as well as using the actual motors, the following values were obtained for P, I, and D: 0, 0.05, and 0.001. Thus the primary control method for the motors was integral. The control of the actual motors was done using two Arduino Nanos and two Atmel A4915 Brushless DC Motor Controller Evaluation Boards. Due to the precise synchronization that is required in order to drive a brushless DC motor, special driver integrated circuits are necessary. The A4915 was chosen for two main reasons, firstly, it allows the use of hall effect sensors in controling the motor synchronization, which is necessary for control at low speeds. Additionally, the driver accepts a simple PWM input to control the speed of the motor, a signal that is easily produced by the Arduino. The motors are controlled in the following way: a target rotational velocity is set by entering it into the serial monitor of the computer connected to the master Arduino. This Arduino then sends this set velocity to the slave Arduino. Both Arduinos then generate a PWM signal and send it to the A4915 drivers. The drivers then utilize the signals from the hall effect sensors to drive the motors. One hall effect sensor on each motor is also used by the Arduino to sense the velocity of the motor so that the Arduino may run a PID controller and keep the velocity as close to the set point as possible. Appendix B.1 is a wiring diagram of the extruder prototype. Appendix B.2 and B.3 contain the master and slave Arduino code respectively. Figure 3-3 shows a picture of the motor driver board that was used. Figure 3-4 is a photograph of the entire prototype setup. This prototype was able to successfully drive filament at very high feed rates in excess of 2000mm/min; however, the aluminum shafts used to fabricate the custom motor shafts wore out extremely quickly.

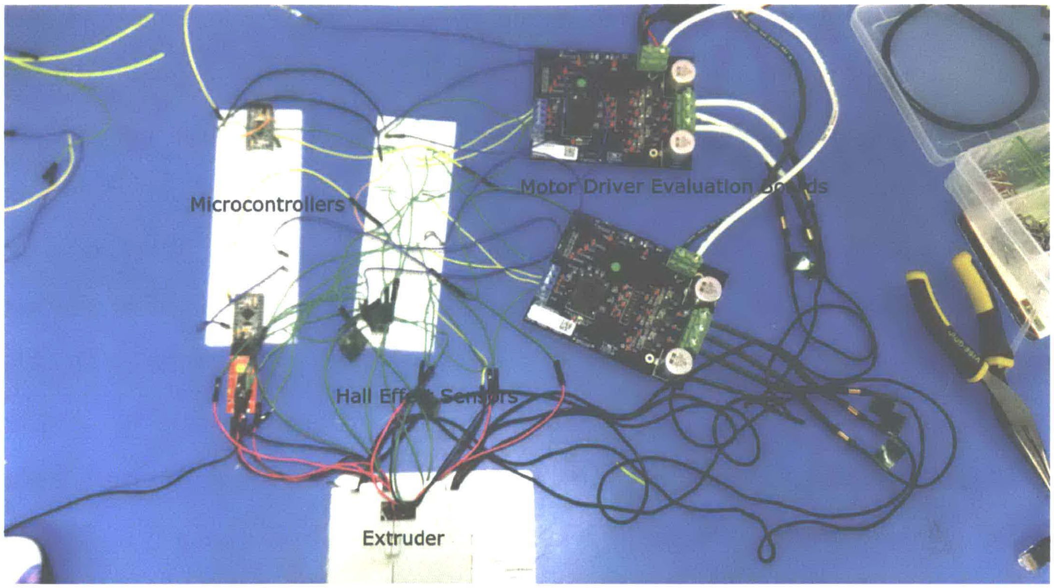

Figure 3-3: Extruder prototype with control system electronics.

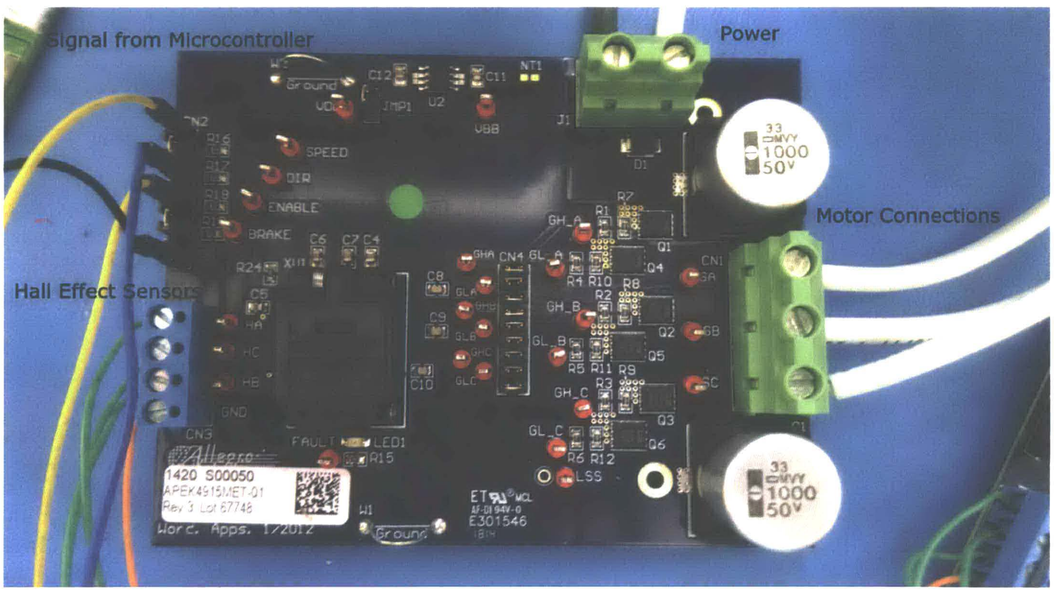

Figure 3-4: Motor Driver Board.

Chapter 4

Modeling the Wear

4.1 Determining the Wear Parameter

The wear on the shafts of the extruder wore down extremely quickly, to the point where they were unable to grip onto the filament and drive it downward. The primary method of wear in this instance is abrasive, which occurs when one material rubs against another, plastically deforming it. In this model, the volume of material worn at failure is most important. Failure occurs when enough of the threads have worn away so that the minor diameter, that is the narrowest diameter within the threaded shaft, becomes greater than the diameter of the filament, 1.75 mm. The threads are modeled as a long isosceles triangular prism with base b, height d, and length L. The worn volume of the threads is then equal to:

$$V = 0.5bdL \tag{4.1}$$

In the case of abrasive plastic wear, it can be assumed that the normal contact pressure, the load, F, divided by the area of contact, is equal to the hardness value, H, of the wearing material.[2] That is:

$$H = \frac{F}{contactarea} = \frac{F}{d \times contactlength} \tag{4.2}$$

By substituting equation 4.2 into equation 4.1 we can obtain an equation that relates the worn volume to the sliding distance, load, and hardness of the worn material. It should be noted that here the term "contact length" refers to the number that is the ratio between the contact area and the worn depth. The exact number does not need to be known because a constant modifier will need to be added to the final equation regardless.

$$V = \frac{0.5bFL}{H \times contactlength} \tag{4.3}$$

Equation 4.3 represents the worn volume in ideal plastic abrasive cutting; however, there are many other modes of abrasive wear. The relative hardness of the materials, the specific material properties they process, the specific geometry of the interaction, and the exact nature of the wear particle generated all affect the worn volume. In order to accommodate for this the abrasive wear parameter, K, is introduced.

$$V = \frac{KFL}{H} \tag{4.4}$$

This is the equation that will be used to solve for K for this particular system. During experimentation the threads wore out very quickly and the filament was difficult to extrude because of issues with the torque balance.

The method of testing was as follows. The hot end was heated to temperature and the motors set to a velocity, usually resulting in a feed rate of approximately 600 mm/min. The filament was then gripped with a pair of pliers and inserted into the rotating motor shafts. After a very short time the filament would either snap due to an in balance of torque in the motors, or more commonly, the threads of motor shafts would wear to failure and the shafts would simple rotate while the filament stayed in place.

Due to these shortcomings, extensive failure time vs velocity data does not exists. Instead, I looked at the total length of the filament extruded and divided that by the number of shafts used and the average feed rate used. The limited data is shown in table 4-1.

Total Length of Filament Extruded	10m
Average Feed Rate	800 mm/min
Number of Shafts	12

Table 4.1: Length of Filament and Feed Rate Data

From this, I determined an approximate average time for failure of t=60s for shafts manufactured from Aluminum 6061. We may also obtain an equation that relates the failure time to the velocity of the filament. First, we assume that the sliding distance, L, at failure is equal to the velocity of the plastic sliding across, U, times the failure time, t. We then rearrange the equation to solve for t.

$$t = \frac{VH}{KFU} \tag{4.5}$$

However, equation 1.7 relates the extrusion force to the velocity. We just need to convert the sliding velocity U to the filament feed rate v by using the pitch diameter, p, of the threads and the radius, r, of the filament. Substituting equation 1.7 into equation 4.5 and making the necessary conversions and simplifications yields the following.

$$t = \frac{VHp}{8\pi^2 Kuxrv^2} \tag{4.6}$$

Equation 4.7 was used to calculate an estimate of K. The relevant data used is shown in table 4.2

Average time to failure	60s
Average Feed Rate	800 mm/min
Nozzle Length	2 cm
PLA Viscosity	1500 Pas
Al Hardness[3]	270 MPa
Worn Volume	2.15 mm^3

Table 4.2: Data used for estimation of wear parameter

Using the above data as well as equation 4.6. Yields an estimation of K=10. This value of the wear parameter K is specific to this system and the geometry and material properties that this involves. This value of K is two orders of magnitude greater than typical cases of abrasive wear, characterizing this specific case as severe.

4.2 Modeling the Time to Failure

Equation 4.6 can be utilized to analyze the time to failure in relation to both velocity and hardness. It is evident that the time to failure will be proportional to the hardness of the material being worn away. This was evident from limited tests with shafts manufactured from A2 tool steel demonstrated this. They lasted slightly longer but only around 2-3 times as long, as predicted by their hardness value of 641 MPa [4] as well as the linear relationship between hardness and time to failure.

The methodology of obtaining data for the steel shafts was equivalent to that of the aluminum shafts. The filament was forced through the shafts until failure. In this case, only two shafts were worn until failure. The average feed rate was again 800mm/min. A total of approximately 4.5 m of filament was able to be extruded before failure of the A2 steel shafts. Again the total length of the filament was divided by the number of shafts as well as the feed rate. This yielded an approximate failure time of nearly 3 minutes.

The time to failure will be inversely proportional to the square of the velocity of the filament. Figure 4-1 shows the graph of failure time vs. velocity. It is clear that at high filament feed rates, the wear rate becomes astronomical, even for steel. Figure 4-2 shows the graph of failure time vs. velocity. From this graph, it is clear that a harder material is also not the solution. In order to achieve a time to failure that was sufficient for implementation of this design in a consumer product, a material 10000 times harder than aluminum would have to be used.

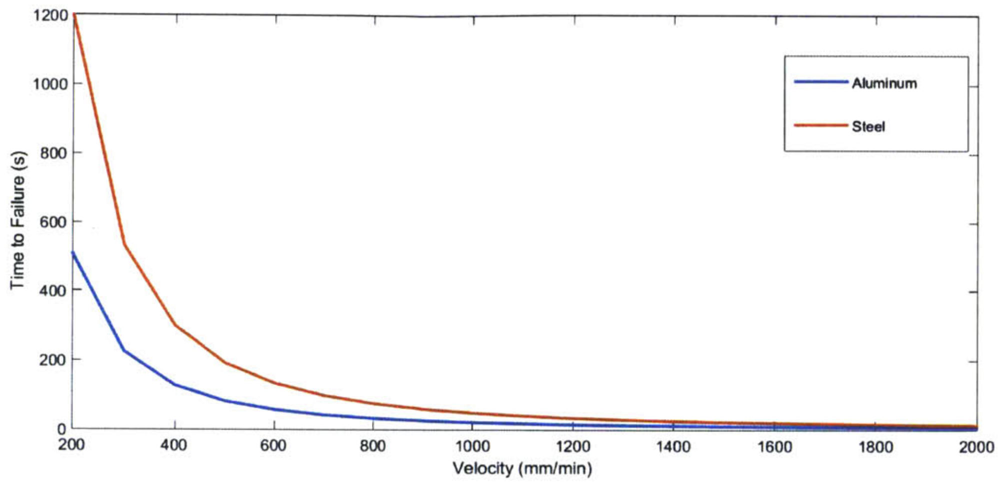

Figure 4-1: Graph showing the relationship between time to failure due to wear and fillament velocity

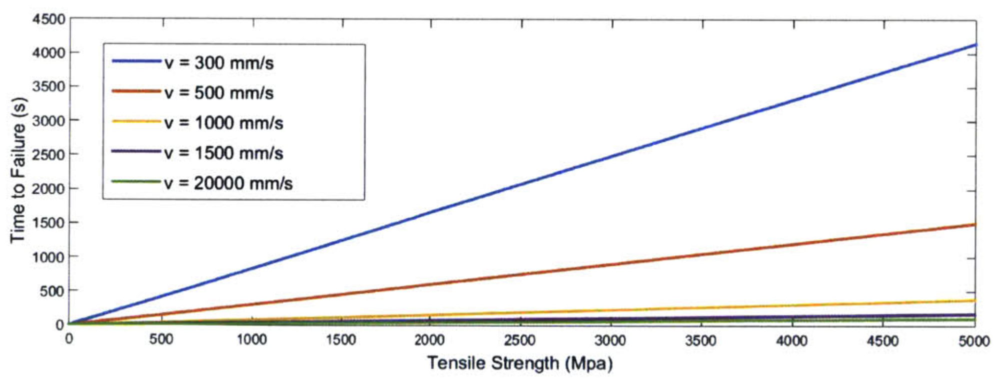

Figure 4-2: Graph showing the relationship between time to failure and hardness

Chapter 5

Conclusion

The design of the counter rotating extruder has potential promises of exciting functionality, but falls short due to massive friction leading to severe abrasive wear and a time to failure that is inadequate for nearly any application. Prior to testing, the extruder had its presumed benefits. Namely, its massive area of contact allows the extruder shafts to withstand extremely high forces in a static state. The high speed of the motors coupled with the large contact allows for rapid filament feed rates unheard of with traditional extruder mechanisms. The design had the potential to extrude filament at over two meters per minute and withstand high amounts of back pressure without jamming.

The physical and mechanical design properties of the extruder that lead to its high feed rate are the same that cause the severe wear that leads to extremely short times to failure. The high speed at which the motors rotate lead to a sliding distance that is very large relative to the worn volume at failure. Furthermore, the greatest attribute of the design, the high speed, causes rapid wear. The wear rate is proportional to the square of the velocity so wear increases dramatically at high speeds.

Due to the nature of the abrasive wear model of this system, there is no simple solution to increase the time to failure so something practical. The wear rate is merely inversely proportional to the hardness, testing the extruder with A2 steel confirmed this. In order for the failure time to reach a satisfactory level, a material several orders of magnitude harder would have to be utilized. The other parameter that could potentially be altered is the wear coefficient, K. This could be done by the introduction of a coating or lubricating fluid; however, the wear rate is proportional to K so the parameter would have to be decreased by several orders of magnitude as well. The combination of a low worn volume to induce failure and a high velocity and sliding distance leads to a form of abrasive wear so severe that it cannot be overcome by choosing a harder material or decreasing the friction. The very property that makes this extruder design so enticing, its ability to maintain a high feed rate and exert large forces on the filament, is ultimately what leads to the extremely high wear rate and makes this design unfeasible for use on a consumer product. In order to create a larger area of contact without a gross increase in sliding distance, a toothed, concave wheel that wraps around the perimeter of the filament is suggested. This design would increase the area of contact between the drive wheel and the filament, while keeping the sliding distance minimal. This simple modification would increase the possible extrusion force without increasing the sliding distance and therefore the wear rate.

Appendix A

Prototype Specifications

A.1 Motor Specifications

Kv	380 rpm/V
Configuration	12N14P
Stator Diameter	35 mm
Stator Length	8 mm
Idle Current	0.4 A
Max Continuous Current	14 A
Internal Resistance	205 m

Table A.1: Motor Parameters

A.2 Shaft Specifications

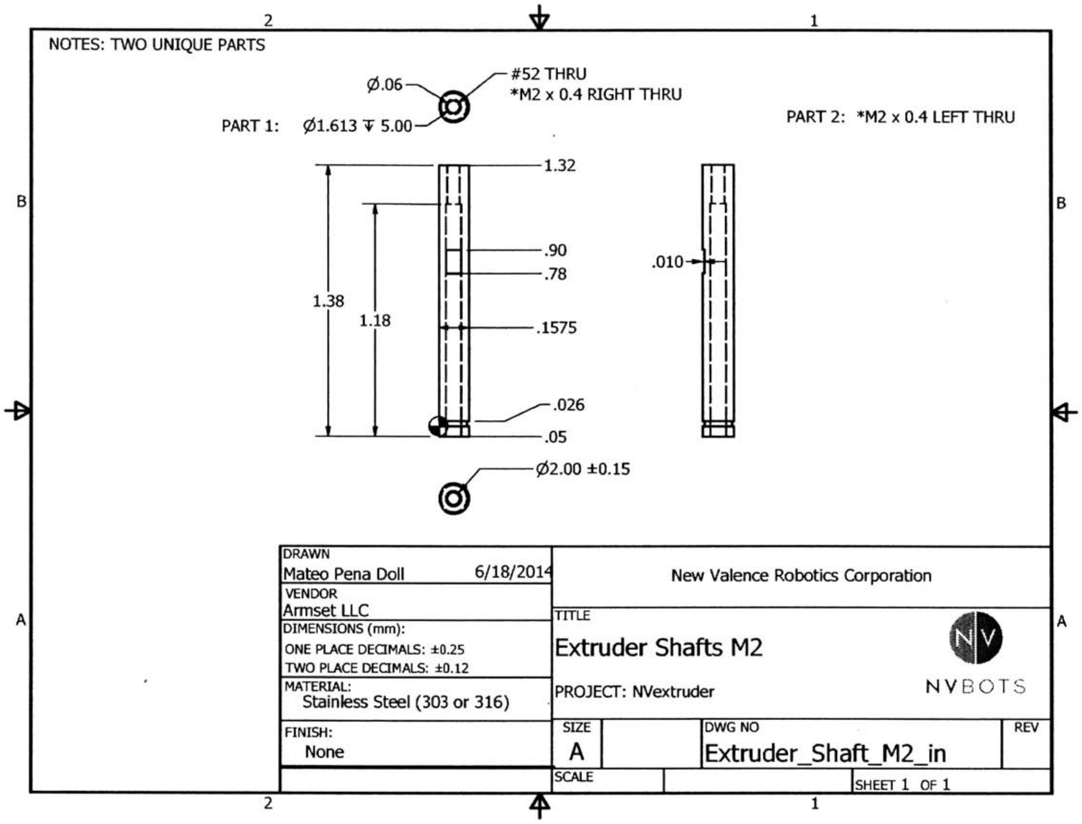

Figure A-1: Dimensioned drawing of extruder shaft

A.3 Motor Housing Specifications

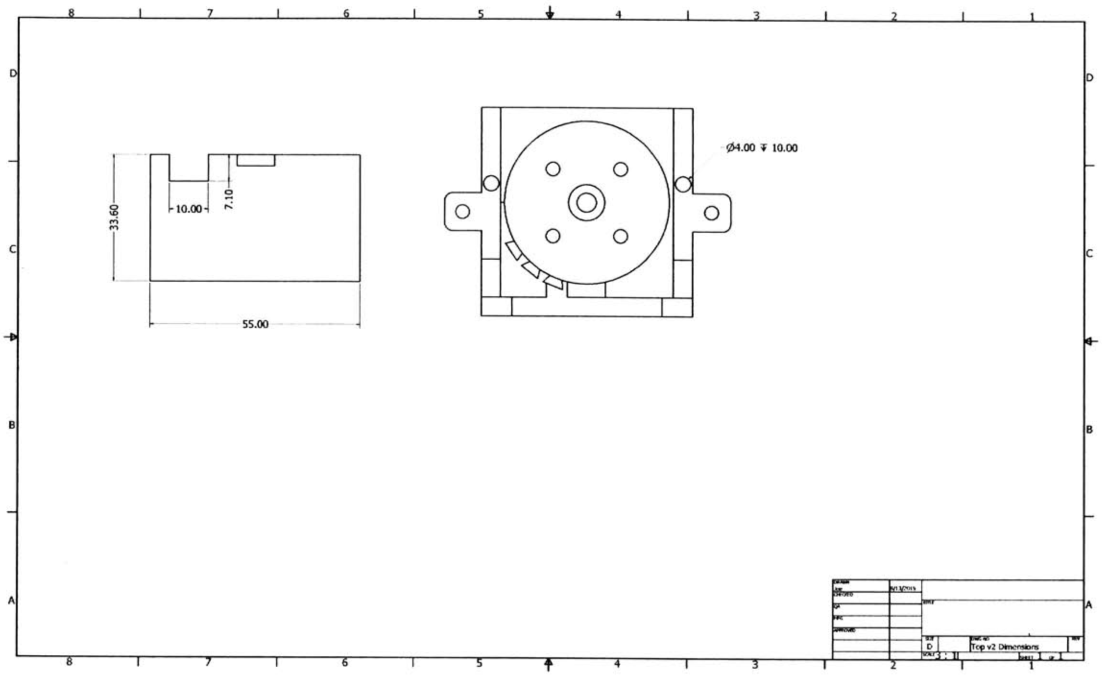

Figure A-2: Dimensioned drawing of extruder motor encasement top

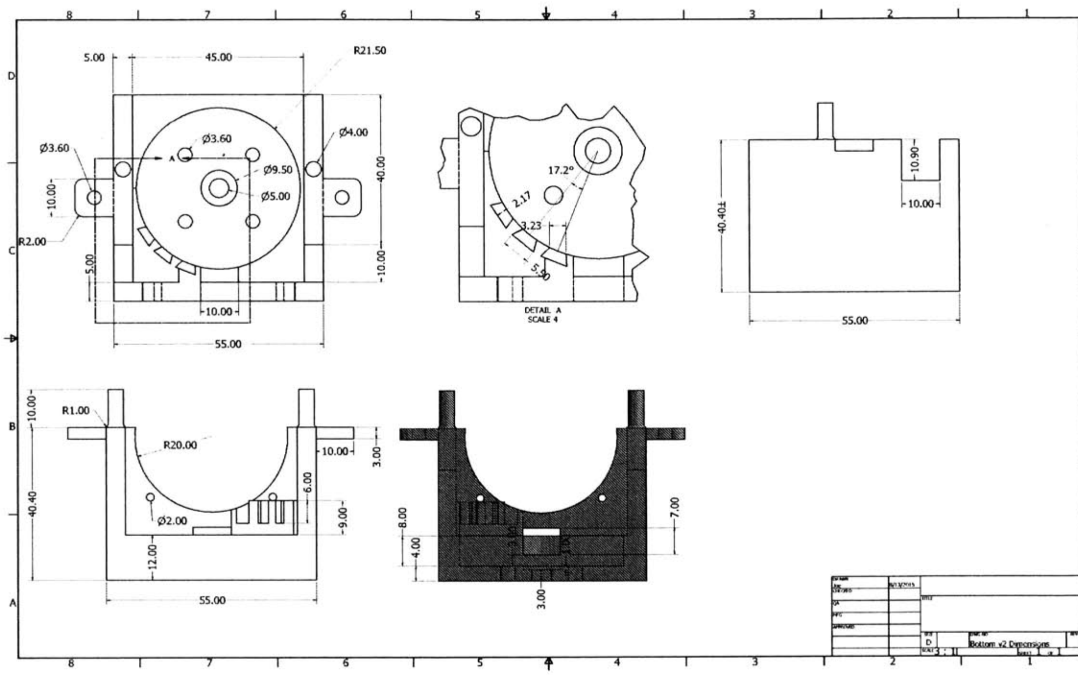

Figure A-3: Dimensioned drawing of extruder motor encasement bottom

Appendix B

Control System

B.1 Wiring Diagram

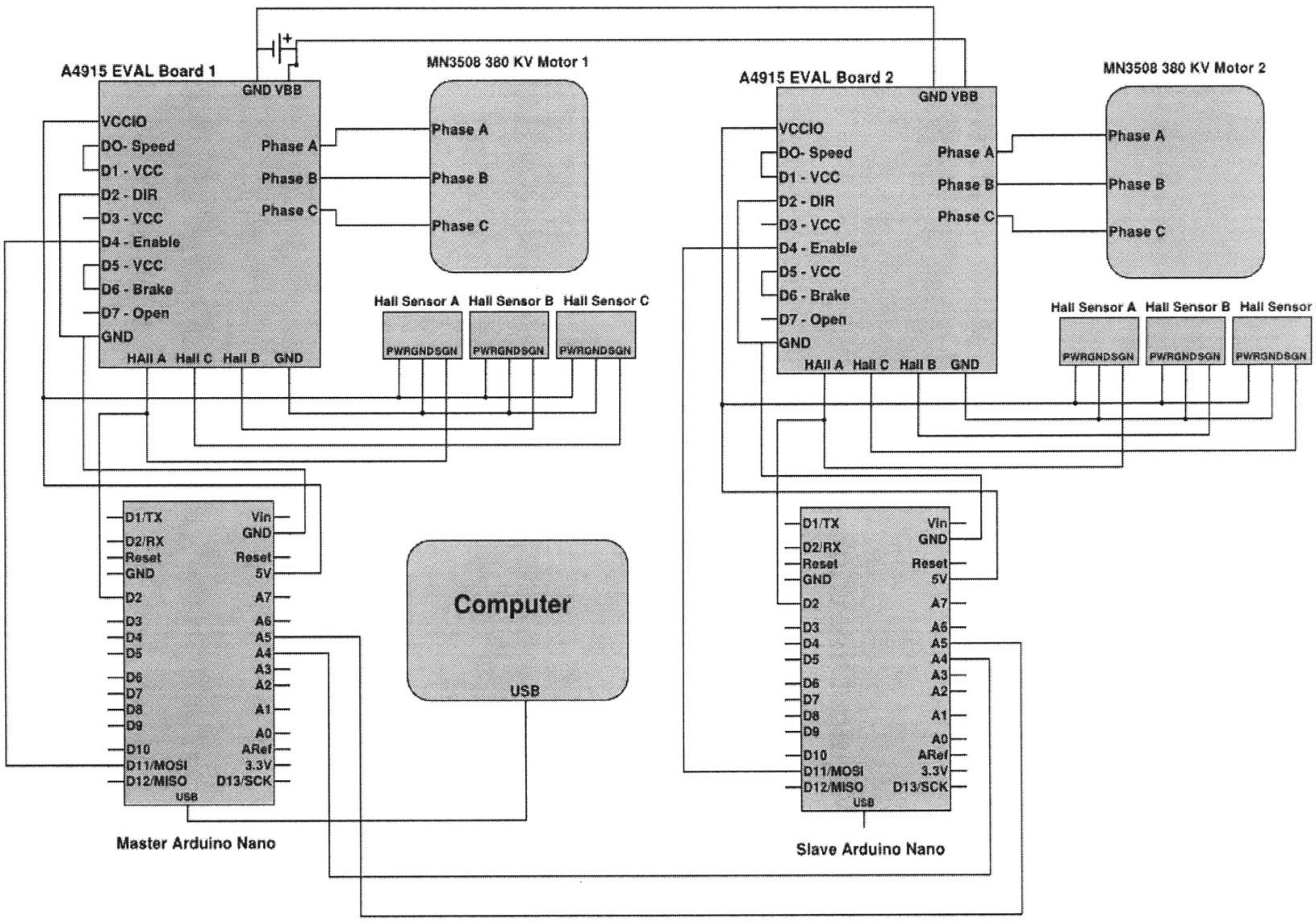

Figure B-1: Wiring diagram of extruder prototype

B.2 Master Arduino Code

```
/* This code controls a brushless DC motor using a PID
   control loop and hall effect sensor rpm sensing.
 *  This code is for the master arduino, which receives
    setpoint commands via serial input, and relays
 *  those commands to the slave arduino. Each arduino runs an
     independent PID control loop.
 */
#include <I2C_Anything.h> // library allowing quick variable
    conversion for I2C
#include <Wire.h>  // library for two wire communication
#include <PID_v1.h>  // PID library
int MotorOut1 = 11; // Pin assignments
int LED = 13;
volatile unsigned long rpmcount; // variable creation
double rpm;
double newrpm;
double halfrpm;
unsigned long timeold;
unsigned long feed_rate;
float pitch;
int pwm;
double Setpoint;
double OldSetpoint;
double kp;
double ki;
double kd;
double Output;
```

```
double Input;
 PID myPID(&Input, &Output, &Setpoint, 0, .05, .001, DIRECT);
     // setup PID controller
//0.02 , .02594, 0.0013
void setup() {
  pinMode (LED, OUTPUT); // signal LED
  Serial.begin(9600);  // initialize serial:
  pinMode(MotorOut1, OUTPUT); // set up pins as outputs/
     inputs
  attachInterrupt(0, rpm_fun, CHANGE); // set up external
     interrupt on pin 2
  digitalWrite(2, HIGH); // Internal pull up resistor
  rpmcount = 0; // set initial values of variables
  rpm = 0;
  timeold = 0;
  pitch = 0.4;
  Input = rpm;
  Setpoint = 0;
  OldSetpoint = 0;
//  TCCR2B = TCCR2B & 0b11111000 | 0x03; // Set PWM frequency
    on timer 2 / pins 3 and 11
// 0x01 - 31372 Hz
// 0x02 - 3921 Hz
// 0x03 - 980 Hz
// 0x04 - 490 Hz
// 0x05 - 245 Hz
  myPID.SetMode(AUTOMATIC); // turn on PID
  myPID.SetSampleTime(100); // controller sample time
  myPID.SetOutputLimits(0, 255); // PID output limits
  Wire.begin(); // join bus as master
```

```
}
void loop() {
 while (Serial.available() > 0) { // read serial if available
    Setpoint = Serial.parseInt();  // look for the next valid
        integer
  if (Serial.read() == '\n') {  // look for the newline. That
      's the end
 Setpoint = constrain(Setpoint, 0, 10000); // constrain the
    values to 0 - 3000
   if (Setpoint != OldSetpoint) { // if a new command is
      entered, send command to slave arduino
        Wire.beginTransmission(8);  // send to serial 8
        I2C_writeAnything (Setpoint);  // send the setpoint
        Wire.endTransmission(); // end
   if (Wire.endTransmission () == 0) // if transmission is
      successful, turn on LED
          digitalWrite (LED, HIGH);
        else
          digitalWrite (LED, LOW);
      }
      OldSetpoint = Setpoint; // reset setpoint for
         comparrison later
      Serial.println(Setpoint);   // print the setpoint in a
         new line
   }
  }
  // RPM counting code
  if (rpmcount >= 7 && (millis() - timeold >= 5)) { // lower
     bound based on rotation and time
    newrpm = (4285 * rpmcount) / (millis() - timeold); //
```

```
       Calculates RPM based on number of magnets
    // rpm = (60000*rpmcount)/(#polepairs*(millis()-timeold))
     timeold = millis(); // update time
    rpmcount = 0; //reset counter
    // feed_rate = pitch * rpm; // calculate feed rate
    // Serial.println(rpm,DEC); // print RPM values
    // Serial.print(" rpm        ");
    // Serial.print(feed_rate,DEC); // print feed rate values
    // Serial.println(" mm/min");
    Serial.print(rpm); // print real time rpm
    Serial.print("           ");
    Serial.println(Output); // print pwm output from
       controller
  }
  if (newrpm <= 10000) // filter out erroneous rpm values
    rpm = newrpm;
  else
    rpm = rpm;
  Input = rpm; // input measured rpm to controller
  myPID.Compute(); // runs controller
  analogWrite(MotorOut1, Output); //write PWM value to motor
     pin
}
void rpm_fun() // interrupt function
{ rpmcount++; // counter
}
```

B.3 Slave Arduino Code

```
/* This code controls a brushless DC motor using a PID
   control loop and hall effect sensor rpm sensing.
 *  This code is for the slave arduino, which receives
    setpoint commands from the master arduino.
 *  Each arduino runs an independent PID control loop.
 */
#include <I2C_Anything.h> // library allowing quick variable
    conversion for I2C
#include <Wire.h> // library for two wire communication
#include <PID_v1.h> // PID library
int MotorOut1 = 11; // Pin assignments
volatile unsigned long rpmcount; // variable creation
double rpm;
double newrpm;
double halfrpm;
unsigned long timeold;
unsigned long feed_rate;
float pitch;
int pwm;
double Setpoint;
double kp;
double ki;
double kd;
double Output;
double Input;
 PID myPID(&Input, &Output, &Setpoint, 0, .05, .001, DIRECT);
      // setup PID controller
//0.02 , .02594, 0.0013
```

```
void setup() {
 Serial.begin(9600);  // initialize serial:
 pinMode(MotorOut1, OUTPUT); // set up pins as outputs/inputs
 attachInterrupt(0, rpm_fun, CHANGE); // set up external
    interrupt on pin 2
 digitalWrite(2, HIGH); // Internal pull up resistor
 rpmcount = 0; // set initial values of variables
 rpm = 0;
 timeold = 0;
 pitch = 0.4;
 Input = rpm;
 Setpoint = 0;
//  TCCR2B = TCCR2B & 0b11111000 | 0x03; // Set PWM frequency
    on timer 2 / pins 3 and 11
// 0x01 - 31372 Hz
// 0x02 - 3921 Hz
// 0x03 - 980 Hz
// 0x04 - 490 Hz
// 0x05 - 245 Hz
 myPID.SetMode(AUTOMATIC); // turn on PID
 myPID.SetSampleTime(100); // controller sample time
 myPID.SetOutputLimits(0,255); // PID output limits
 Wire.begin(8); // join bus as slave 8
 Wire.onReceive(receiveEvent); // function to run when
    command received
}
 void loop() {
if (rpmcount >= 7 && (millis()-timeold >= 5)) { // lower
   bound based on rotaion and time
    newrpm = (4285*rpmcount)/(millis()-timeold); //Calculates
```

```
    RPM based on number of magnets
  // rpm = (60000*rpmcount)/(#polepairs*(millis()-timeold))
  timeold = millis(); // update time
  rpmcount = 0; //reset counter
 // feed_rate = pitch * rpm; // calculate feed rate
 // Serial.println(rpm,DEC); // print RPM values
 // Serial.print(" rpm       ");
 // Serial.print(feed_rate,DEC); // print feed rate values
 // Serial.println(" mm/min");
 Serial.print(rpm); // print real time rpm
 Serial.print("        ");
 Serial.println(Output); // print pwm output from
    controller
}
 if (newrpm <= 10000)  // filter out erroneous rpm values
  rpm = newrpm;
  else
  rpm = rpm;
Input = rpm; // input measured rpm to controller
Setpoint= Setpoint; // defines setpoint for controller
myPID.Compute(); // runs controller
analogWrite(MotorOut1, Output); //write PWM value to motor
  pin
}
void rpm_fun() // interrupt function
{rpmcount++; // counter
}
void receiveEvent(int howmany) // function when command
   received
{
```

```
  if (howmany >= (sizeof Setpoint)) // wait for entire
     string
  {
    I2C_readAnything (Setpoint); // write received string
       to setpoint
  }
}
```

Bibliography

[1] John G. Brisson II Ernest G. Cravalho, Joseph L. Smith and Gareth H. McKinley. *An Integrated Approach to Thermodynamics, Fluid Mechanics, and Heat Transfer*, chapter 9. Oxford University Press, 2005.

[2] Koji Kato and Koshi Adachi. *Modern Tribology Handbook*, chapter 7. CRC Press, 2001.

[3] MatWeb. Aluminum 6061-t6, May 2016. http://www.matweb.com/search/datasheet.aspx?matguid=

[4] McMaster-Carr. A2 steel, May 2016. http://www.mcmaster.com/standard-steel-rods/=12cvf2n.

[5] Rep Rap. Extruders, September 2015. http://reprap.org/wiki/Extruder.

[6] Rep Rap. Fused filament fabrication, January 2015. http://reprap.org/wiki/Fused$_{f}$*ilament*$_{f}$*abrication.*