Mechanical Design and Learned Control System Development of Fiber Extrusion Device on Industrial Programmable Logic Controller (PLC) Platform.

By
Gazi S. Sakib

B.S.
SUNY Stonybrook University, 2018

Submitted to the Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING IN MECHANICAL ENGINEERING
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAY 2024

Authored by: Gazi S. Sakib
Department of Mechanical Engineering
May 18th, 2024

Certified by: Dr. Brian W. Anthony
Principal Research Scientist, Thesis Supervisor

Accepted by: Nicolas Hadjiconstantinou
Chairman, Department Committee on Graduate Theses

Mechanical Design and Learned Control System Development of Fiber Extrusion Device on Industrial Programmable Logic Controller (PLC) Platform

by

Gazi S. Sakib

Submitted to the Department of Mechanical Engineering on May 18, 2024
In partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Mechanical Engineering

ABSTRACT

Optical fibers are ubiquitous in the 21st century as they form the backbone of the internet and electronic communication and enable a global village to exist. Optical fibers play a pivotal role in modern technology and communication for several reasons. They enable high speed data transmission over large distances, while minimizing the data interception. In addition, they are also used in fields like medicine (fiber-optic imaging and endoscopy), sensing technologies (used in temperature, pressure, and strain sensors), and industrial settings (for data transmission and control systems). Therefore, it is of utmost importance that the manufacturing process of optical fibers is better controlled by developing advanced control algorithms that enhance the state-of-the-art PID (Proportional–Integral–Derivative) controllers. This thesis attempts to showcase the work done to establish a framework and a "digital twin" for deploying advanced learned control algorithms on industrial platforms such as Programmable Logic Controllers (PLC) based on machine learning models such as DDPG (Deep Deterministic Policy Gradient). To develop and train such control algorithms, a desktop version of a fiber draw tower was designed, manufactured, and controlled via a PLC. System dynamics data was collected using a readily available preform substitute and the manufactured desktop Fiber Extrusion Device (FrED) was used to train the DDPG-based control algorithms/model. The model was then subsequently tested and compared to state-of-the-art PID algorithms. To that effect, this thesis establishes a framework and enables the path to further develop advanced control algorithms to better control the manufacturing process of fiber optics. This pivotal step promises to significantly enhance the precision and efficacy of optical fiber manufacturing processes, amplifying their impact across industries and technological frontiers.

Thesis Supervisor: Dr. Brian W. Anthony
Title: Principal Research Scientist, Department of Mechanical Engineering

Acknowledgments

Firstly, I would like to show my deep appreciation to my family for their unconditional love and support throughout my journey at MIT. A special thanks goes to my father, A.R. Gazi, and my mother, Kaniz Fatema, for teaching me the value of hard work and instilling in me the confidence to aim high in my life. I would also like to thank my sisters, Saima Sultana and Nusrat Sultana for their support that enabled me to embark on this journey at MIT. I would like to show my special appreciation to my wife, Kashpa Khan, and her family for having confidence in me and supporting me through the ups and downs that come with pursuing a graduate program at MIT.

Furthermore, I would like to thank my teammates Yutong Zhang and Keeghan Patrick who played critical roles in making this research project successful. Their inventiveness, ingenuity, and team spirit has progressed this research project beyond what was initially thought to be possible. Their individual brilliance and drive have made it possible for us to bring three difficult components of this project (Mechanical Design, Advanced Controls and Machine Learning) together to obtain novel results as discussed in this thesis. In addition, I would like to thank the colleagues and lab mates at the Device Realization Lab and FrED Factory (especially Somesh Jaiswal, Russel Bradley and Wenhao Xu) who provided countless acts of kindness, friendship, and technical guidance throughout my time at MIT.

I would like to show my appreciation to the professors and mentors that played critical roles in making my journey at MIT successful. A special thanks to my advisor Dr. Brian Anthony who spared his invaluable time to provide the team and myself with both technical guidance and high-level research direction on a bi-weekly basis. He balanced providing technical guidance and letting me develop autonomy by allowing me to frame research questions and spearhead research direction. His unique approach to mentorship has allowed me to be a more independent problem solver and gave me the confidence to go into unknown domains and contribute. I would also like to thank my program directors Jose Pacheco and Professor David E. Hardt for accepting me into the graduate program and their continued belief in my success. A special thanks goes to Professor Alexander H. Slocum and the Fall 2023, 2.77 (Precision Product Design) class who provided invaluable technical guidance on the mechanical design aspects of this research work.

MIT has certainly lived up to its reputation associated with the saying “Drinking from a Fire Hose” but the love, support, guidance and mentorship from family, friends, colleagues, mentors, peers and the professors has made it possible for me to not only overcome countless challenges but to thrive in such a demanding but enriching atmosphere.

This page is intentionally left blank.

Chapter 1

1.1 Introduction

Fiber optics has been a cornerstone of modern communication in the Western world and has been increasingly pivotal in the developing world such as in the Asia Pacific region. Fiber optics work on the principle of total reflection (1). The two main components of the optical fiber are the core and the cladding. The core is a thin strand of pure glass or plastic through which light can travel over vast distances. The core is surrounded by cladding which has a lower refractive index relative to the core which helps to keep light encapsulated in the core (1). Addanki et al., also describes the concept of critical angle the following way: “If the light strikes the interface at an angle greater than the critical angle, it will not pass via the other medium” (1). Therein lies the importance of the critical angle which helps to ensure light stays encapsulated in the fiber so data can be transmitted over thousands of miles. According to “Fiber Optics Market Size, Share and Growth Report”, “The global fiber optics market size was valued at USD 8.76 billion in 2022 and is expected to grow at a compound annual growth rate (CAGR) of 6.9% from 2023 to 2030.” (2). The highest demand growth for fiber optics is in the Asia Pacific region. According to “Fiber Optics Market Size, Share and Growth Report”, “Asia Pacific region dominated global market at a revenue share of 28.8% in 2022. Increased technological advancements, widespread adoption in IT & telecommunications, administrative sectors, and development of fiber-integrated infrastructure are attributed to the growth” (2). This increased demand for fiber optics comes with a need to manufacture these fibers with tolerances on the order of ± 1µm. This is a valid need as the process of laying fiber cables (which range from 100s to 1000s of fibers bundled together) involves laying (3 ft - 33 ft) cables and connecting cables (much like Legos) across thousands of miles across ocean beds and landscapes. As a result of this need for precision alignment of cables, precision control of the manufacturing process of these fibers is desired. There are many functional disadvantages of not achieving this precision alignment. They include but are not limited to signal loss, increased attenuation (increased noise), modal dispersion (limits data rate), increased reflections (degrades signal), and damaged fibers leading to data loss. The motivation for this thesis is derived from a need to precisely control the manufacturing process of these fibers. In addition to precision, adaptability is also desired as the manufacturing process/environment is ever changing despite best attempts to control the manufacturing environment. This thesis establishes a framework that shows that learned control algorithms such as DDPG, which shows promise in achieving both precision and adaptability, can be implemented on a desktop fiber extrusion device (FrED). To achieve this goal, this work was divided into three major domains, namely the mechanical design of the PLC FrED, development of the electrical architecture and development of the learned control model. This thesis discusses the author’s contribution in all three domains while acknowledging the fact that this work was done in parallel and in collaboration with the work done by Patrick (5) and Zhang (4), who contributed significantly to develop the learned control model and the electrical architecture, respectively. The work discussed in this thesis is the continuation of the work done by Othman, who had shown that industrial PLC can be used to control stepper motor, DC motor, and heater block, the core components of FrED (9).

1.2 Thesis Overview

The goal of the project described in this thesis is to design and fabricate a relatively cheap device that can be used to test out and develop new adaptive/machine learning control algorithms with a relatively cheap preform material (Surebonder Glue Stick - a derivative of Ethylene Vinyl Acetate). Such validated algorithms can then be further fine-tuned to be deployed in industrial fiber optics manufacturing towers while minimizing controller training/development costs. Chapter 2 provides a background in the fiber manufacturing process as conducted by Sterlite Inc. This chapter also briefly summarizes further motivations, controller deployment architecture and deficiencies of the previous FrED versions and the need for mechanical design improvements. Chapter 3 talks about the mechanical design improvements of the new PLC FrED and the associated design, fabrication and testing processes and decisions. Chapter 4 explores the thermal effects on fiber diameter and how the extrusion temperature affects the fiber microstructure. Chapter 5 describes the electrical hardware architecture, the PLC communication methods, and associated block diagram/ladder logic programs. This chapter also details the tuning process and results of the PID controller and explores two different heuristic tuning methods. Chapter 6 details the learned control algorithms that have been implemented and the associated results from both offline and online performance. Chapter 7 summarizes the thesis and puts forward recommendations for future work.

Chapter 2 Background

2.1 Sterlite and Optical Fiber Manufacturing

Sterlite Technologies Ltd. (hereinafter referred to as Sterlite) is a manufacturer of optical fibers based out of India. MIT's Device Realization Lab (DRL), in partnership with Sterlite, is inventing state-of-the-art controller models/algorithms to better control the fiber optics manufacturing process. The fiber needs to be tightly controlled (125 ± 1µm) to ensure large segments of optical cables (made from bundling optical fiber) can be connected seamlessly while the optical fiber is laid across ocean beds and vast geographic landscapes. The fiber optics manufacturing process is rather complex and needs to be tightly controlled due to the need for precision alignment of the cables. The manufacturing process can be briefly described in the following way.

The foundation of optical fiber lies in high-grade silica, undergoing a critical process known as soot deposition, pivotal in shaping the optical qualities of the resulting fiber. Soot deposition involves Open Vapor Deposition, where Silicon Chloride ($SiCl_4$) vapor, infused with Germania, is carried via a medium to burners for deposition onto mandrels (9). These coated mandrels move to a sintering machine, heating them below the melting point, allowing particles to merge into a solid glass piece. The sintered glass then swiftly enters a soaking furnace for high-temperature treatment, releasing trapped gases and alleviating thermal stress (11). This process yields a parent preform, which is drawn through a tower, transforming into smaller rods (child preform) (9). Induction furnaces then soften these preforms into core rods, rigorously tested for optical properties (9). Passing inspection leads to the cladding phase, where additional soot layers augment the core rod to its final diameter. Cylindrical rod preforms are continuously fed through a draw furnace, maintaining tension for precise fiber diameter control (11). Throughout extrusion, the preform undergoes a significant diameter shift from centimeters to micron-level fiber, a transformation observed in industrial extrusion towers on the shop floor (refer to Figure 2.1).

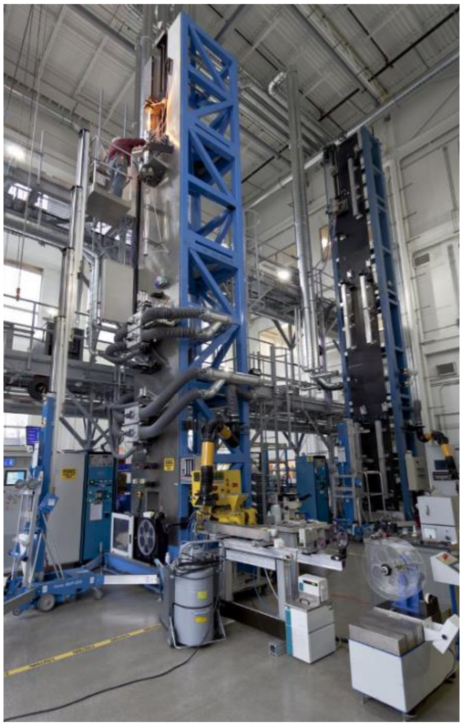


Figure 2.1. Sterlite Optical Fiber Tower

After exiting the furnace, the fiber undergoes cooling in a specialized unit aided by helium injection before being wound onto a capstan, ensuring consistent drawing force (9). The last step involves applying a dual-layer coating onto the fiber surface through ultraviolet curing, finalizing the optical fiber production. Figure 2.2 depicts a simplified representation of the fiber extrusion tower and its components.

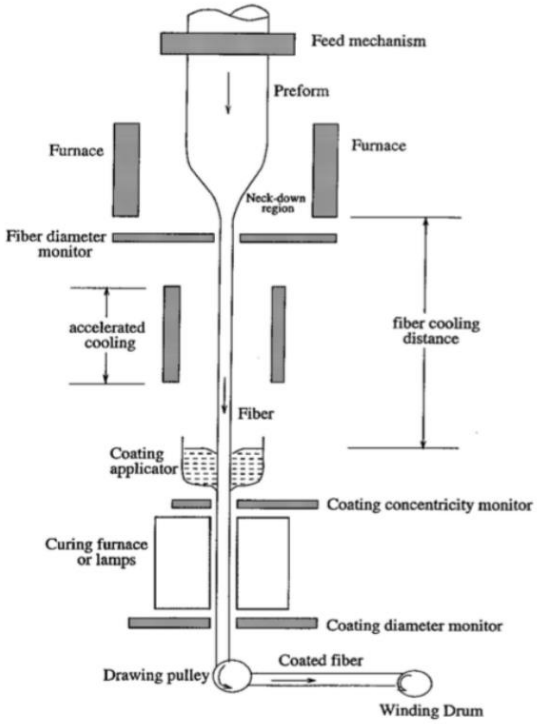


Figure 2.2. Sterlite Optical Fiber Manufacturing Process

There are multiple sensors that take various real-time system parameters which get fed into PID controllers that output the adjusted parameters to various actuators and drivers with the goal of achieving diameter and tension set points. The key factors influencing the Bare Fiber Diameter (BFD) controlled by the draw tower include furnace power, preform entry speed into the furnace, capstan speed for spooling the fiber, and the temperature of the helium used in the fiber cooling process. Feedback for regulating the system relies on the measured fiber diameter and tension. Within the system, three controllers, highlighted in red boxes in Figure 2.3, manage the fiber extrusion process. The tension controller (K_t) operates as a PID controller, using measured tension error to adjust the radiation furnace output. Similarly, the diameter controller (K_d), also a PID controller, utilizes BFD error to modify capstan velocity (4). The preform velocity controller, positioned centrally in Figure 2.3, employs a discrete look-up table to correlate capstan acceleration and slope with pre-form speed (9). Feedback control utilizes tension and BFD as output signals within the system.

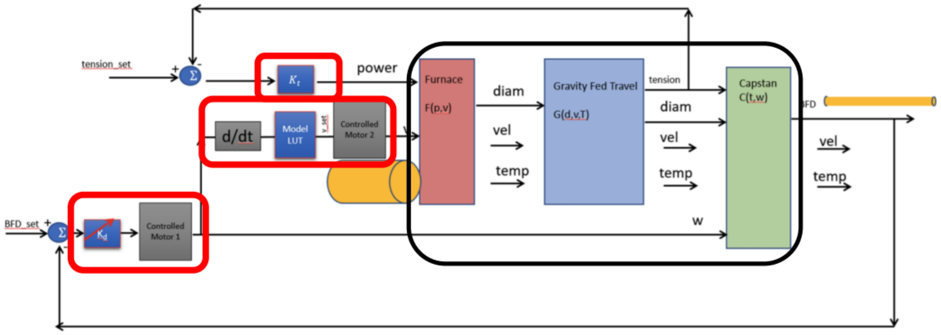


Figure 2.3. Sterlite PID Block Diagram

2.2 Current FrED Development and Motivation.

The Fiber Extrusion Device (FrED) is a desktop device that attempts to recreate the glass optical fiber manufacturing process for the research of control methodologies at the FrED Factory in the Device Realization Lab. A hot glue stick is used as the "pre-form" which is fed into a heater to be melted and extruded using a stepper motor. The extruded fiber then falls and cools in a water bath (or by air cooling) before being measured for diameter using a laser micrometer. It is then spooled using the spooling system, which consists of a DC motor to rotate the spool, and a stepper motor and lead screw to reciprocate the spooling system. Upstream of the spooling system, the heating temperature, fan cooling rate, and extrusion speed are fixed. The DC motor relays the rotation via a timing belt and is controlled by a PID control based on the diameter measurement. The lead screw converts the rotation of the stepper motor into linear traverse motion, together with a nut. The spooling system glides on an 80-20 aluminum extrusion and the 2cm range of motion is limited by two limiting switches at both ends of the aluminum extrusion. The rate of rotation of the stepper motor is not changed.

To be able to test the various learned control algorithms that were being developed for Sterlite in a laboratory setting, using a relatively cheap setup, a Fiber Extrusion Device (FrED) was first manufactured in MIT's Device Realization Laboratory by David Donghyun Kim and Shreya Dhar as part of their Doctoral and Master's work, respectively.

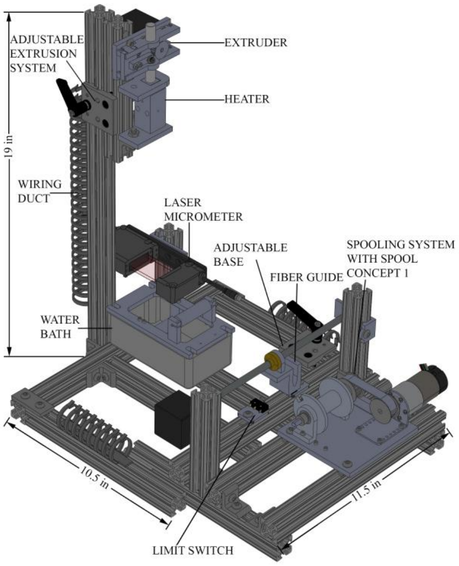


Figure 2.4. Water Based Cooling FrED system.(9)(11)

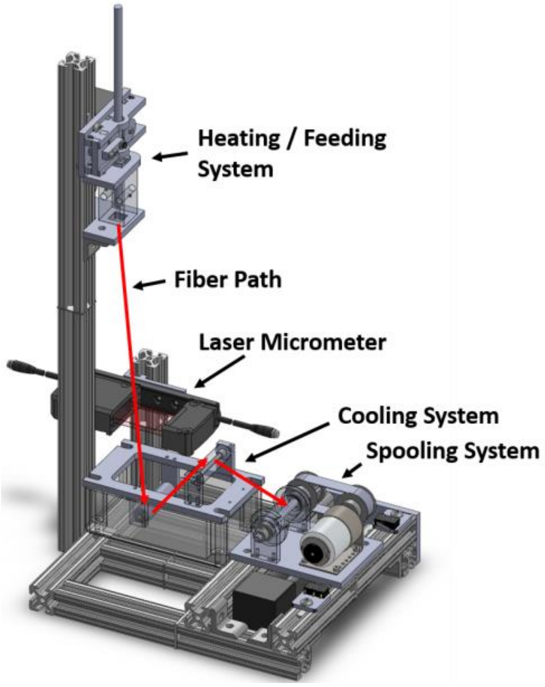


Figure 2.5. Water Based Cooling FrED Fiber Path (Isometric View) (9)(11)

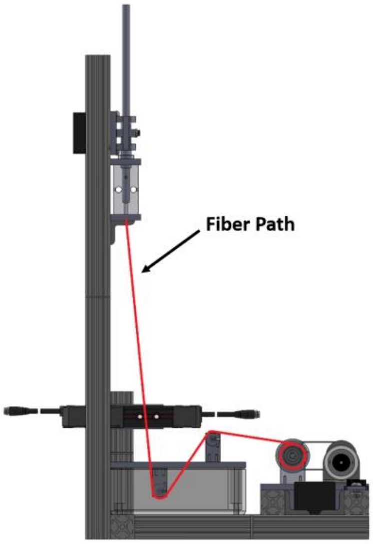


Figure 2.6. Water Based Cooling FrED Fiber Path (Side View) (9)(11)

While this inherited FrED system was a major step in allowing the fiber manufacturing process to be emulated in a laboratory setting, this platform (referred to as the “water bath FrED”) had some limiting factors that were hindering large-scale data collection needed to train a DDPG model. This is summarized in table 2.1.

Table 2.1. Water Bath PLC FrED Inefficiencies

	Undesirable Operating Feature	**Description**
1	Range of Fiber Diameter Produced	Range is on the higher side (0.3 mm – 0.5 mm) compared to actual optical fiber diameter which is on the order of 0.05 mm.
2	High Variation of Fiber Diameter Produced	Variation is on the order of 0.22 mm
3	Heater Block High Frequency Behavior	See section 3.5 and chapter 4.
4	Limited Spooling Volume	Spool Volume is currently 0.66 in^3. Need it to be increased by at least 30%.
5	Long Time needed to begin the Spooling Process	It takes more than 50s to start spooling on the water bath system
6	Limited Spooling Time	This ties back to point 4. This becomes a problem when trying to collect large amounts of data to train a learned model.

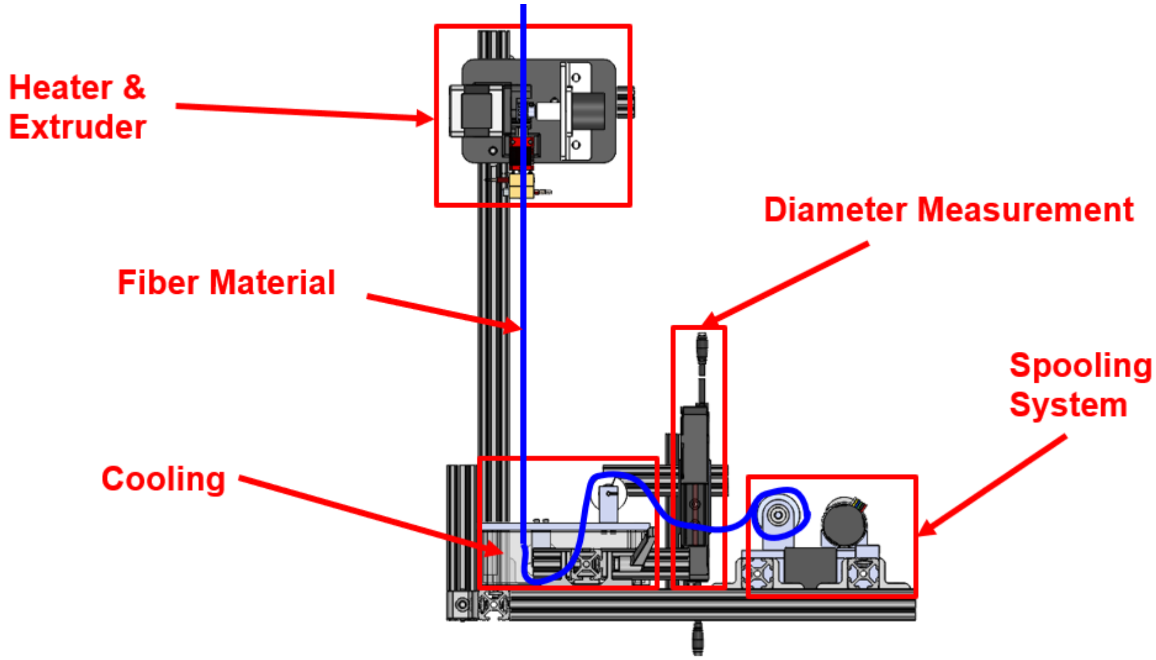


Figure 2.7. Side view of the desktop fiber extrusion device (water bath system), measuring roughly 60cm (H), 60cm (L), 40cm (W)

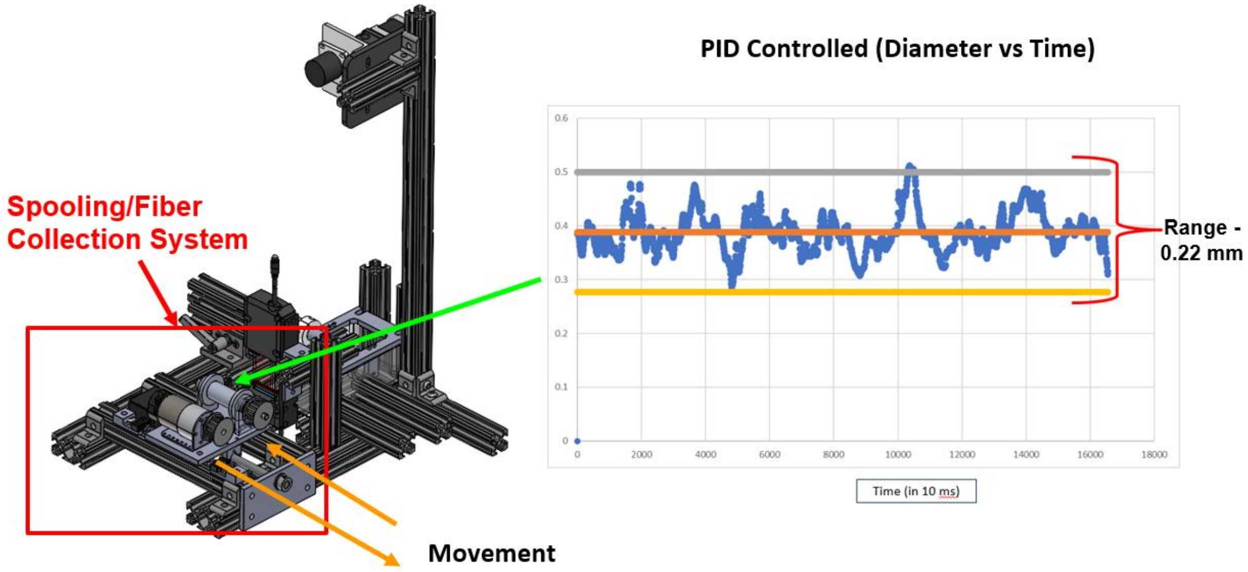


Figure 2.8. Isometric view of the desktop fiber extrusion device (water bath system), measuring roughly 60cm (H), 60cm (L), 40cm (W)

2.3 Controller Deployment Architecture

There are three phases to fully implementing the learned controller.

Phase 1: In this phase an optimized one-time gain setting of the PID controller is implemented on the training platform. It is necessary to increase the range and reduce the variation of the fiber diameter as much as possible to ensure a wide range of data is available to evaluate the performance of the PID and any black box model controllers built on top of that.

Phase 2: In this phase, a black box model controller is trained to perform optimally in a single environment. Usually the training is performed offline (not on the FrED/PLC environment), to take advantage of superior computing resources and limit random performances especially during the early phases of training. After the training, the black box model controller can optimally perform for changing input parameters of the system while the environment remains unchanged. This performance is hypothesized to be better than the PID controller in terms of setpoint change response time, reduced variation, etc.

Phase 3: In this phase, a black box model controller is sufficiently intelligent and adaptive enough that the training of the model can occur online (in the FrED/PLC environment). This phase relaxes the constraint pf the environmental variables being constant. Therefore, the black box model controller can fully adapt to both changing input parameters to the system and the environment within which this system functions. For instance, for the FrED system specifically, this controller would be able to perform optimally when both the input parameters (such as fiber diameter setpoint) and environment (heater temperature, fiber tension, spool traverse speed, perturbations to tension, etc.) are changing.

This research implements phase 1 completely and phase 2 partially. That is, through this research, an optimal PID has been implemented (see section 5.3) and black box model controller has been implemented (see section 6.2-6.8). However, the black box model controller that has been implemented in this paper is not optimal relative to the PID controller due to limited resources and data availability.

2.4 FrED System Deficiencies

The overarching problem that the FrED faces is the variation in the diameter of the produced fiber, shown in the above section in Figure 2.9, which is one of the representations of the quality of the fiber in the industry. Even with the PID control in place, the relatively long response time of 9-10 seconds means that there is a need to control variation due to hardware. The reduction of the response time via novel means is not within the scope of this project.

Backlash was identified as one of the most apparent causes of the diameter variation. It was quantified by manually moving the lead screw in the manner shown below. The backlash is estimated to be 1.51mm or 3.78% of the total required travel of the spool of 40mm. There is therefore the need to reduce backlash.

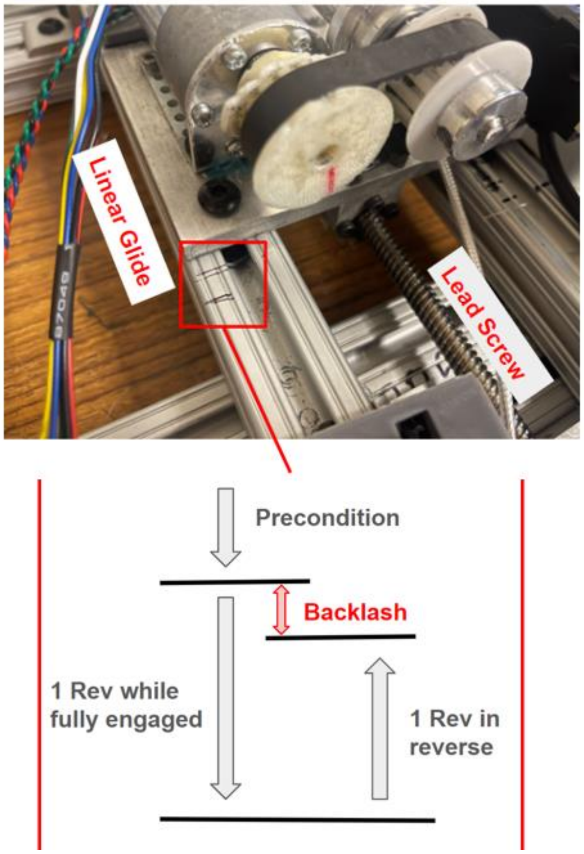


Figure 2.9. Lead Screw Linear Traversing System (Has Backlash)

As shown above, the linear guide is also made of 80-20 aluminum extrusion, which could contribute to periodic stoppages, or reduce the life of the lead screw due to friction.

Subsequent Fast Fourier Transform (FFT) performed on the fiber diameter vs time graph (Figure 2.10) also showed low frequencies of about 0.15-0.2 Hz having high amplitudes and was used to look for potential causes other than the ones mentioned above.

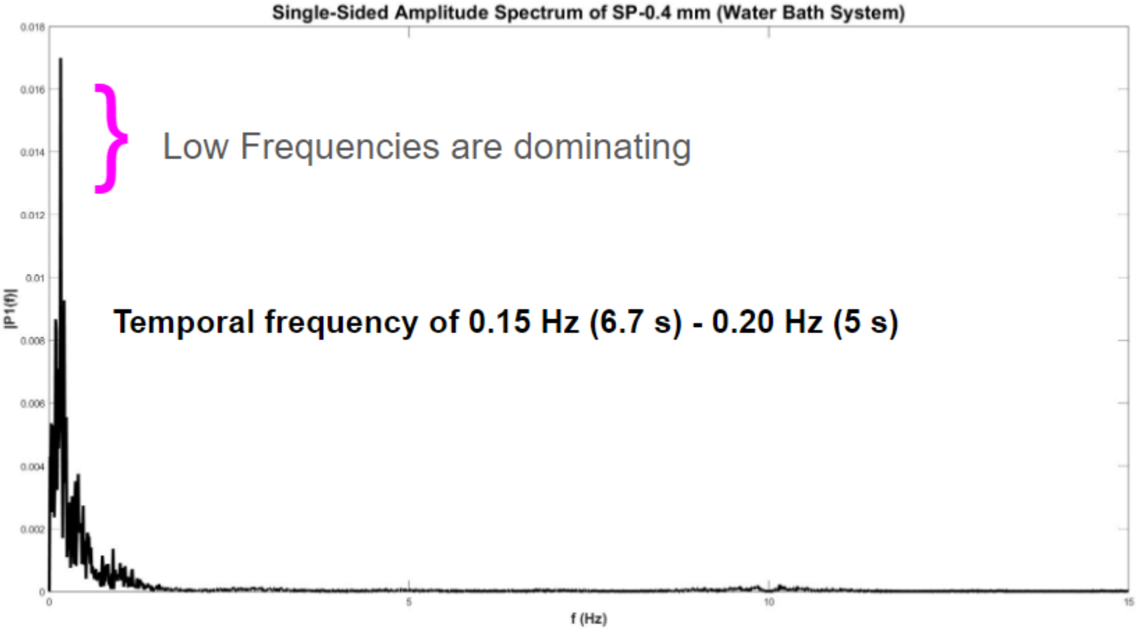


Figure 2.10. FFT of Water Bath Architecture Diameter Data

While using the FrED device, it was also identified that the process to initiate the spooling process is taking too long, at about 4 minutes, as the user would need to guide the fiber horizontally through a series of pulleys which are partially enclosed in the water tank with no side access, as shown in Figure 2.11. The fiber would often break, or being too slow, would cause accumulation of the fiber, which requires restarting the spooling process.

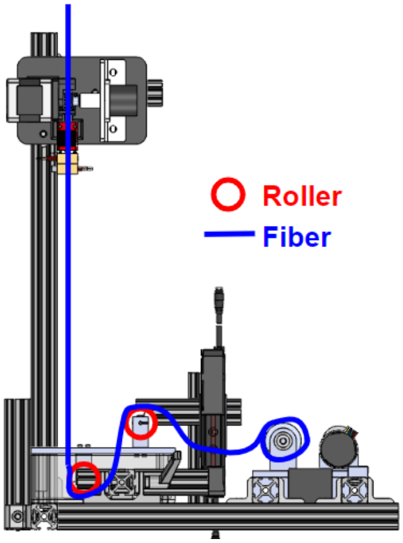


Figure 2.11. Side View of Water Bath FrED

The spool also has a small volume such that this process needs to be done fairly frequently after stopping the device and removing the spooled fiber, which is also difficult to do since the spool is not removable, as shown in Figure 2.12.

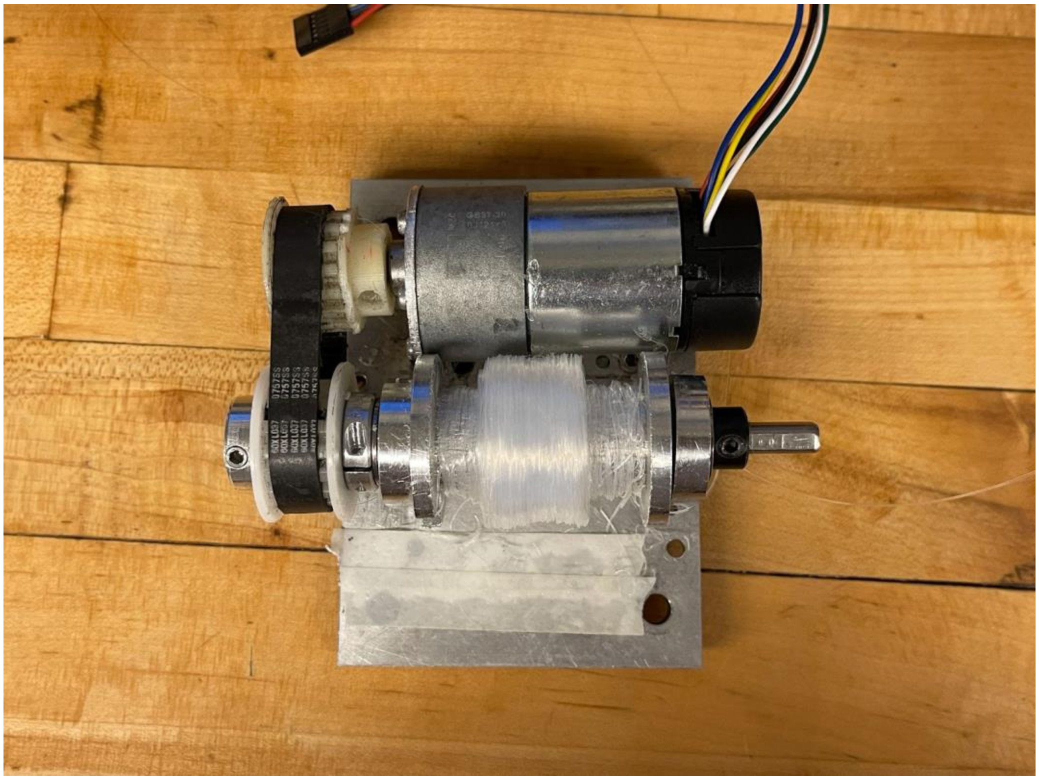


Figure 2.12. Top View of Old Spooling Platform

Chapter 3 PLC FrED Mechanical Design

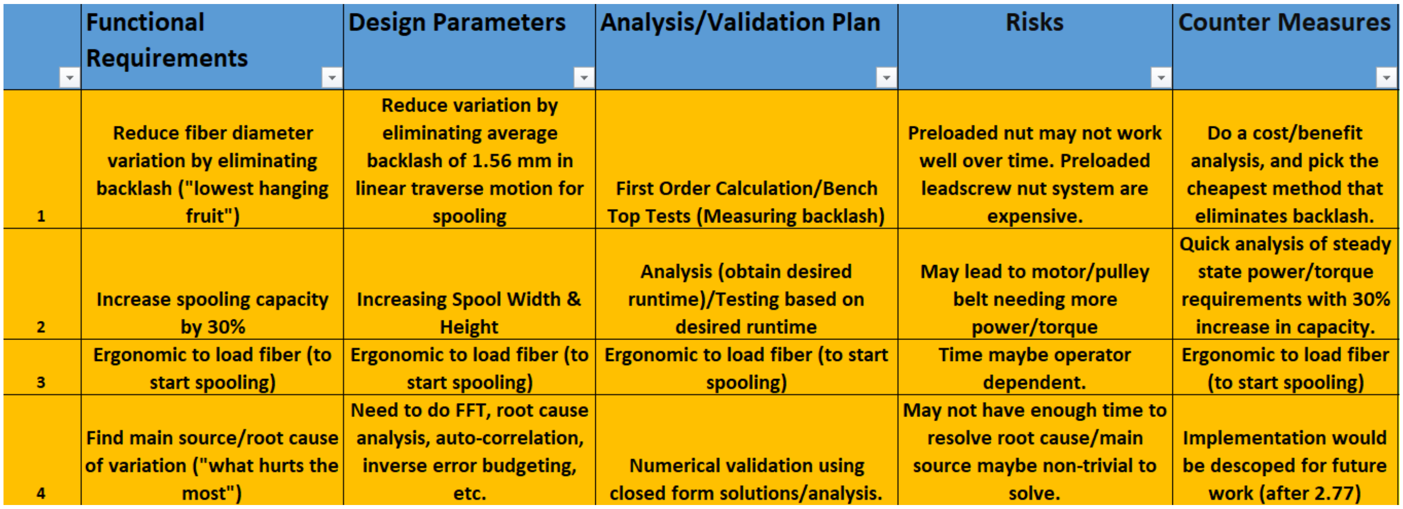
3.1 Motivation for Mechanical Design Changes

In this chapter, some of the hardware inefficiencies (of older FrED systems) as described in section 2.2 are addressed by derivation of new functional requirements, predicting risks and countermeasures, and designing, analyzing, and manufacturing new hardware as shown in table 3.1. Further testing of the newly developed hardware quantitatively shows mitigation/improvement of the risks/inefficiencies described in section 2.2 table 2.1. The mechanical design aspect of this research work has progressed through the 2.77 (Precision Product Design) Fall 2023 class. It was decided that the air-cooling system would be chosen to be improved for this research project due to two primary reasons. Firstly, experimentally it has been shown to be more reliable (easier to start the process and keep the process of spooling the fiber going without the fiber breaking). Furthermore, the range of fiber diameter the water bath system can achieve is (0.4-0.6) mm, while the range of fiber diameter the air-cooling system can achieve is 0.16 mm to 0.38 mm.

3.2 Functional Requirements and Design Parameters

Table 3.1 summarizes the main functional requirements, design parameters, analysis/validation plan, risks, and countermeasures (FREDPARC) that are derived from the inefficiencies summarized in section 2.2 table 2.1.

Table 3.1. Initial FREDPARC Table



	Functional Requirements	Design Parameters	Analysis/Validation Plan	Risks	Counter Measures
1	Reduce fiber diameter variation by eliminating backlash ("lowest hanging fruit")	Reduce variation by eliminating average backlash of 1.56 mm in linear traverse motion for spooling	First Order Calculation/Bench Top Tests (Measuring backlash)	Preloaded nut may not work well over time. Preloaded leadscrew nut system are expensive.	Do a cost/benefit analysis, and pick the cheapest method that eliminates backlash.
2	Increase spooling capacity by 30%	Increasing Spool Width & Height	Analysis (obtain desired runtime)/Testing based on desired runtime	May lead to motor/pulley belt needing more power/torque	Quick analysis of steady state power/torque requirements with 30% increase in capacity.
3	Ergonomic to load fiber (to start spooling)	Ergonomic to load fiber (to start spooling)	Ergonomic to load fiber (to start spooling)	Time maybe operator dependent.	Ergonomic to load fiber (to start spooling)
4	Find main source/root cause of variation ("what hurts the most")	Need to do FFT, root cause analysis, auto-correlation, inverse error budgeting, etc.	Numerical validation using closed form solutions/analysis.	May not have enough time to resolve root cause/main source maybe non-trivial to solve.	Implementation would be descoped for future work (after 2.77)

This method of summarizing the key functional requirements, design parameters, analysis/validation plan, risks, and countermeasures are crucial first steps in properly defining the problem. Additionally, the following algorithm that has been derived from SpaceX has been utilized while developing hardware for the project.

Algorithm 1: First Principles Hardware Design Guidelines

1: Are the design requirements correct? Often the inputs are suspect - input loads, input geometry, input interfaces.

2: Can any element be deleted? Can any process be deleted? The best component is no component. The best process (e.g., machining operation, heat treatment, welding) is no process. Can components be re-used, rather than having subtle deviations? Duplicates are far faster to create than new components. It is often worth small mass losses to avoid new parts.

3: Can any element be simplified? Often, simplicity will lead to weight optimizations, often more than what the optimization would itself have removed.

4: Finally, is the component optimized as needed? This won't always be necessary. Often, so long as it closes (works), that’s enough.

3.3 Analysis, Risks, and Counter Measures

The root cause of backlash was analyzed using first-order hand calculations, using fundamental equations and diagrams as shown in Figure 3.1.

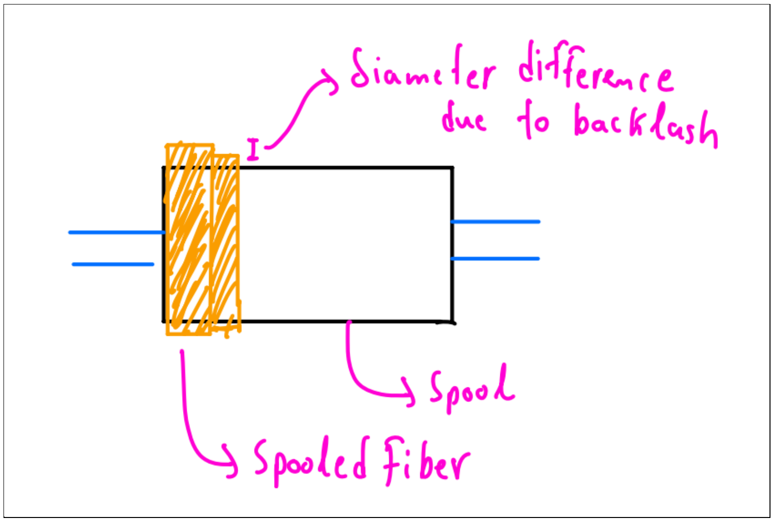


Figure 3.1. Diameter Variation caused due to Backlash.

The hypothesis is that since backlash between the lead screw and the nut causes the spool to be stationary for a certain amount of time, there is some finite amount of time where the spool is not traversing, and the spooling occurs at one location. The exact time for which the spool stays stationary for a backlash of 1.51 mm depends on the spool platform traversing velocity. As a result, there is a difference in the overall diameter of the spool as depicted in Figure 3.1, which causes an instantaneous change in the linear speed with which the fiber is spooled. This undesired change in linear speed causes the fiber to be spooled faster or slower than what the PID controller is commanding and can compensate for (as it has a compensation time of as much as 9-10 seconds). This undesired change in linear speed causes the fiber to be pulled faster/slower which causes the fiber to be smaller/larger (respectively), which is the root cause of variation.

To do a sensitivity analysis, the mass conservation principle is used to relate the fiber diameter with the instantaneous spool diameter, as shown in Figure 3.2.

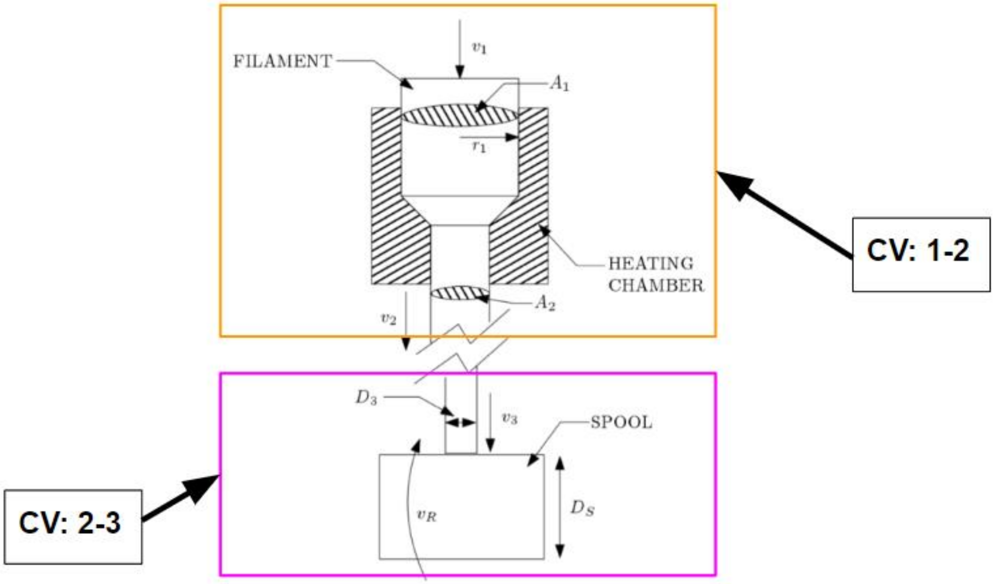


Figure 3.2. FReD Control Volume Schematic

From the first control volume and mass conservation the following relationship can be derived,

$$\dot{m}_1 = \dot{m}_2 \tag{3.1}$$

$$\rho_1 v_1 A_1 = \rho_2 v_2 A_2 \tag{3.2}$$

Assuming densities are constant at the control volume boundaries,

$$\rho_1 \sim \rho_2 \tag{3.3}$$

$$v_1 A_1 = v_2 A_2 \tag{3.4}$$

$$v_3 = 2\pi r_s = \pi D_s v_r \tag{3.5}$$

From the first control volume and mass conservation the following relationship can be derived,

$$\dot{m}_2 = \dot{m}_3 \tag{3.6}$$

$$\rho_2 v_2 A_2 = \rho_3 v_3 A_3 \tag{3.7}$$

Assuming densities are constant at the control volume boundaries,

$$\rho_2 \sim \rho_3 \tag{3.8}$$

$$v_2 A_2 = v_3 A_3 \tag{3.9}$$

$$v_2 D_2^2 = v_3 D_3^2 \tag{3.10}$$

D_3 can be explained as a function of D_2 in the following expression,

$$D_3 = \sqrt{\frac{v_2}{v_3} D_2^2} = D_2 \sqrt{\frac{v_2}{v_3}} \tag{3.11}$$

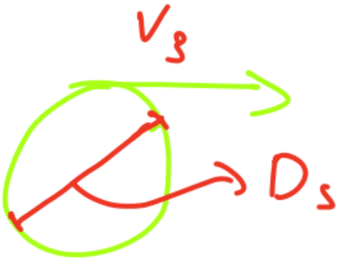


Figure 3.3. FReD Spool Linear Speed (v3)

v_r is related to v_3 in the following way,

$$v_r = \frac{v_3}{\pi D_s} = \frac{4r_2^2 v_2}{D_3^2 \pi D_s} = \frac{D_2^2 v_2}{D_3^2 \pi D_s} \tag{3.12}$$

$$v_r = \frac{v_1 D_1^2}{\pi D_3^2 D_s} \tag{3.13}$$

D_3 can be explained as a function of D_s in the following expression,

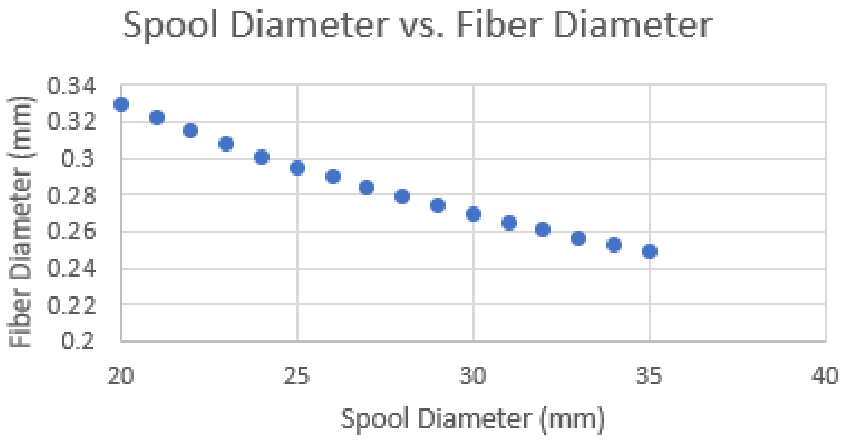
$$D_3 = \sqrt{\frac{60 v_1 D_1^2}{\pi D_s v_r}} \tag{3.14}$$

The description of the variables used in this derivation is explained in table 3.2.

Table 3.2. eq (3.14) Derivation Symbols

Symbol	**Description**
$D_{1,2,3}$	Fiber Diameter at each control volume
$v_{1,2,3}$	Linear Fiber Velocity at each control volume
D_1	Preform Diameter
D_s	Total Spool Diameter
r_s	Total Spool Radius
v_r	Rotation speed of spool (in RPM)
$\rho_{1,2,3}$	Density of the fiber at each control volume
$\dot{m}_{1,2,3}$	Mass flow at each control volume.

The sensitivity of the Fiber Diameter (D_3) as a function of the overall Spool Diameter (D_s) according to eq(3.14) is plotted in Figure 3.4.

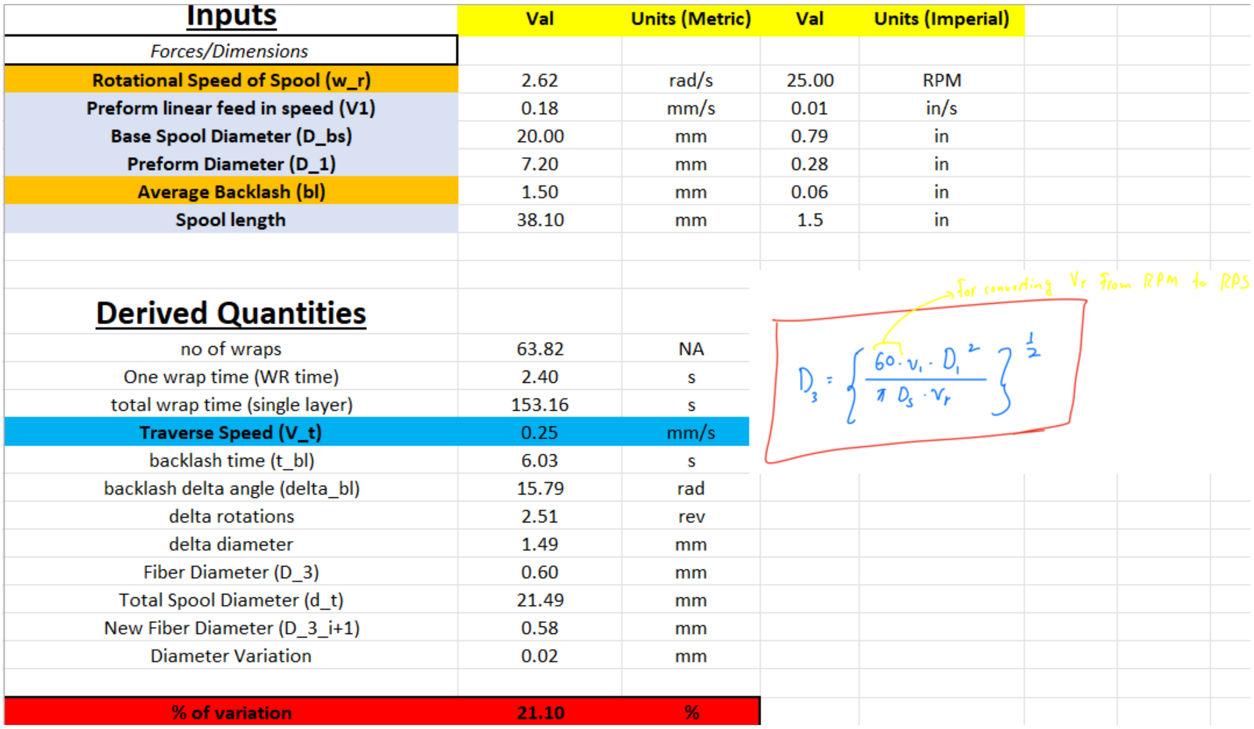


Uncertainty $\pm 0.4mm$

Figure 3.4. Theoretical Spool Diameter as a function of Fiber Diameter.

The sensitivity of the fiber diameter to traverse speed is also investigated via a spreadsheet. Table 3.3 shows the impact of traversing at 0.25mm/s which is obtained when the spooling strategy is to wrap the incoming fiber right next to the previous rotation around the spool. This is typically used in the industry, but because it is slow, there will be more time spent in the backlash zone and therefore has a higher effect on the spool diameter, and hence the fiber diameter, accounting for around 21.1% of the fiber variation.

Table 3.3. Theoretical Variation Improvement (at 0.25 mm/s traverse speed)



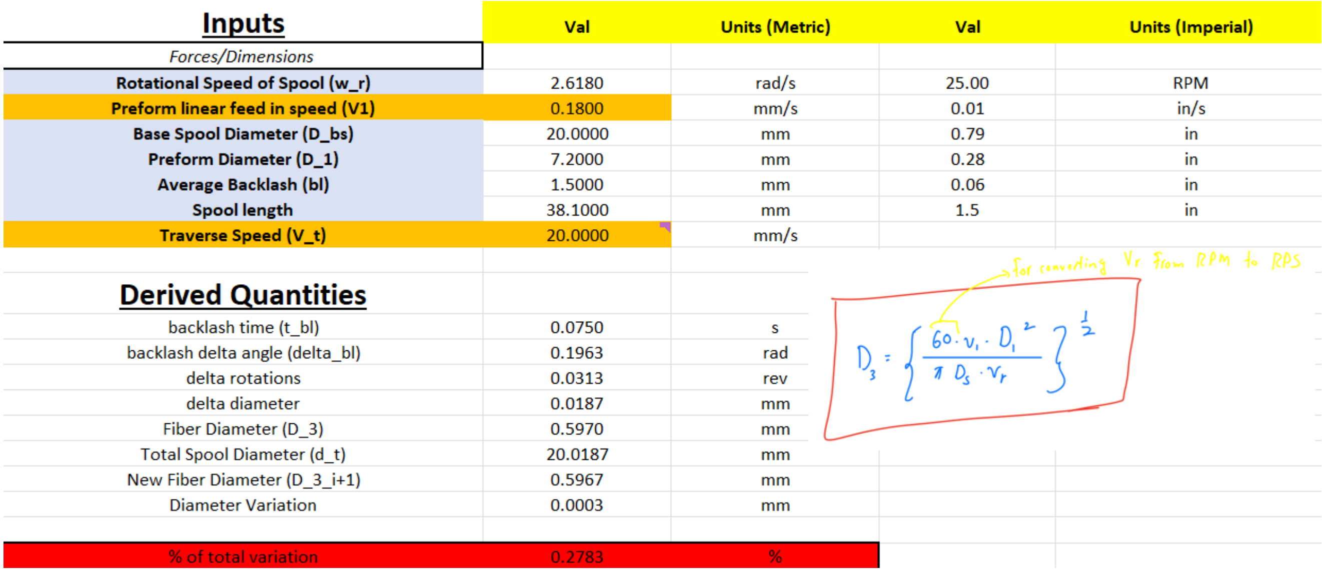
Inputs	**Val**	**Units (Metric)**	**Val**	**Units (Imperial)**
Forces/Dimensions				
Rotational Speed of Spool (w_r)	2.62	rad/s	25.00	RPM
Preform linear feed in speed (V1)	0.18	mm/s	0.01	in/s
Base Spool Diameter (D_bs)	20.00	mm	0.79	in
Preform Diameter (D_1)	7.20	mm	0.28	in
Average Backlash (bl)	1.50	mm	0.06	in
Spool length	38.10	mm	1.5	in
Derived Quantities				
no of wraps	63.82	NA		
One wrap time (WR time)	2.40	s		
total wrap time (single layer)	153.16	s		
Traverse Speed (V_t)	0.25	mm/s		
backlash time (t_bl)	6.03	s		
backlash delta angle (delta_bl)	15.79	rad		
delta rotations	2.51	rev		
delta diameter	1.49	mm		
Fiber Diameter (D_3)	0.60	mm		
Total Spool Diameter (d_t)	21.49	mm		
New Fiber Diameter (D_3_i+1)	0.58	mm		
Diameter Variation	0.02	mm		
% of variation	**21.10**	**%**		

for converting V_r from RPM to RPS

$$D_3 = \left\{ \frac{60 \cdot v_1 \cdot D_1^2}{\pi D_s \cdot v_r} \right\}^{\frac{1}{2}}$$

However, as table 3.3 shows, it is possible to increase the traversing speed of the spool to mitigate the effect of the backlash zone when the traversing speed is significantly faster at 20mm/s. In fact, running at low speeds of 0.25 mm/s proved to be impractical for this particular desktop FrED as it causes the fiber to be wound around the spool loosely due to lack of tension from the low traverse speed.

Table 3.4. Theoretical Variation Improvement (at 20 mm/s traverse speed)



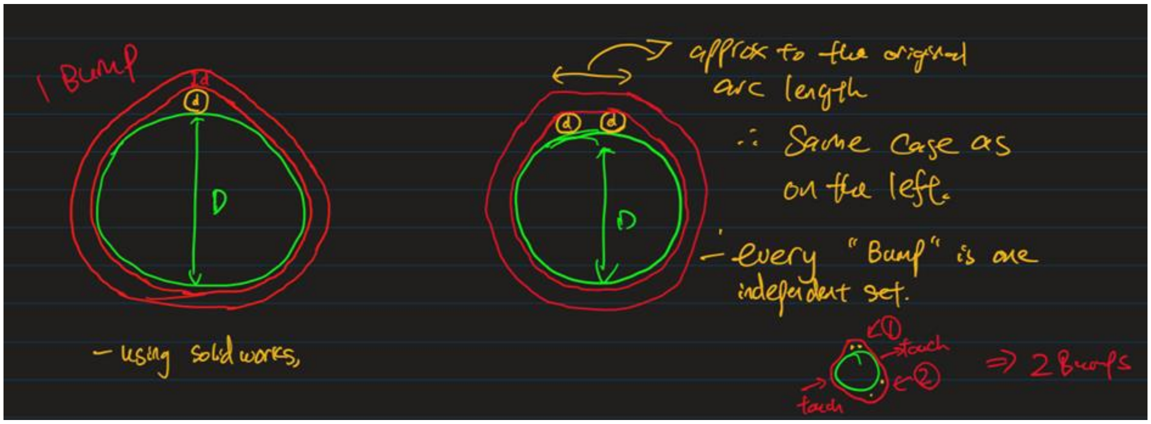
Inputs	Val	Units (Metric)	Val	Units (Imperial)
Forces/Dimensions				
Rotational Speed of Spool (w_r)	2.6180	rad/s	25.00	RPM
Preform linear feed in speed (V1)	0.1800	mm/s	0.01	in/s
Base Spool Diameter (D_bs)	20.0000	mm	0.79	in
Preform Diameter (D_1)	7.2000	mm	0.28	in
Average Backlash (bl)	1.5000	mm	0.06	in
Spool length	38.1000	mm	1.5	in
Traverse Speed (V_t)	20.0000	mm/s		
Derived Quantities				
backlash time (t_bl)	0.0750	s		
backlash delta angle (delta_bl)	0.1963	rad		
delta rotations	0.0313	rev		
delta diameter	0.0187	mm		
Fiber Diameter (D_3)	0.5970	mm		
Total Spool Diameter (d_t)	20.0187	mm		
New Fiber Diameter (D_3_i+1)	0.5967	mm		
Diameter Variation	0.0003	mm		
% of total variation	0.2783	%		

for converting V_r from RPM to RPS

$$D_3 = \left\{ \frac{60 \cdot v_1 \cdot D_1^2}{\pi D_s \cdot v_r} \right\}^{\frac{1}{2}}$$

Analysis has been done to investigate the effect of fibers crossing over other fibers because of the faster traversing speed, which in turn results in the fiber being sparsely wound around the spool. As shown in Figure 3.5, when the incoming fiber crosses one or more fibers that lift it away from the spool and touch the spool again, it is considered as one bump. The bridge distance between two adjacent underneath fibers that results in the incoming fiber not touching the spool, as shown on the right side of Figure 3.5 is assumed to be the same as the arc length along the surface of the spool. Therefore, the effect of crossing another fiber would only be dependent on the number of bumps that the incoming fiber has.

Using SolidWorks to measure and compare the distances and using actual values of the fiber diameter (0.4mm) and spool diameter (20mm), the expected change in the length of the incoming fiber is 0.239% larger for every bump, which means it is somewhat negligible for causing a change in linear spooling velocity.



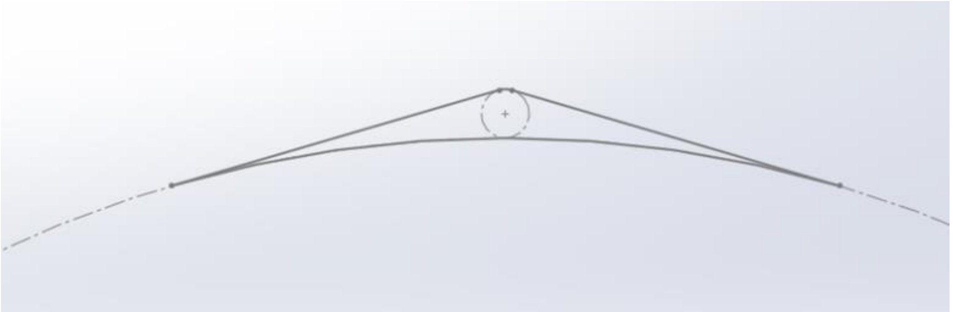


Figure 3.5. Cross Threading Simulation.

Therefore, with the backlash, the short-term solution to reduce variation would be to traverse quickly, but it doesn't eliminate the effects on fiber diameter entirely. To resolve the backlash issue ("lowest hanging fruit") of the traversing platform, there were two primary modes of proposed solutions, namely using a preloaded screw and timing belt. These design options are discussed in detail in section 3.4.

3.4 Hardware Design and Manufacturing.

3.4.1. Overall Design/Manufacturing

There are two main architectures of FrED that have been experimented with in this project. The first is the water bath system which is shown in Figure 2.5. The second is the air-cooling system which is shown in Figure 3.6 and 3.7 below. Section 2.4, 3.1 and 3.2 discussed the mechanical design inefficiency of the previous FrED systems. Section 3.3 delved into the justifying the hypothesis of the root cause variation is backlash in the traversing mechanism of the fiber spool. This section will introduce the new hardware designs that meet the functional requirements of section 3.2 and mitigate the inefficiencies discussed in section 2.4.

The water bath based FrED (shown in Figure 2.5) posed the major problem of starting the spool and maintaining spooling for over 5 minutes. The reason ease of beginning the spooling process and maintain the spooling process for over 15 minutes at a time is that a data set on the order of 10^6 is desired to train the learned control models described in Chapter 6. To address this and some of the other inefficiencies detailed in table 2.1, a design that minimized the change in fiber direction was desired as it not only mimicked industrial optical fiber towers (Figure 2.1), but it also minimized working against gravity (as opposed to the water bath shown in Figure 2.5). This design philosophy has two main advantages. Firstly, as it allowed for a much smaller size of the fiber to be manufactured (on the order of 0.15 mm – 0.35 mm) as opposed to (0.3 mm – 0.8 mm) as not a lot of tension is needed to change direction when working with gravity. Secondly, it allowed the spooling process to be started much faster. With the water bath design, the water container was quite small, and it was difficult to get the fiber to start wrapping around the rollers (as shown in Figure 2.12). The new design solved many of these issues (shown quantitatively in table 4.2). The CAD of the new PLC FrED is shown in Figure 3.6.

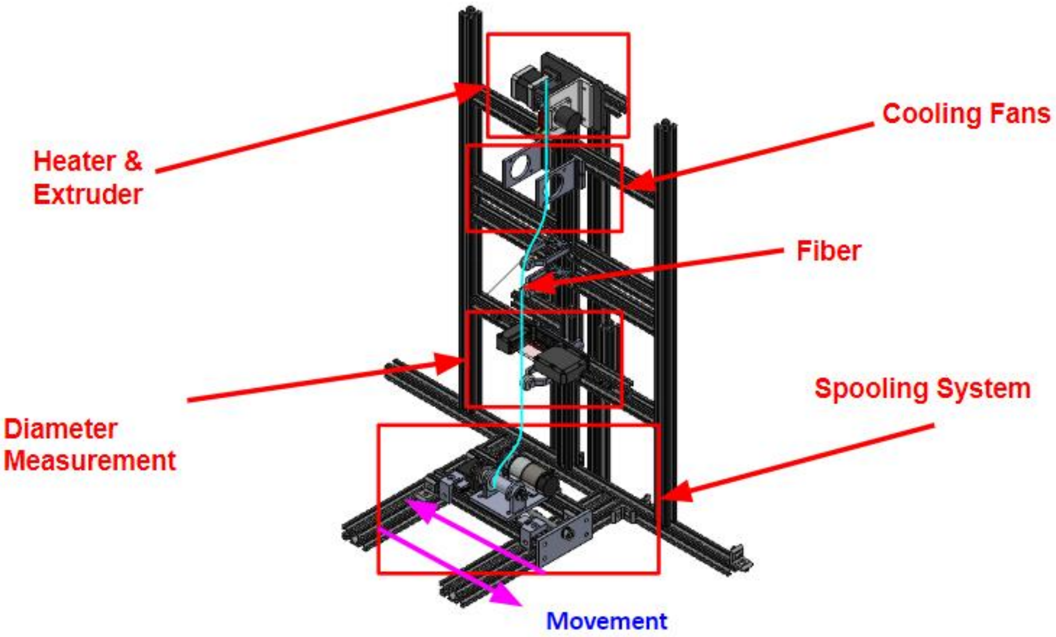


Figure 3.6. Side view of the desktop fiber extrusion device, measuring roughly 73.7 cm (H), 35.6 cm (L), 71.1 cm (W)

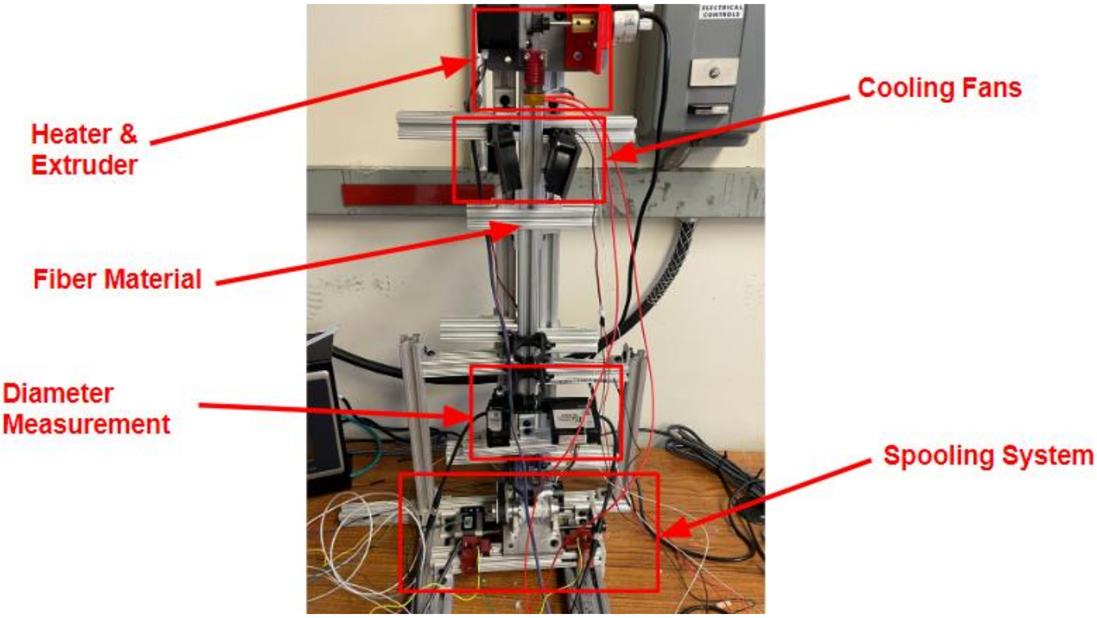


Figure 3.7. Front View of Air Cooling FrED

The new PLC FrED hardware was manufactured using Aluminum 6061-T6, 80/20 structural framing and various off the shelf hardware such as gears, shafts, timing belts, bushings, fasteners, etc. The full bill of materials and CAD structure breakdown can be found in Appendix E.

3.4.2. Preform Assembly Design Changes/Slippage Factor

One of the key issues that was solved using a combination of stepper motor micro stepping and mechanical design is the slippage problem that was especially evident when an encoder had been installed as shown in Figure 3.8. The performance of the pre-form assembly (shown in Figure 3.9) was crucial in determining the quality of the fiber that was being manufactured and the success to which learned control algorithms can be deployed.

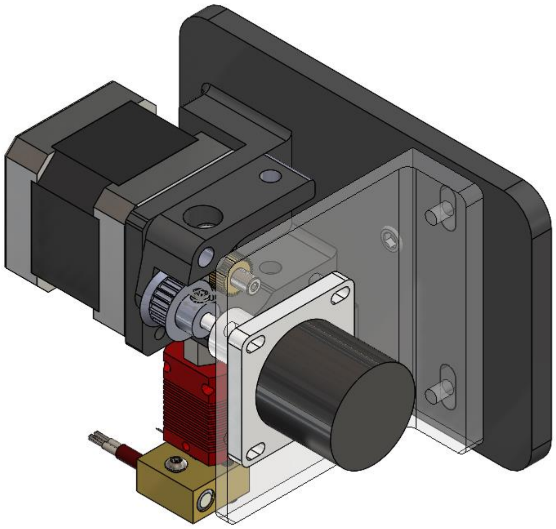


Figure 3.8. Isometric View of the Preform/Heater Assembly

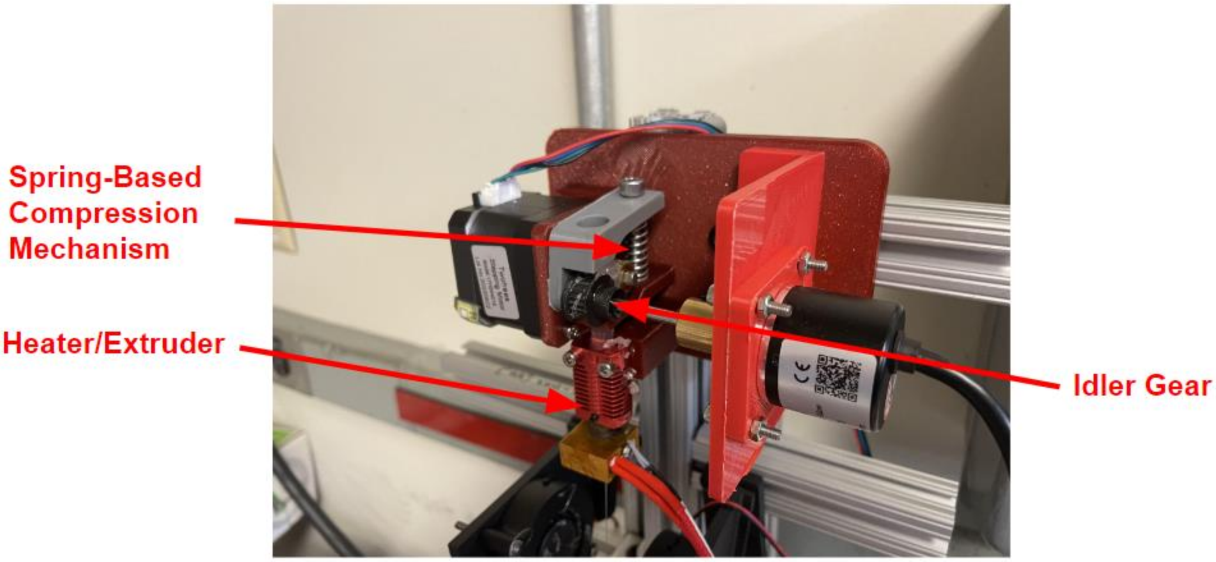


Figure 3.9. Manufactured Isometric View of the Preform/Heater Assembly

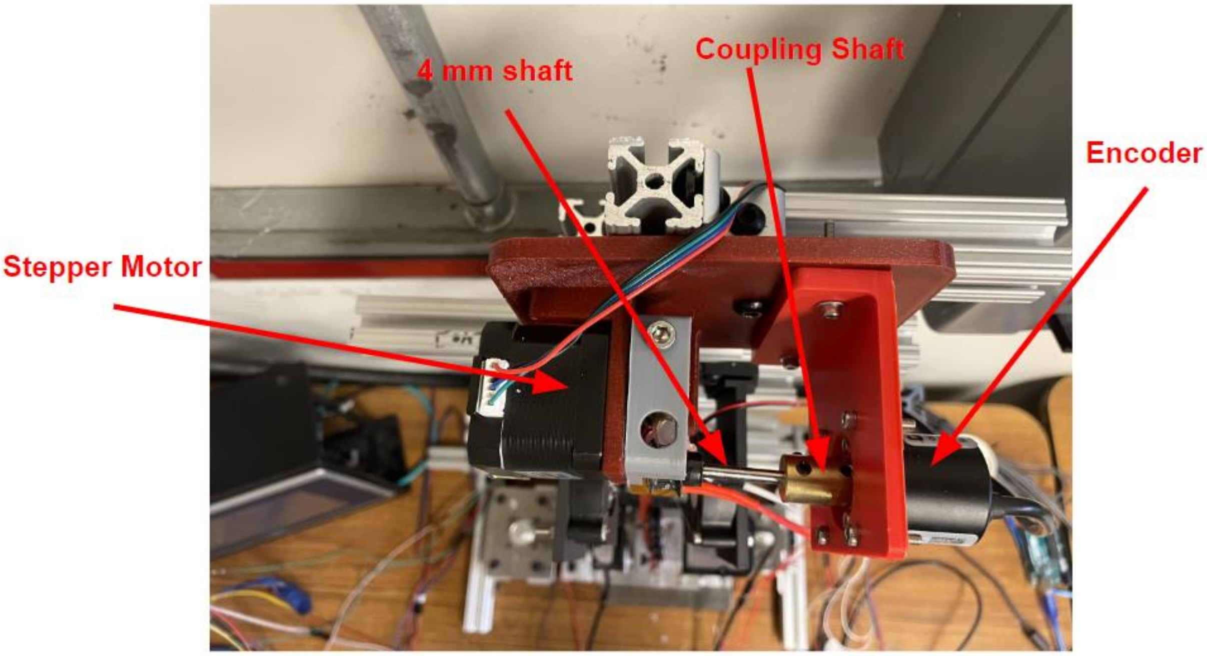


Figure 3.10. Top View of the Manufactured Preform/Heater Assembly

The slippage issue was first noticed when there were disruptions in the pre-form idler gear speed. This happened as the idler gear was rotating too fast and was slipping on the body of the pre-form material which took some finite time to go to the glass transition temperature. It is at this temperature that a certain volume of the preform material would reach a certain viscosity that would allow the idler gear to push the upper parts of the pre-form material into the heating chamber. From this observation, it was concluded that the two main factors contributing to slippage was the stepper motor speed (see Figure 3.11) and the cantilever effect (see Figure 3.12)

The slippage factor can be mathematically expressed as shown in eq (3.15)

$$SF = \frac{v_{r_{theoretical}}}{v_{r_{observed}}} \tag{3.15}$$

Here, $v_{r_{observed}}$ is the linear speed of the pre-form that was measured experimentally. The $v_{r_{theoretical}}$ is obtained from eq().

$$v_{r_{observed}} = \frac{60 v_1 D_1^2}{\pi D_3^2 D_s} \tag{3.16}$$

The dependency of the *SF* on the rotational speed of the main gear of the pre-form motor can be concluded from the plot as shown in Figure 3.11.

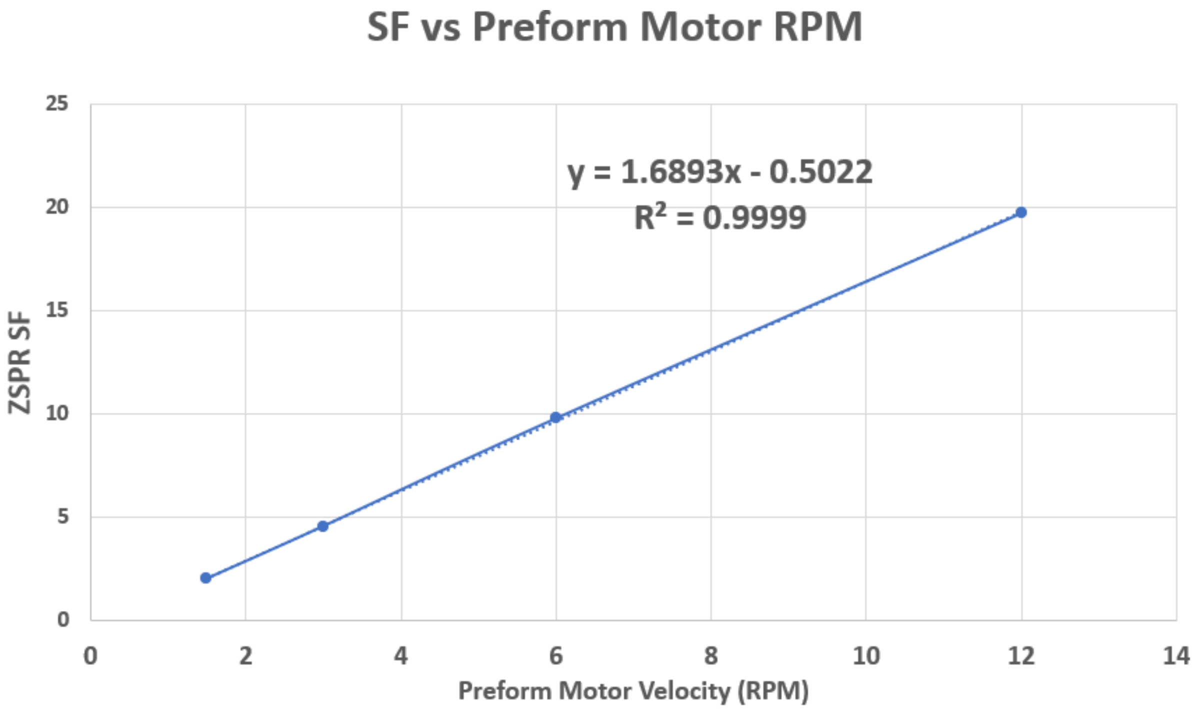


Figure 3.11. Plot of Slippage Factor vs Preform Motor RPM

The second dependency of the *SF* comes from the cantilever effect as shown in the free body diagram depicted in Figure 3.12 and Figure 3.13.

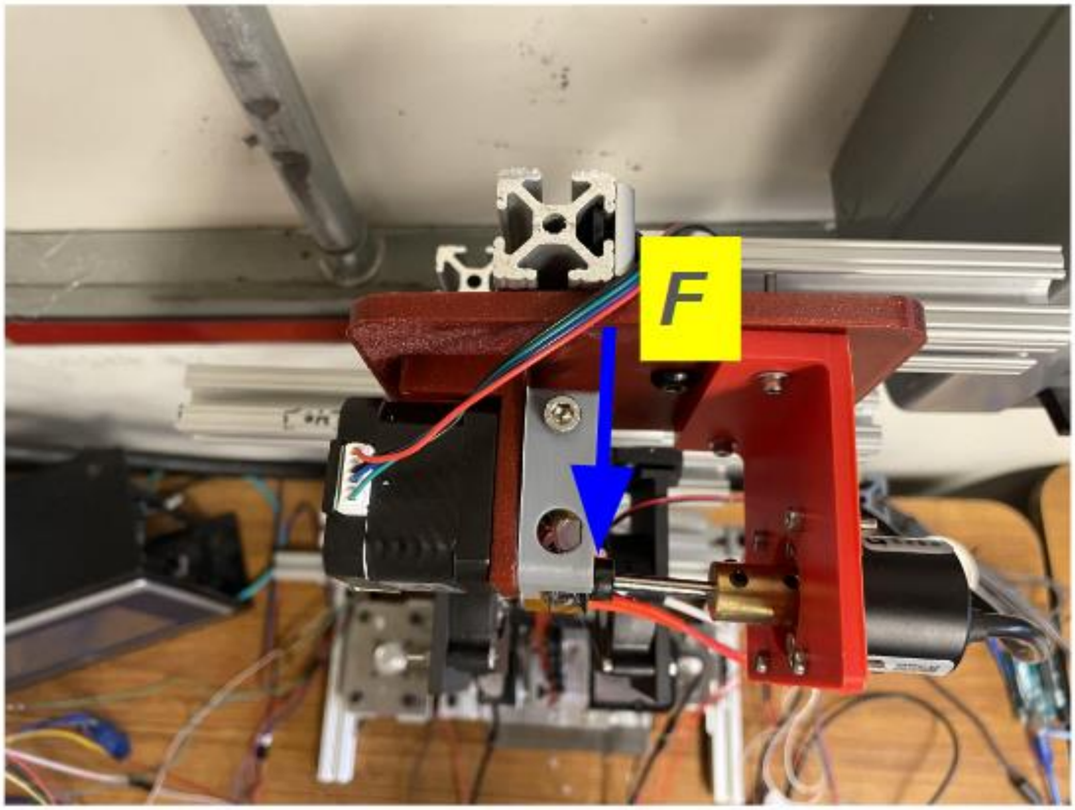


Figure 3.12. Preform/Heater Assembly Free Body Diagram. (Actual)

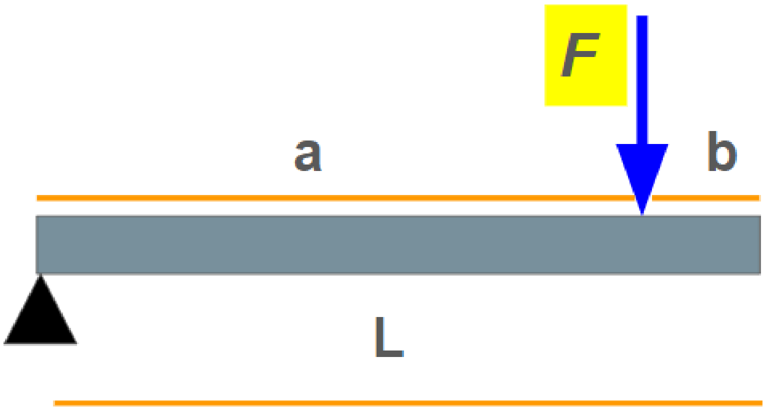


Figure 3.13. Preform/Heater Assembly Free Body Diagram. (Block Diagram)

The maximum deflection can be modeled as a cantilever beam with a downward force *F* and mathematically expressed through eq(3.17) derived from classical Euler-Beam theory.

$$\delta_{\mathrm{OL\_max}} = \frac{Fa^3}{3EI} \tag{3.17}$$

This force *F* comes from the displacement condition that is the diameter D_1, which is the preform diameter. Since this diameter D_1 is 6.9 mm while the nominal distance between the main gear and the idler gear is 6.5 mm (see Figure 3.14)

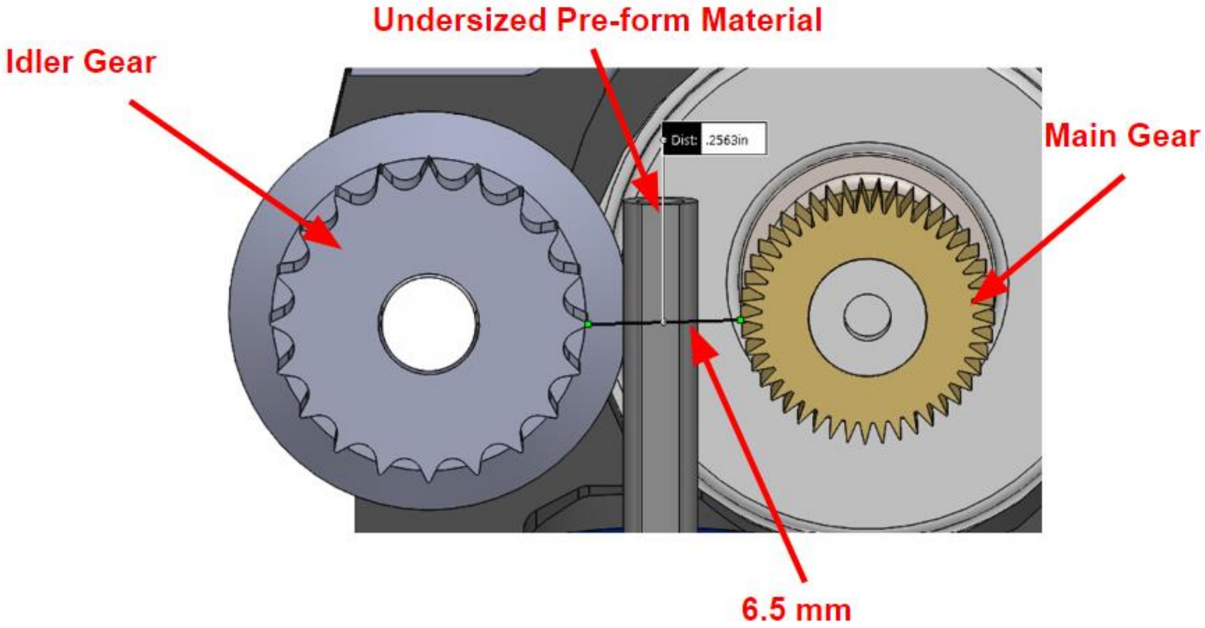


Figure 3.14. Preform/Heater Assembly Main/Idler Gear.

As a result, the shaft cantilevers/deflects due to the displacement condition coming from the preform material. If there is too much deflection and given the stiffness of the aluminum shaft is low, there will be almost no reaction force coming from the idler gear onto the preform material that is needed to push the pre-form material into the heating chamber. To solve this issue either the length (*a*) at which the force *F* acts on the shaft could be reduced, the Young's Modulus (*E*) of the shaft could be increased (changing to a stiffer material, which would also lead to higher stresses) or the second moment of inertia (*I*) could be increased (meaning getting a thicker shaft or using a tube). While these options would work to minimize the maximum deflection in theory, none of these options seemed to be a fast, cheap option that wouldn't affect the design of adjacent component.

To simplify this problem (using Occam's Razor) it is important to visualize the flow of forces and use the methodology of structural loops as defined by Slocum and as shown in Figure 3.15 (13). After visualizing the structural loop, the problem was solved much more simply by adding a pin joint at the free end. This could be done physically by adding in a bushing at the free end of the shaft and using a longer shaft that goes through that bushing. This is shown in the new free body diagram in Figure 3.16.

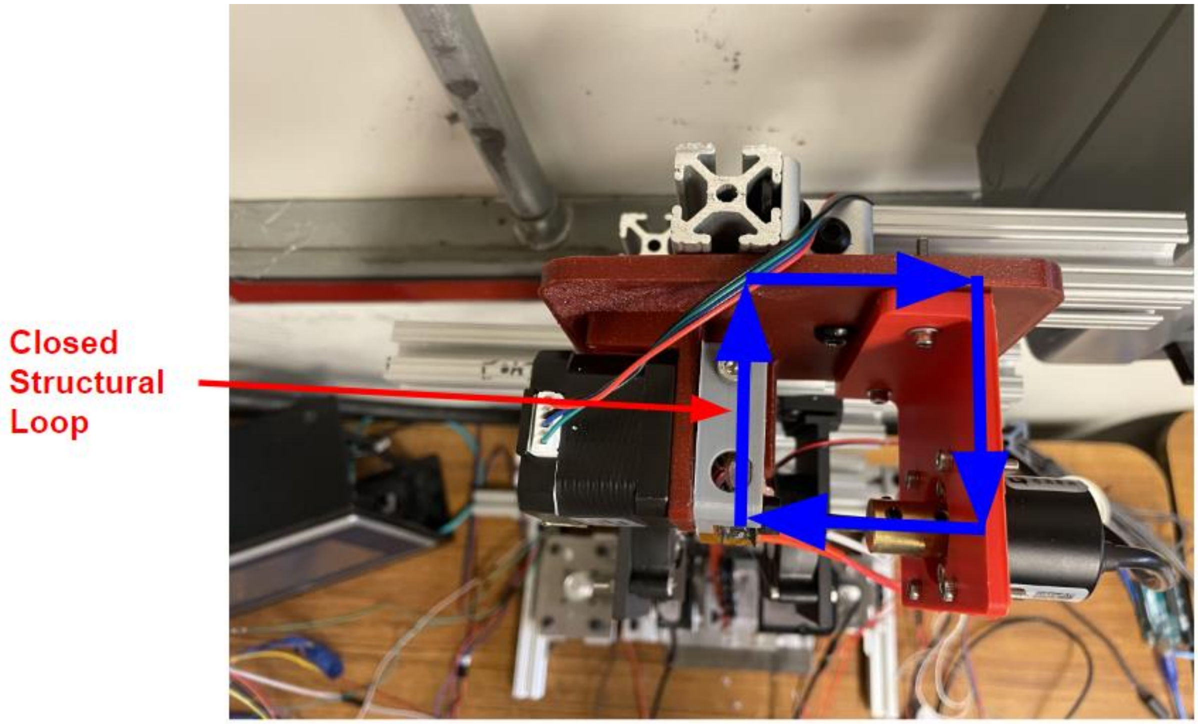


Figure 3.15. Preform/Heater Assembly Free Body Diagram. (Actual)

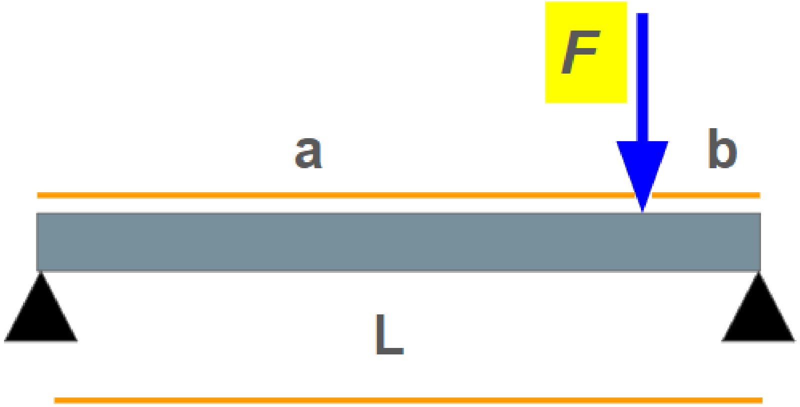


Figure 3.16. Preform/Heater Assembly Free Body Diagram. (Block Diagram)

The maximum deflection of this closed structural loop is shown mathematically through the new equation,

$$\delta_{\mathrm{CL\_max}} = \frac{Fb(3L^2-4b^2)}{48EI} \tag{3.18}$$

If *b* is small enough (cantilever pin joint placed as close to when the force *F* is reacted), then at the limit, $\delta_{\mathrm{CL\_max}} \cong 0$ as shown in eq (3.19).

$$\lim_{b \to 0} \frac{Fb(3L^2-4b^2)}{48EI} \cong 0 \tag{3.19}$$

This elimination of slippage can be verified by re-calculating the SF, which at 0.6 rpm rotational speed of the pre-form stepper motor is ~ 0.

3.4.3. Traverse Section Mechanism Design

To resolve the issue of backlash (as discussed in detail in section 3.3), it was decided to re-design the traverse mechanism in a way to remove all backlash. This was done by replacing the lead screw/nut mechanism to using either a pre-loaded screw/nut system or a timing belt traversing system. As the new traversing system was designed a short-term solution to remove backlash was implemented to utilize a pulley/mass system to always maintain a constant force on the spool platform. This helped to keep the lead screw and the nut in contact all times so there was no backlash when the platform/nut changed direction. This setup is shown in Figure 3.18. Figure 3.17 shows a constant force spring but a constant load using a mass/pulley system has the equivalent effect as a constant force spring in this context.

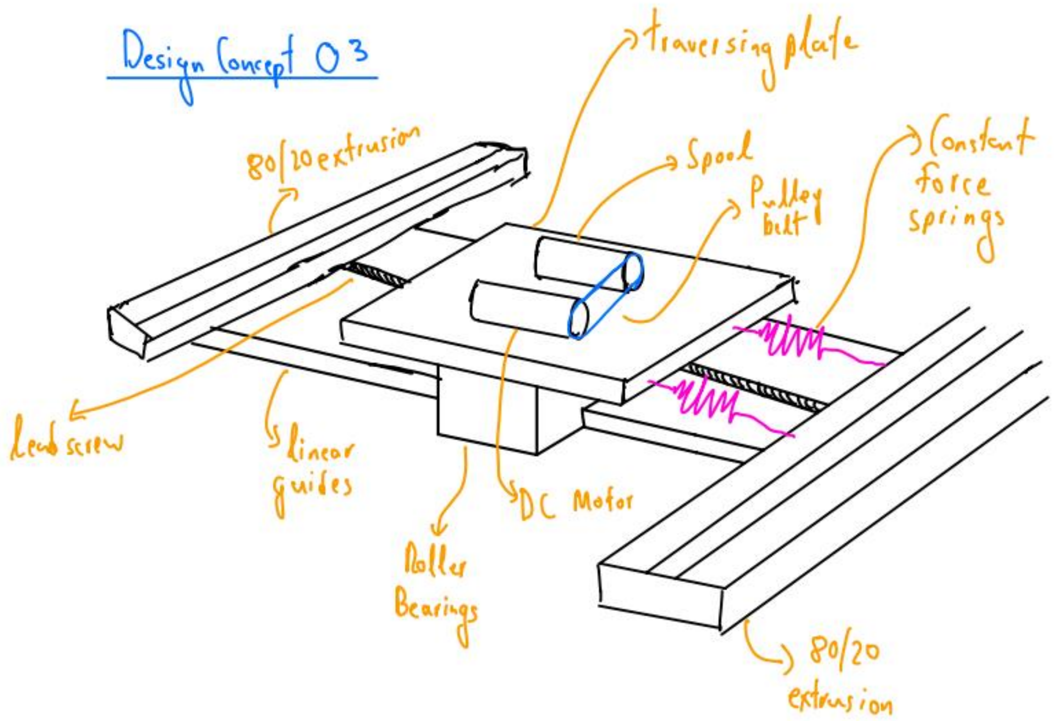


Figure 3.17. Conceptual Sketch for Eliminating Backlash in Spool Platform

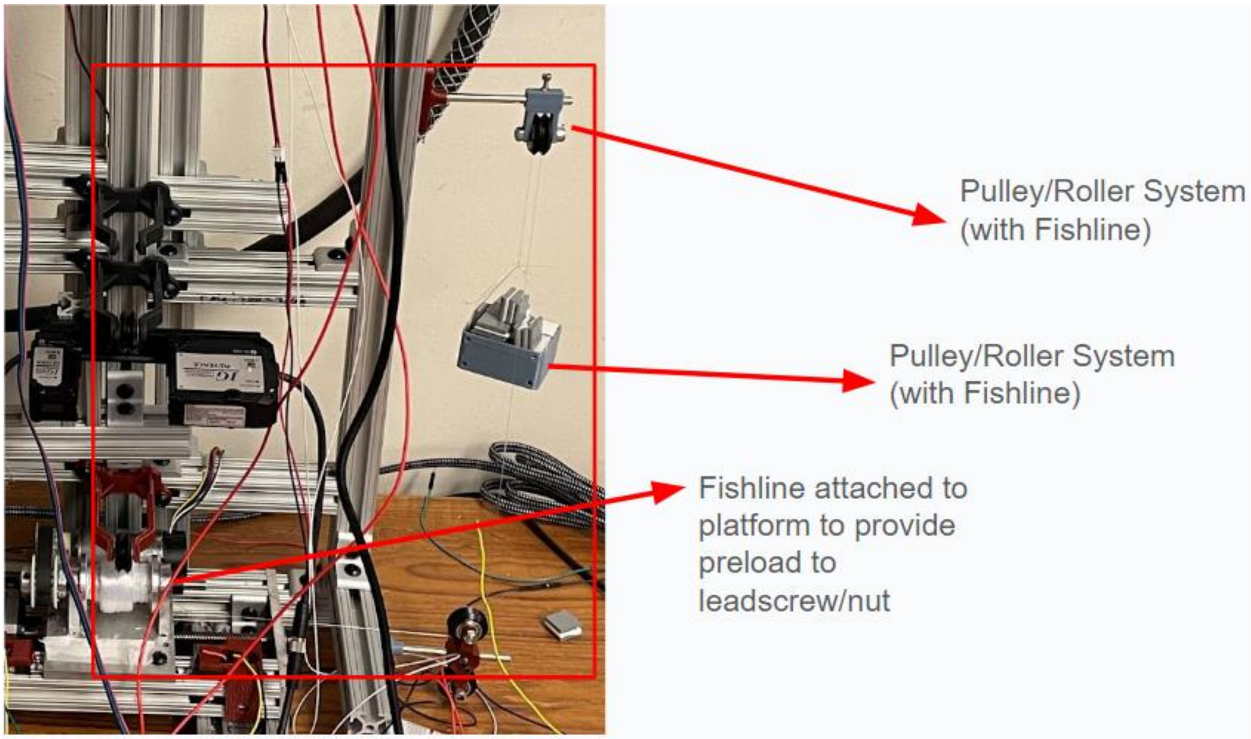
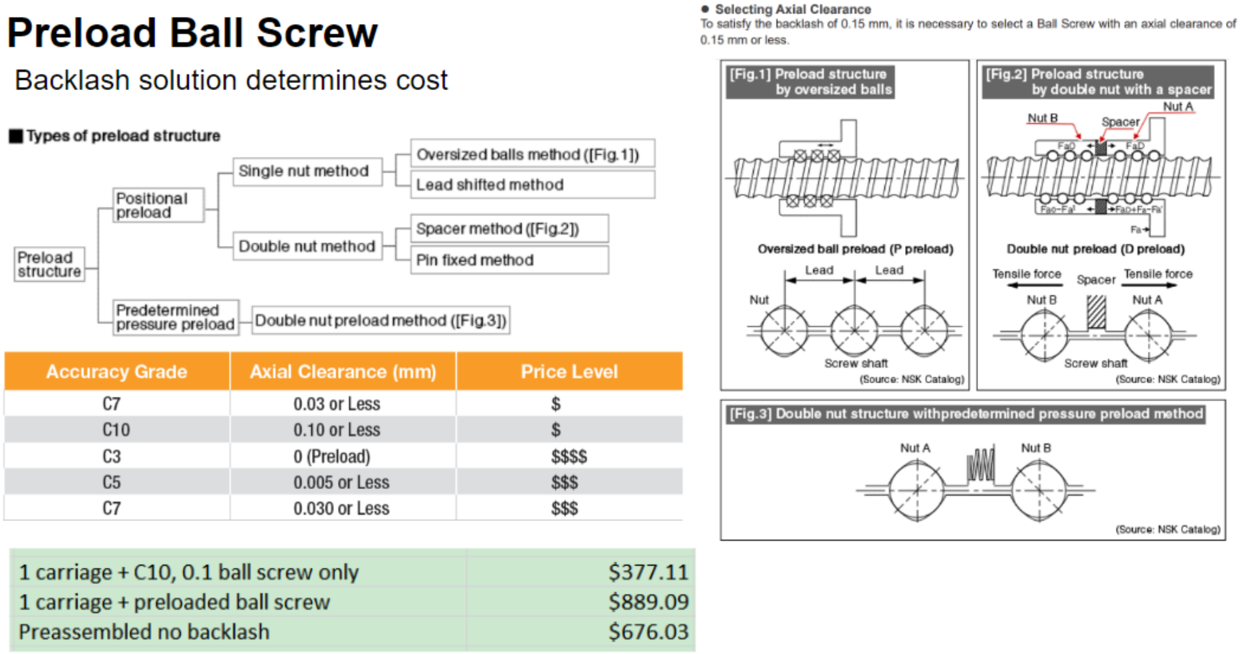

Pulley/Roller System (with Fishline)
Pulley/Roller System (with Fishline)
Fishline attached to platform to provide preload to leadscrew/nut

Figure 3.18. Manufactured Pulley/Mass System for Eliminating Backlash in Spool Platform

3.4.3.1 Preloaded Lead Screw/Nut Design

Preloading on the ball screw is done either with oversized balls or a preload device like a spring or a spacer. The latter is much more expensive. However, the oversized ball solution does not eliminate backlash. The cost estimates of both systems are shown in Figure 22. The idea of a preloaded ball screw is dropped as the timing belt system can remove backlash entirely with the right amount of belt tension, and it costs much less.



Preload Ball Screw

Backlash solution determines cost

■ Types of preload structure

Accuracy Grade	Axial Clearance (mm)	Price Level
C7	0.03 or Less	$
C10	0.10 or Less	$
C3	0 (Preload)	$$$$
C5	0.005 or Less	$$$
C7	0.030 or Less	$$$

1 carriage + C10, 0.1 ball screw only	$377.11
1 carriage + preloaded ball screw	$889.09
Preassembled no backlash	$676.03

● Selecting Axial Clearance

To satisfy the backlash of 0.15 mm, it is necessary to select a Ball Screw with an axial clearance of 0.15 mm or less.

[Fig.1] Preload structure by oversized balls
Oversized ball preload (P preload)
Lead
Lead
Nut
Screw shaft
(Source: NSK Catalog)
[Fig.2] Preload structure by double nut with a spacer
Nut B
Spacer
Nut A
Fa0
FaD
Fa0−Fa'
Fa0+Fa−Fa'
Fa
Double nut preload (D preload)
Tensile force
Spacer
Tensile force
Nut B
Nut A
Screw shaft
(Source: NSK Catalog)
[Fig.3] Double nut structure withpredetermined pressure preload method
Nut A
Nut B
(Source: NSK Catalog)

Figure 3.19. Preload Ball Screw System Cost

3.4.3.2 Timing Belt Design

The timing belt system involves a timing belt that is attached to a linear roller bearing platform rolling on a linear guide/rail. This can be shown in Figure 3.20 and Figure 3.21.

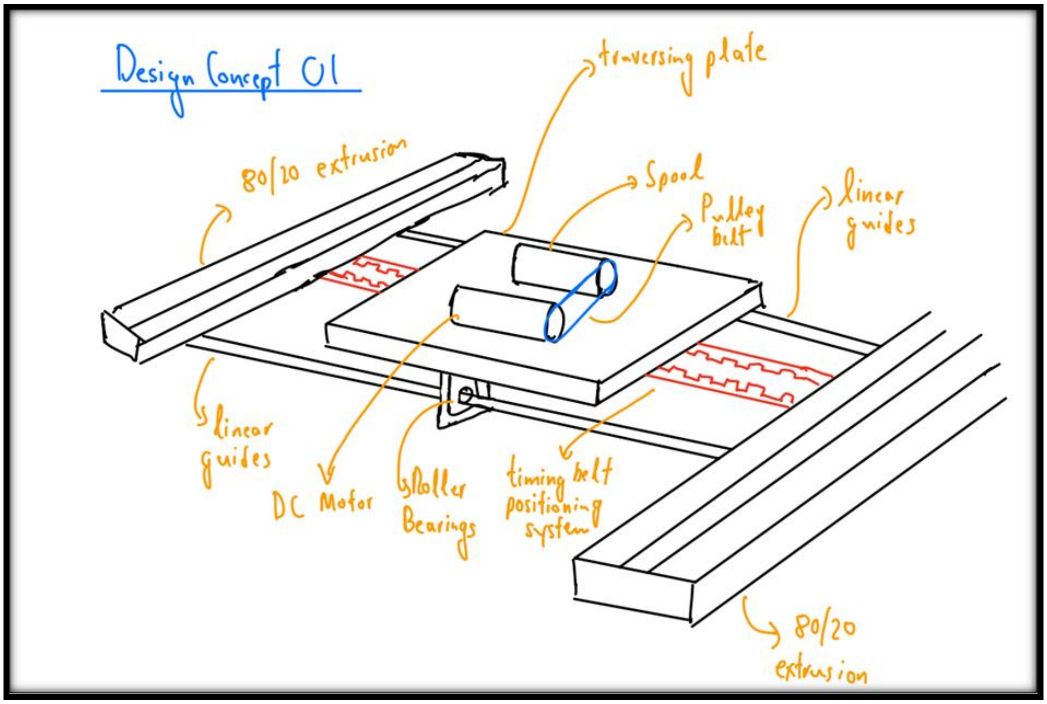


Figure 3.20. Timing Belt Linear Guide System (Two Linear Guide Block)

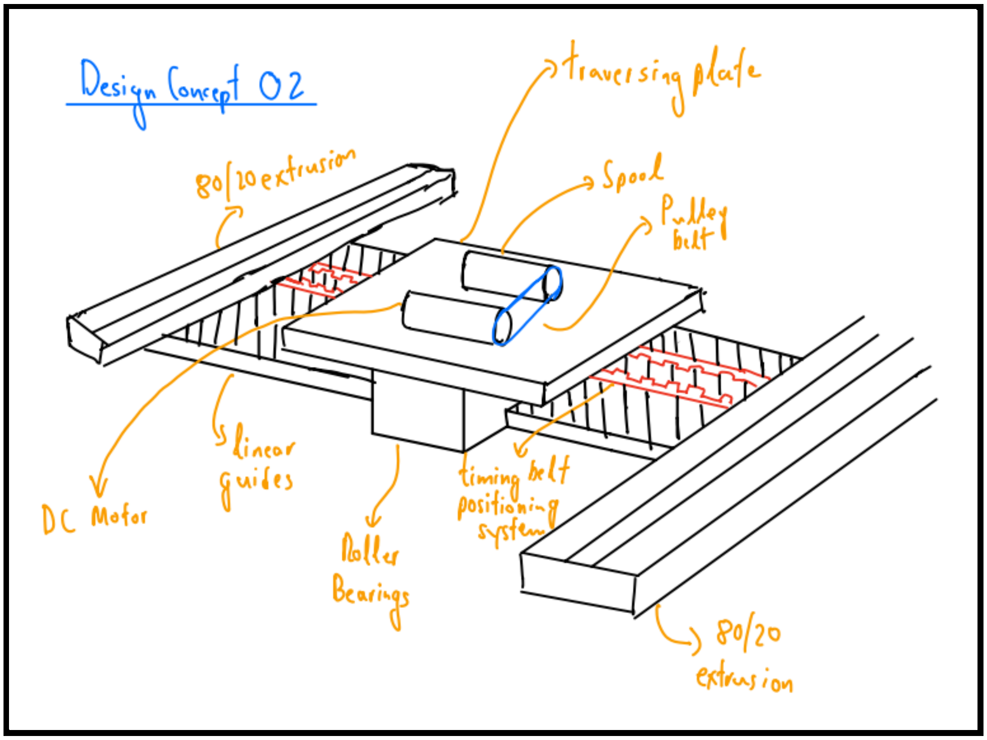


Figure 3.21. Timing Belt Linear Guide System (One Linear Guide Block)

The primary difference between Figure 3.20 and Figure 3.21 is using one linear guide block versus using two linear guide blocks. Since the center of mass resting on the platform will go through the center of mass of the linear guide block, there should be almost no difference in bearing life resulting from the roll moment. Therefore, using the principle of Occam's Razor, one linear guide block design was chosen to make the analysis of the roll and pitch moments easier, make the assembly process easier, keep costs low, and minimize any chance of the bearings binding due to non-parallel guides.

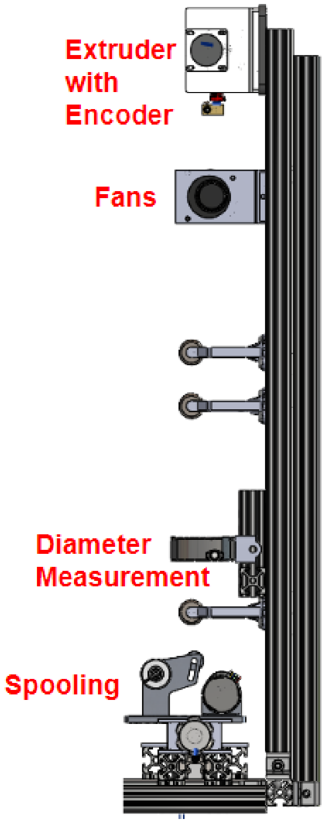


Figure 3.22. Side View of Air Cooling FrED (with timing belt traverse system)

The CAD of the timing belt traverse system and the various features are shown in Figure 3.23.

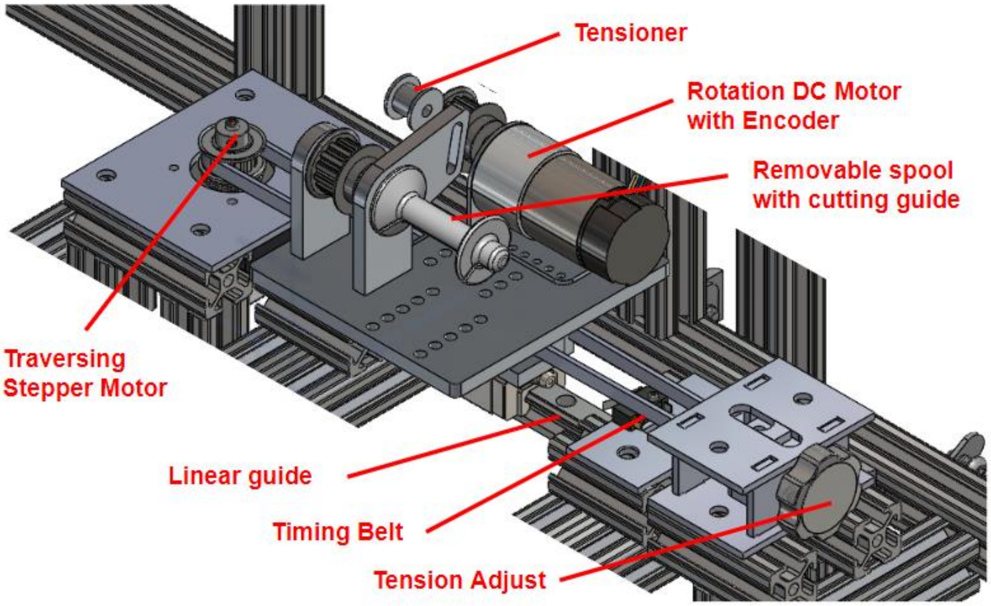


Figure 3.23. Isometric View of Air Cooling FrED (with timing belt traverse system)

manufactured hardware of the timing belt traverse system and the various features are shown in Figures 3.24, 3.25, and 3.26.

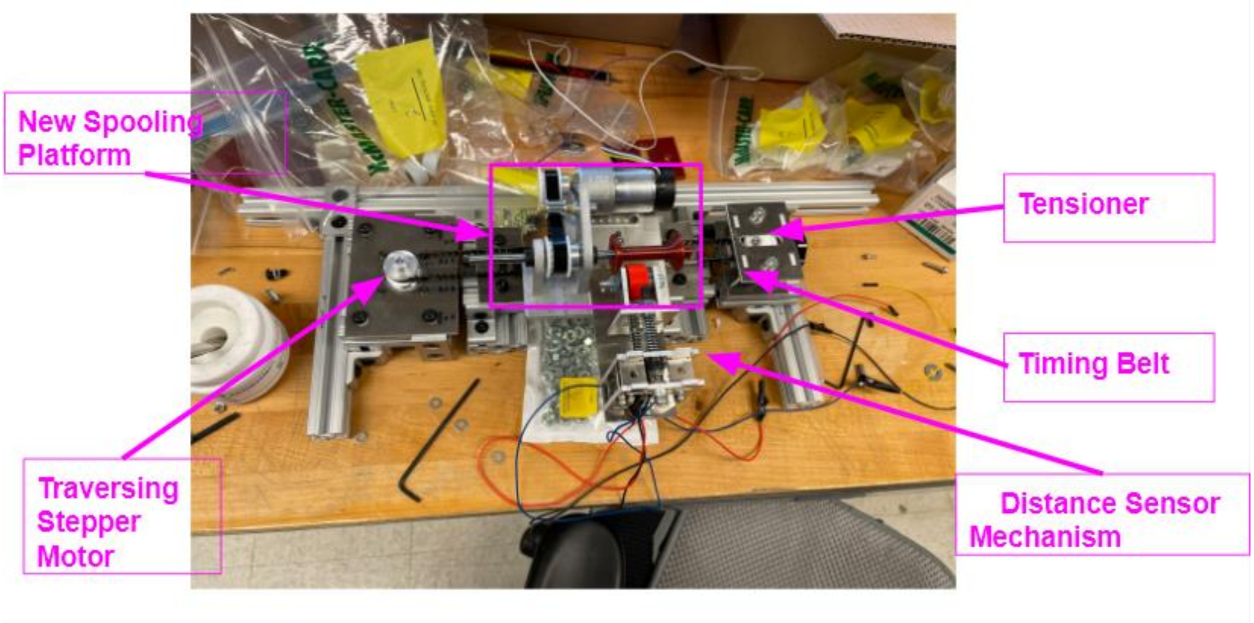


Figure 3.24. Top View of Isolated Timing Belt Traverse Mechanism (Including Distance Sensor Mechanism)

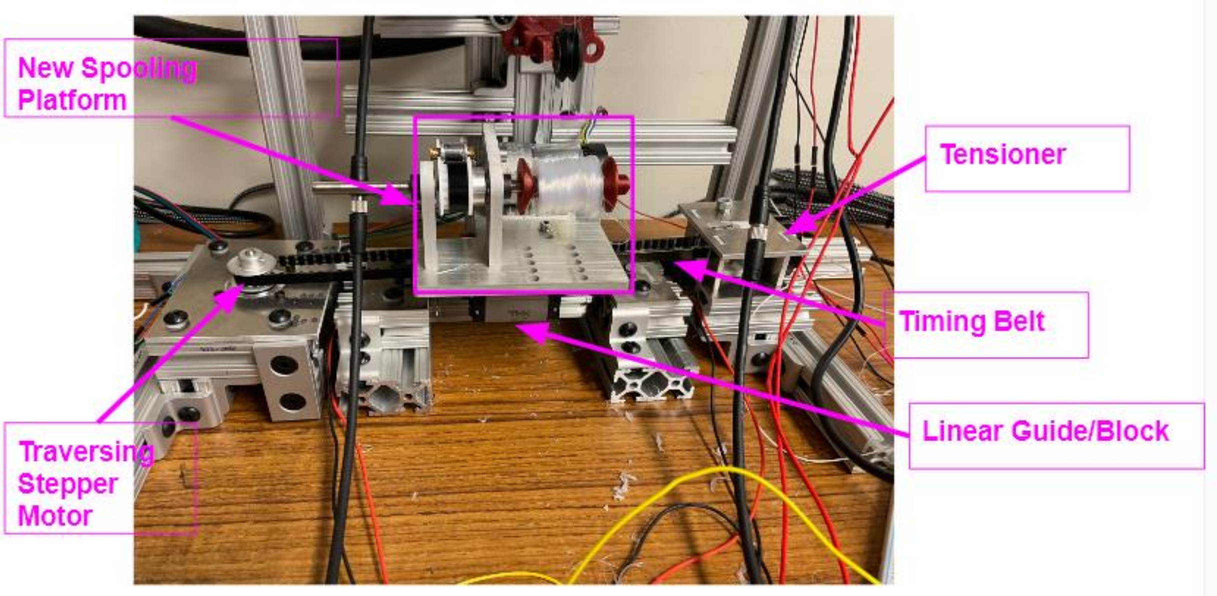


Figure 3.25. Front View of Integrated Timing Belt Traverse System (with Air Cooled FrED)

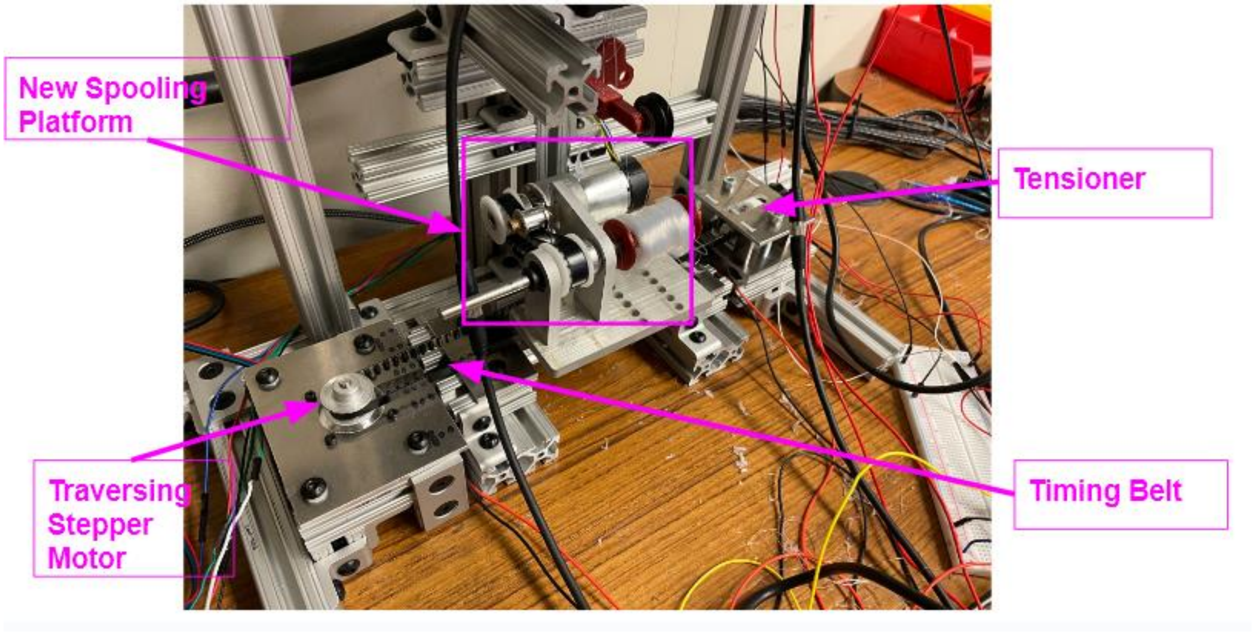


Figure 3.26. Isometric View of Integrated Timing Belt Traverse System (with Air Cooled FrED)

The purpose of the distance sensor is to get the real time diameter of the spool that can be used as a parameter to train an adaptive learned controller that can control the diameter variation much better than the conventional PID controllers. Even though the principle of symmetry would be violated in the introduction of the spool growth sensor mechanism (the design and manufacturing of which is yet to be implemented), sufficient analysis (as detailed in Figure 3.31 and table 3.5) has been done to account for the additional roll moment carried by the roller bearing of the linear guide due to the center of mass (CM) shifting.

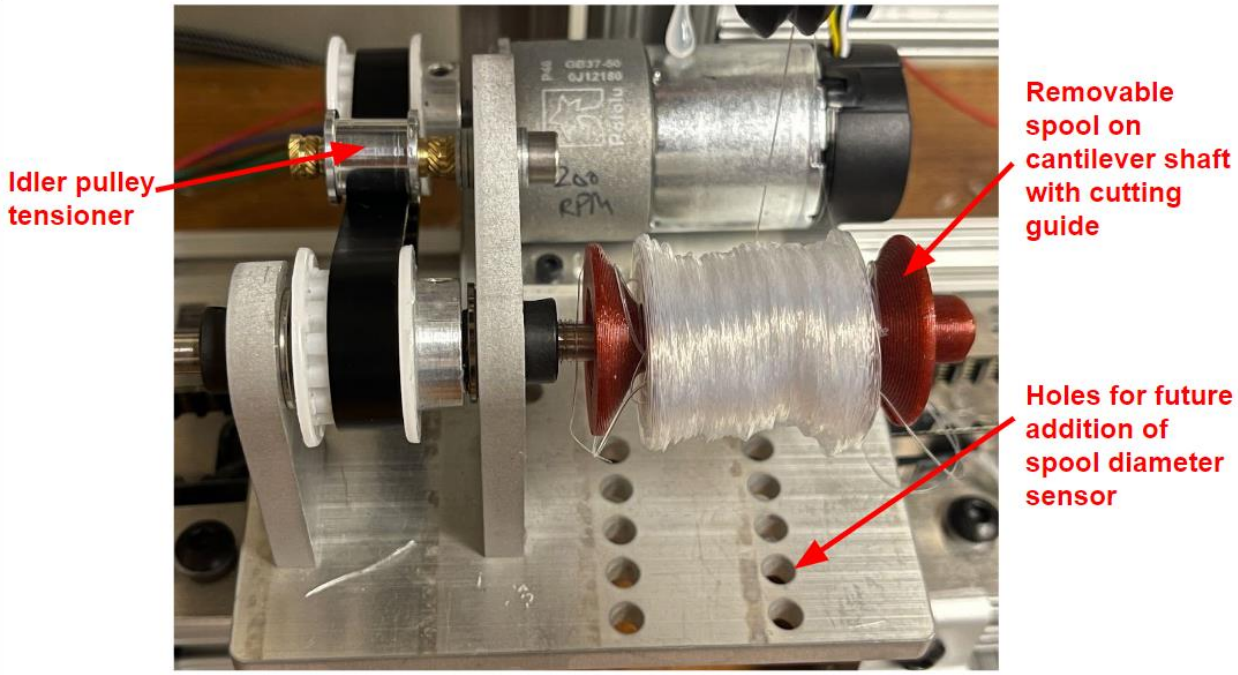

Idler pulley tensioner
Removable spool on cantilever shaft with cutting guide
Holes for future addition of spool diameter sensor
200 RPM

Figure 3.27. New Spooling Platform

To increase the spooling capacity, the spool is moved further away from both the DC motor and the base of the platform, increasing capacity by 144%. Easier and faster spooling initialization is made possible by making the spool attached to a cantilevered shaft so that one end is open, and this enables the user’s hand to warp the start of the fiber to the spool with ease, hence reducing the time taken to initialize spooling by 91%.

There are two timing belts. First is the timing belt of the DC spool motor. The second is the timing belt associated with the stepper motor of the traverse system. There are changes in how the timing belt of the spool motor is tensioned. Previously, the user would have to loosen the screws attaching the spool stands to the platform and pull it along the slot to tension the belt, which requires flipping the platform over. Now, using an idler pulley that can be locked and moved perpendicular to the timing belt, belt tension can be easily adjusted. In addition, fiber removal from the spool is also made easier by having a groove along the cylindrical spool body that the pen knife can follow to cut through the fiber, without worrying that the blade will slip and cut the user, time taken to remove fibers reduced by around 50%. A new tensioner mechanism for the traverse mechanism was also designed and manufactured.

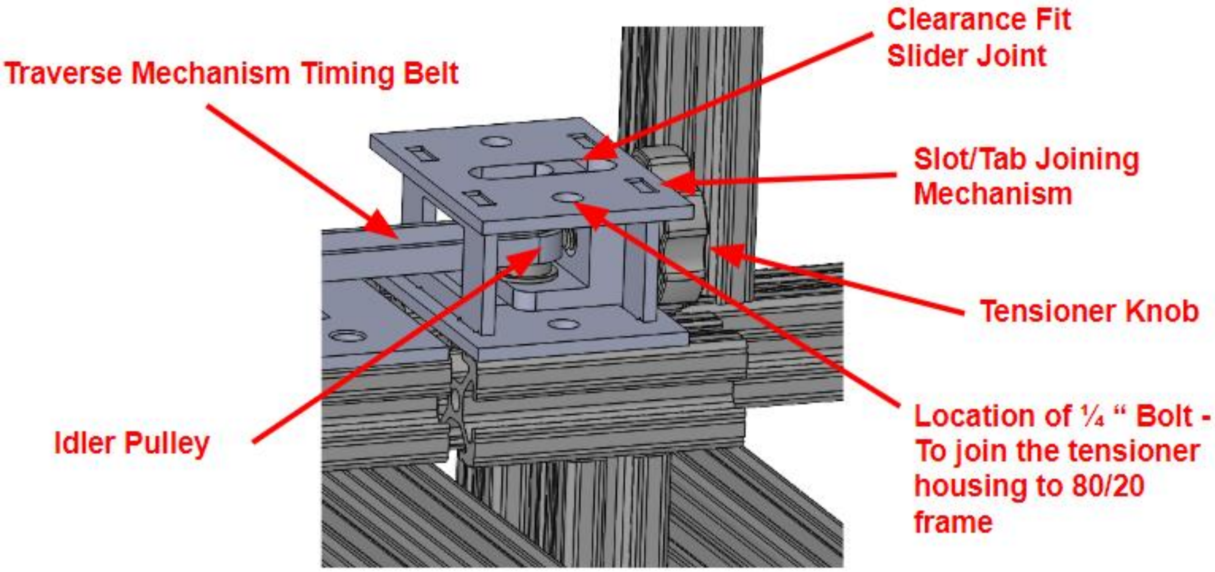
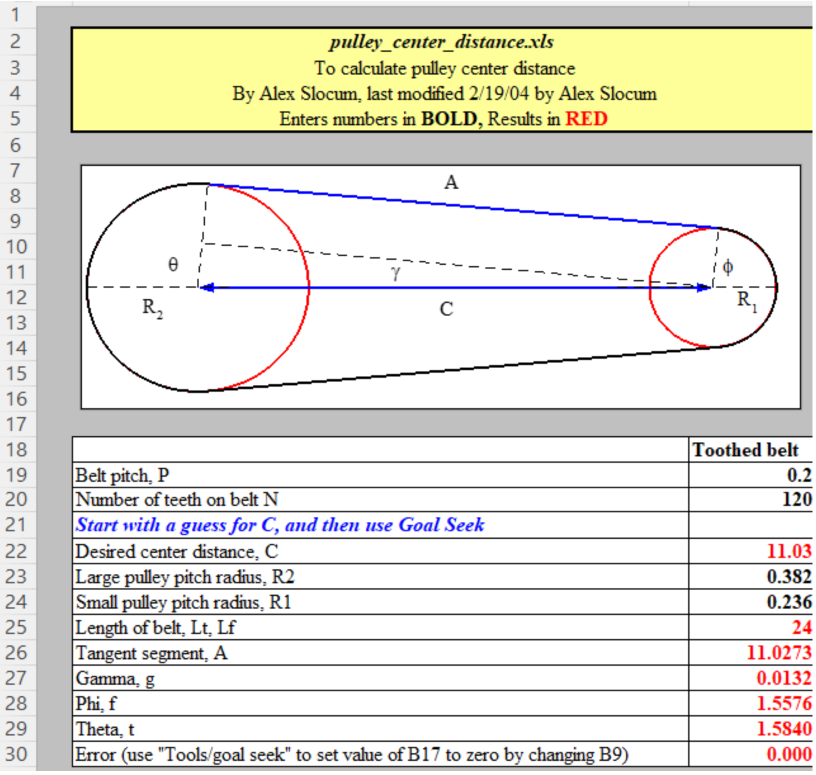

Traverse Mechanism Timing Belt
Clearance Fit Slider Joint
Slot/Tab Joining Mechanism
Tensioner Knob
Idler Pulley
Location of ¼ " Bolt - To join the tensioner housing to 80/20 frame

Figure 3.28. Traverse Mechanism Timing Belt Tensioner

The length of the traverse mechanism timing belt was determined by using “pulley_center_distance” spreadsheet published by Slocum (13). The spreadsheet uses desired center distances, belt pitch, number of tooth on the belt, radius of the roller and the idler pulley as inputs (highlighted in black). The sheet then outputs the length of the belt (highlighted in red) which can be bought from a vendor. The main logic behind the sheet is the iterative solving non-linear equations where,

$$Length\ of\ Belt = f(pulley\ diameters, center\ distance, \theta, \phi) \tag{3.20}$$



pulley_center_distance.xls
To calculate pulley center distance
By Alex Slocum, last modified 2/19/04 by Alex Slocum
Enters numbers in **BOLD,** Results in **RED**

	Toothed belt
Belt pitch, P	**0.2**
Number of teeth on belt N	**120**
Start with a guess for C, and then use Goal Seek	
Desired center distance, C	**11.03**
Large pulley pitch radius, R2	**0.382**
Small pulley pitch radius, R1	**0.236**
Length of belt, Lt, Lf	**24**
Tangent segment, A	**11.0273**
Gamma, g	**0.0132**
Phi, f	**1.5576**
Theta, t	**1.5840**
Error (use "Tools/goal seek" to set value of B17 to zero by changing B9)	**0.000**

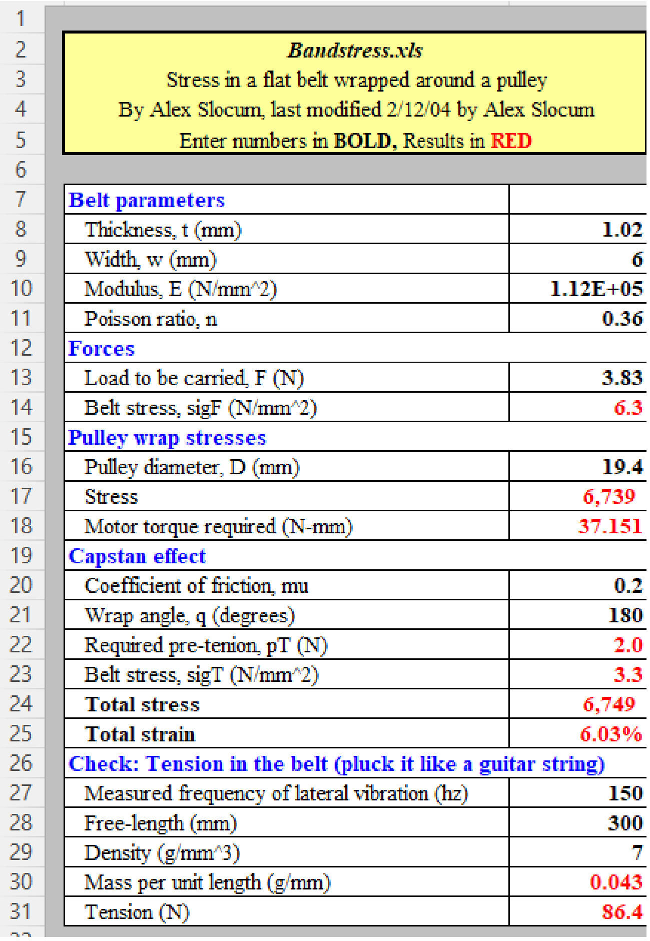
Figure 3.29. Traverse Mechanism Timing Belt Center Distance Calculations

“Bandstress” is another spreadsheet where belt geometric parameters (finalized from “pulley_center_distance”) are input. The output are the three different types of normal stresses the belt experiences. They are given by eq(3.21) – eq(3.23)

$$\sigma_{load} = \frac{F}{wt} \tag{3.21}$$

$$\sigma_{pre-tension} = \frac{T}{wt} \tag{3.22}$$

$$\sigma_{pulley} = \frac{tE}{D(1-\nu^2)} \tag{3.23}$$



Bandstress.xls
Stress in a flat belt wrapped around a pulley
By Alex Slocum, last modified 2/12/04 by Alex Slocum
Enter numbers in **BOLD**, Results in RED

Belt parameters	
Thickness, t (mm)	**1.02**
Width, w (mm)	**6**
Modulus, E (N/mm^2)	**1.12E+05**
Poisson ratio, n	**0.36**
Forces	
Load to be carried, F (N)	**3.83**
Belt stress, sigF (N/mm^2)	6.3
Pulley wrap stresses	
Pulley diameter, D (mm)	**19.4**
Stress	6,739
Motor torque required (N-mm)	37.151
Capstan effect	
Coefficient of friction, mu	**0.2**
Wrap angle, q (degrees)	**180**
Required pre-tenion, pT (N)	2.0
Belt stress, sigT (N/mm^2)	3.3
Total stress	6,749
Total strain	6.03%
Check: Tension in the belt (pluck it like a guitar string)	
Measured frequency of lateral vibration (hz)	**150**
Free-length (mm)	**300**
Density (g/mm^3)	**7**
Mass per unit length (g/mm)	0.043
Tension (N)	86.4

Figure 3.30. Traverse Mechanism Timing Belt Tension Calculations

Using these two tools, a length of ~24 inches long made of Kevlar reinforce urethane was purchased through McMaster Carr (see Appendix E for more information).

3.4.3.3 Carraige Ball Bearing Life Analysis

To validate if the linear guide system is capable enough to withstand the load of the traversing platform and the spooling growth sensor mechanism (yet to be implemented), first-order calculations were done using force and moment balance as depicted in Figure 3.31.

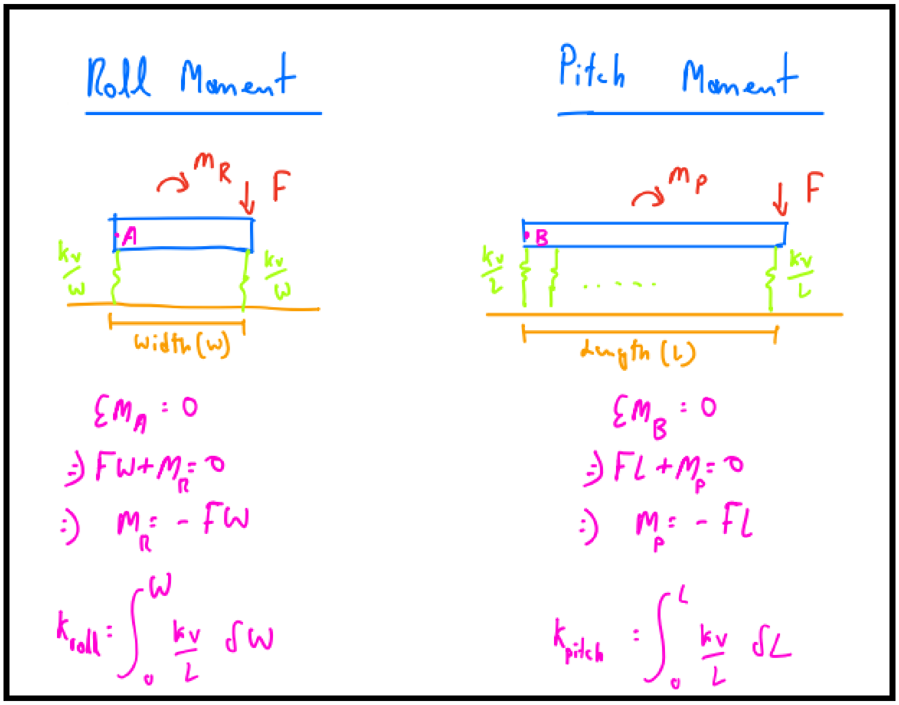
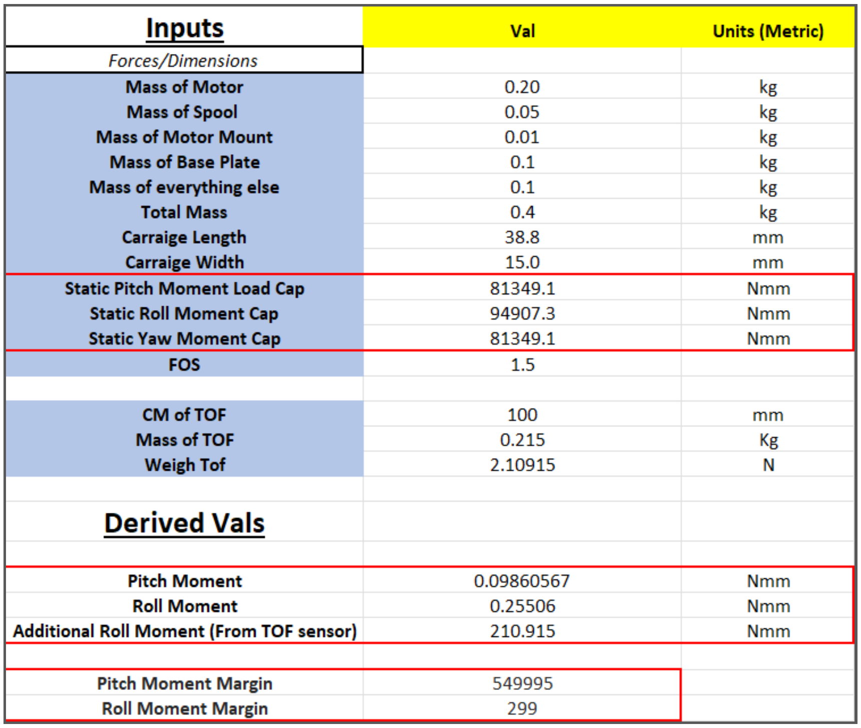


Figure 3.31. Traverse Spool Platform Carraige Roll/Pitch Moment Derivations

Table 3.5. Linear Guide Pitch and Roll Moment Margins.



Inputs	Val	Units (Metric)
Forces/Dimensions		
Mass of Motor	0.20	kg
Mass of Spool	0.05	kg
Mass of Motor Mount	0.01	kg
Mass of Base Plate	0.1	kg
Mass of everything else	0.1	kg
Total Mass	0.4	kg
Carraige Length	38.8	mm
Carraige Width	15.0	mm
Static Pitch Moment Load Cap	81349.1	Nmm
Static Roll Moment Cap	94907.3	Nmm
Static Yaw Moment Cap	81349.1	Nmm
FOS	1.5	
CM of TOF	100	mm
Mass of TOF	0.215	Kg
Weigh Tof	2.10915	N
Derived Vals		
Pitch Moment	0.09860567	Nmm
Roll Moment	0.25506	Nmm
Additional Roll Moment (From TOF sensor)	210.915	Nmm
Pitch Moment Margin	549995	
Roll Moment Margin	299	

In Table 10, the margin has been calculated using the following equation,

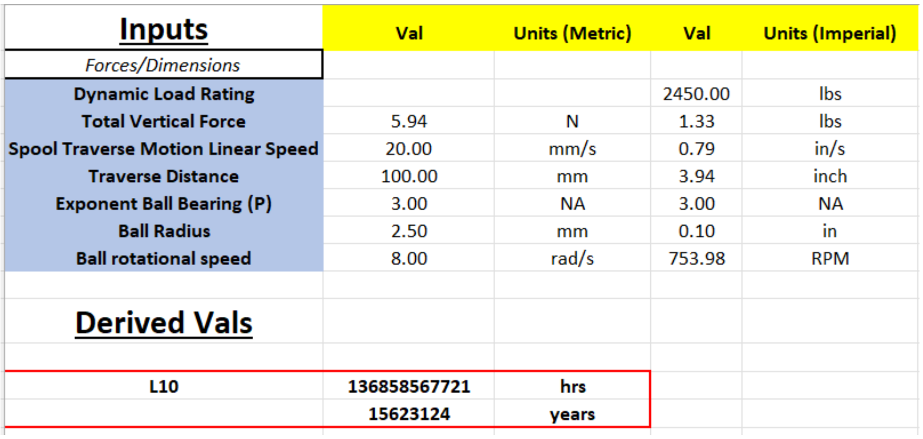
$$Margin\,of\,Safety\,(MOS) = \frac{Capability}{(Worst\,Case\,Loads)(Factor\,of\,Safety)} - 1 \qquad (3.24)$$

It can be seen that MOS is well above 1, and bearing failure due to static loads is not likely. Similarly L10 life of the bearing has been calculated with the expression in eq(3.25)

$$L_{10} = \left(\frac{C}{F}\right)^{P}\left(\frac{10^{6}}{60N}\right) \qquad (3.25)$$

where, C - Dynamic load rating, F - applied load/weight, P - 3 (ball bearings) or 10/3 (roller Bearings), N - RPM

Table 3.6. L10 Linear Guide Roller Bearing Life



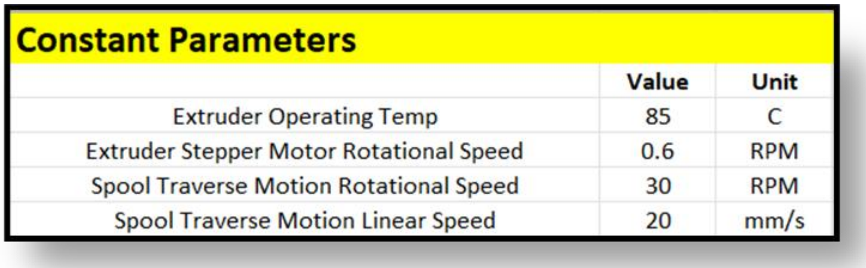
Inputs	**Val**	**Units (Metric)**	**Val**	**Units (Imperial)**
Forces/Dimensions				
Dynamic Load Rating			2450.00	lbs
Total Vertical Force	5.94	N	1.33	lbs
Spool Traverse Motion Linear Speed	20.00	mm/s	0.79	in/s
Traverse Distance	100.00	mm	3.94	inch
Exponent Ball Bearing (P)	3.00	NA	3.00	NA
Ball Radius	2.50	mm	0.10	in
Ball rotational speed	8.00	rad/s	753.98	RPM
Derived Vals				
L10	**136858567721**	**hrs**		
	15623124	**years**		

Therefore, it can be concluded that the ball bearings in the linear guide will have infinite life.

3.5 Hardware Testing

This section explores the performance of the PLC FrED, the design and manufacturing of which was discussed in section 3.1-3.4. The constant operating parameters of the air cooling FrED are summarized in Table 3.7.

Table 3.7. Constant Operating Parameter of Air-Cooling System



Constant Parameters		
	Value	Unit
Extruder Operating Temp	85	C
Extruder Stepper Motor Rotational Speed	0.6	RPM
Spool Traverse Motion Rotational Speed	30	RPM
Spool Traverse Motion Linear Speed	20	mm/s

This run chart of the fiber diameter that was generated is shown in Figure 3.32. At the test run which is shown in Figure 10 with a set point of 0.2mm, the variation range is about the same at 50% of the set point.

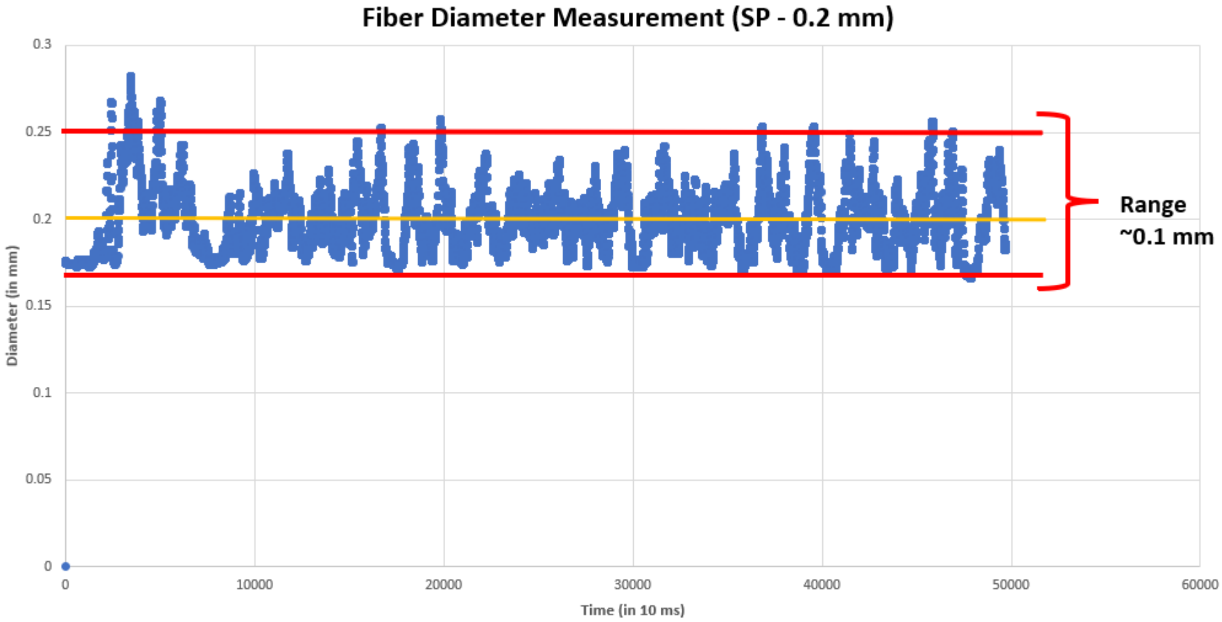


Figure 3.32. PID Controlled Diameter Run Chart of Air-Cooled System (Backlash and Temp: 85C)

To get an estimate regarding which frequency component of the diameter plots in Figure 10 are dominating, the Fast Fourier Transform (FFT) of this data was done. The Fast Fourier Transform (FFT) is an algorithm that efficiently computes the Discrete Fourier Transform (DFT) of a sequence, reducing the computational complexity, where N is the number of samples in the sequence. This can be process can be mathematically expressed as shown in eq (3.26).

$$x[k] = \sum_{n=0}^{N-1} x[n]\, e^{\frac{-j2\pi kn}{N}} \tag{3.26}$$

The FFT plots and associated dominating frequencies of the data shown in Figure 3.33 and 3.34.

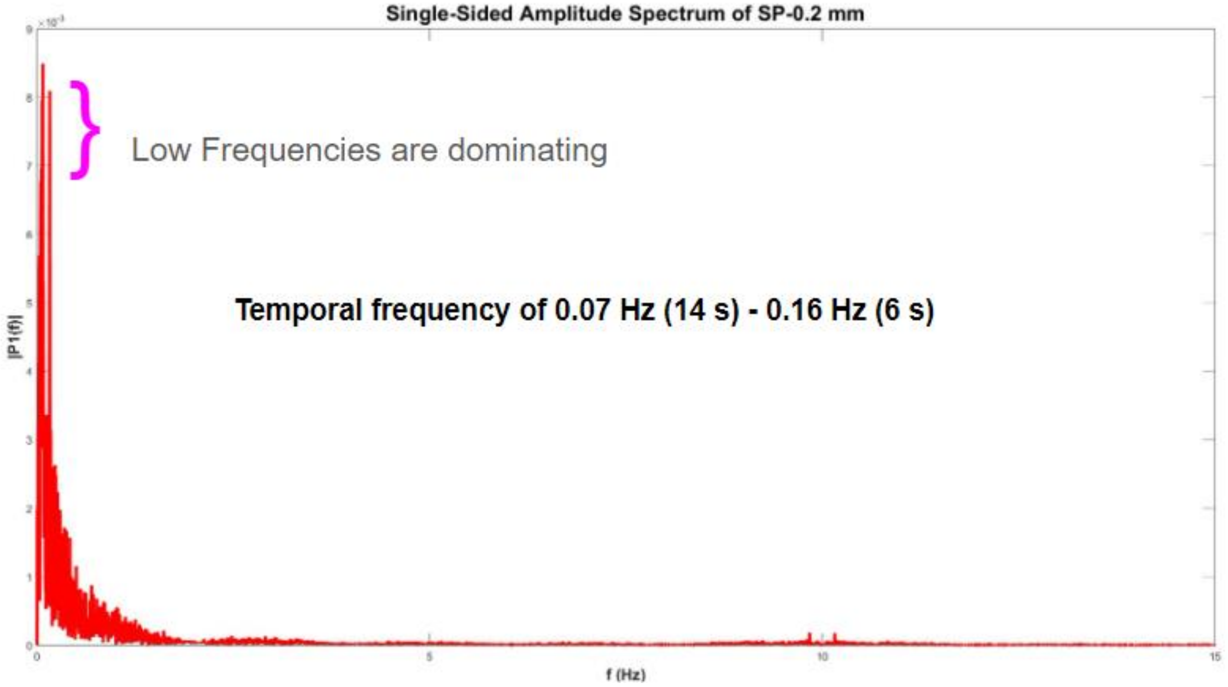


Figure 3.33. Temporal FFT of Fiber Diameter (Backlash & Temp: 85C)

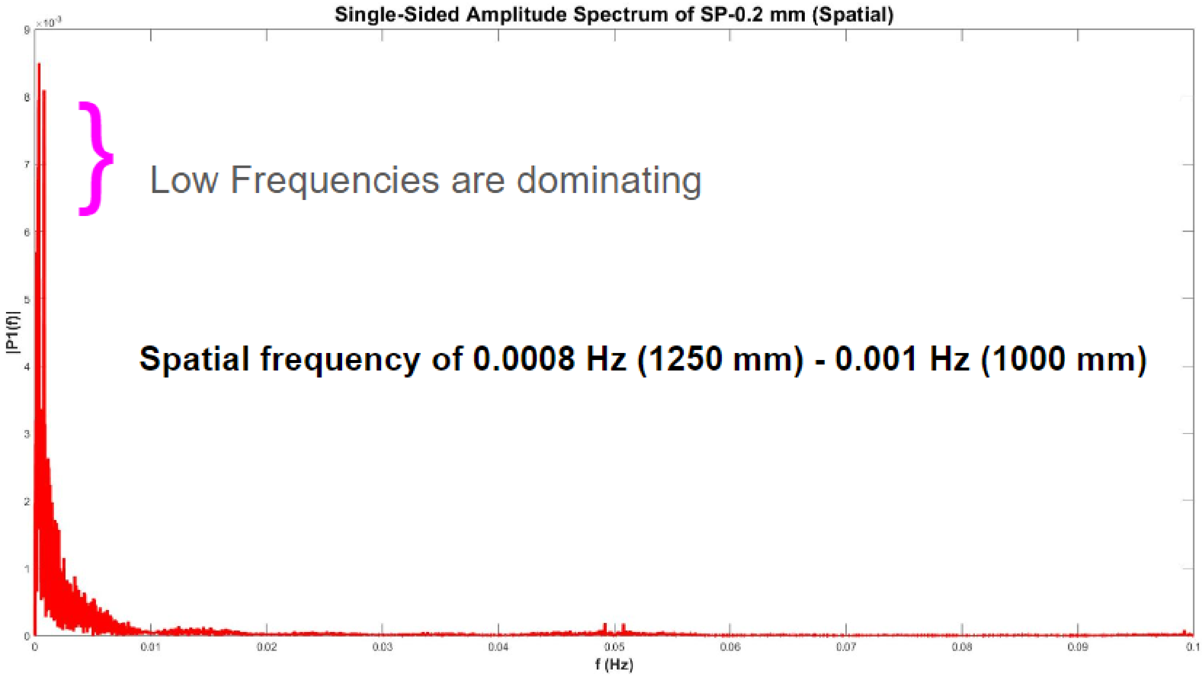
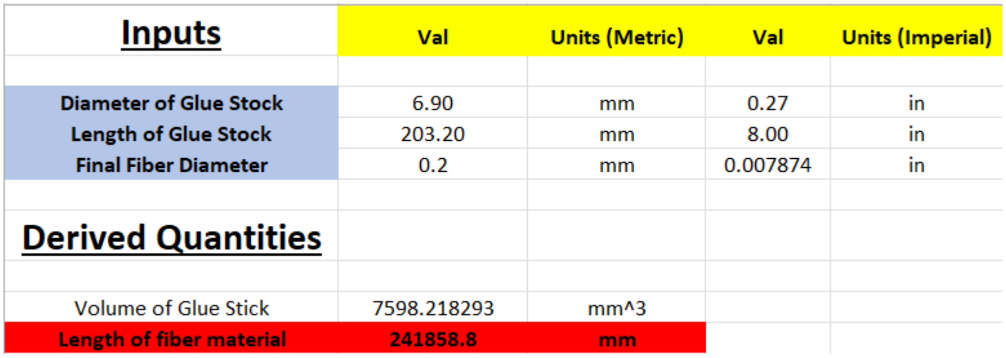


Figure 3.34. Spatial FFT of Fiber Diameter (Backlash & Temp: 85C)

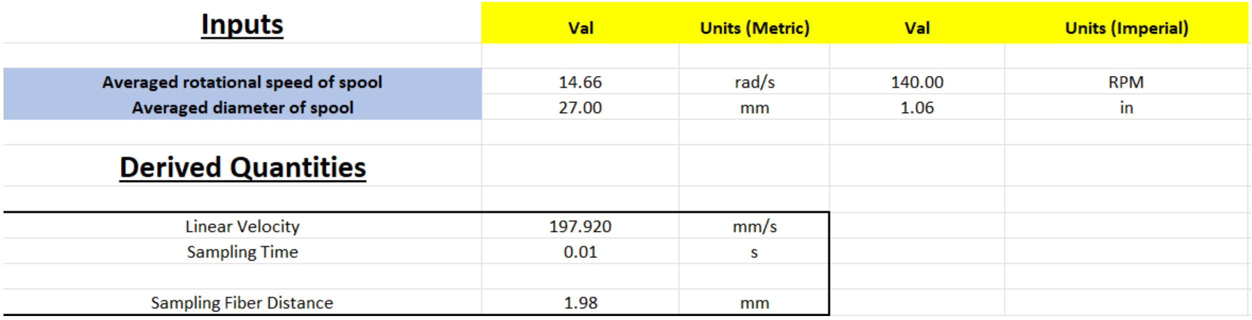
The entire preform glue stick (6.9 mm diameter and 203 mm long) can produce a length of 241858 mm fiber of 0.2 mm diameter according to the calculation (derived based on volume conservation) shown in table 3.8.

Table 3.8. Length of Fiber Generated



Inputs	**Val**	**Units (Metric)**	**Val**	**Units (Imperial)**
Diameter of Glue Stock	6.90	mm	0.27	in
Length of Glue Stock	203.20	mm	8.00	in
Final Fiber Diameter	0.2	mm	0.007874	in
Derived Quantities				
Volume of Glue Stick	7598.218293	mm^3		
Length of fiber material	**241858.8**	**mm**		

Table 3.9. Fiber Sampling Distance



Inputs	**Val**	**Units (Metric)**	**Val**	**Units (Imperial)**
Averaged rotational speed of spool	14.66	rad/s	140.00	RPM
Averaged diameter of spool	27.00	mm	1.06	in
Derived Quantities				
Linear Velocity	197.920	mm/s		
Sampling Time	0.01	s		
Sampling Fiber Distance	1.98	mm		

The spatial FFT length of (1000-1250) mm being correlated to the dominating frequency makes sense. Furthermore, looking closely at the spatial and temporal FFT plots, it tells us the same thing, that low frequency components dominate at 1250 mm or (1250mm/197mm/s) ~ 6 seconds. Therefore, temporal FFT plots are used in the remainder of this project as it says the same thing as spatial FFT but is more intuitive. Observing the temporal FFT plots (Figure 3.33), it can be hypothesized that the dominating frequencies have associated time periods of 6s – 14s. This period range can be concluded to be on the order for heating the control volume of the preform material (Table 3.8 – 6.9 mm diameter and 25 mm long) using the heat conduction equation eq (3.27) (derived from the 2nd Law of Thermodynamics).

$$\begin{aligned} &Net\ rate\ of\ Heat\ Conduction\ into\ the\ Control\ Volume \\ &\qquad = Rate\ of\ Change\ of\ Thermal\ Energy\ in\ the\ Control\ Volume \end{aligned}$$

$$\frac{d^2T}{dx^2} + \frac{d^2T}{dy^2} + \frac{d^2T}{dxz^2} = \frac{\rho c_p}{k}\frac{dT}{dt} \tag{3.27}$$

Further analysis of thermal dependency is shown in chapter 4.

This page is intentionally left blank.

Chapter 4 Exploration of Thermal Effects

4.1 Motivation for Thermal Optimization

In Chapter 3, it was hypothesized the dominant frequencies from the FFT analysis (Figure 3.33 and Figure 3.34) is dependent on temperature of the heating chamber. In this section, the heating chamber temperature is varied and the effects on the dominant frequencies are explored. Through this experimentation, a local optimum temperature of the heating chamber is obtained, which ensures lower variation in the fiber diameter at a certain desired setpoint.

4.2 Analyzing Temperature Dependencies

From the initial FFT of diameter data, it was evident that the dominating frequency components with the highest energy were on the order of (6-14) s which is roughly the time it takes for 6.9 mm diameter and 25 mm long preform material (25 mm is the length of the heater block) to get to the softening point. Experiments were run to analyze the effects of the heater temperature by doing FFT on run charts at 3 different temperatures (83℃, 87℃ and 93℃) as depicted in Figure 4.1.

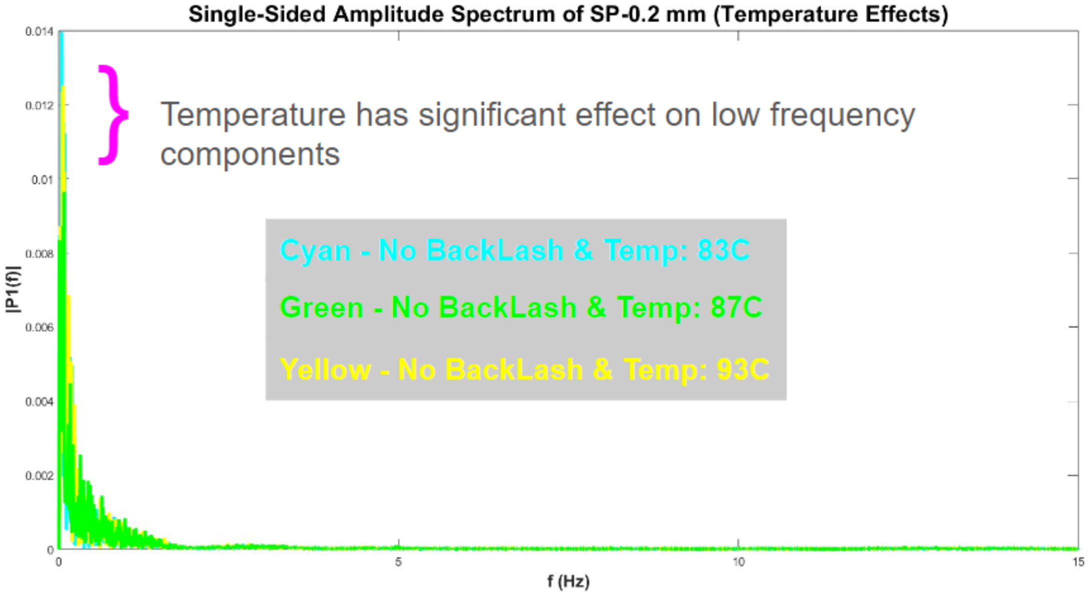


Figure 4.1. Temperature affecting FFT of Fiber Diameter

From this experiment it was seen that the magnitude of the low dominating frequencies reduced with increasing temperature until 93℃. It was observed from the experiment that 93℃ exceeds the higher bound on temperature for this PLC FrED as at this temperature the tension becomes unfavorable since the weight of the viscous fiber causes loss in tension and leads to some low frequency variability. Therefore, it was chosen to run subsequent experiments on the pseudo-optimized temperature of 87℃

4.3 Optimized Operating Parameters

The diameter data was taken at 87℃ and 85℃ for this timing belt system and compared in Figure 4.2.

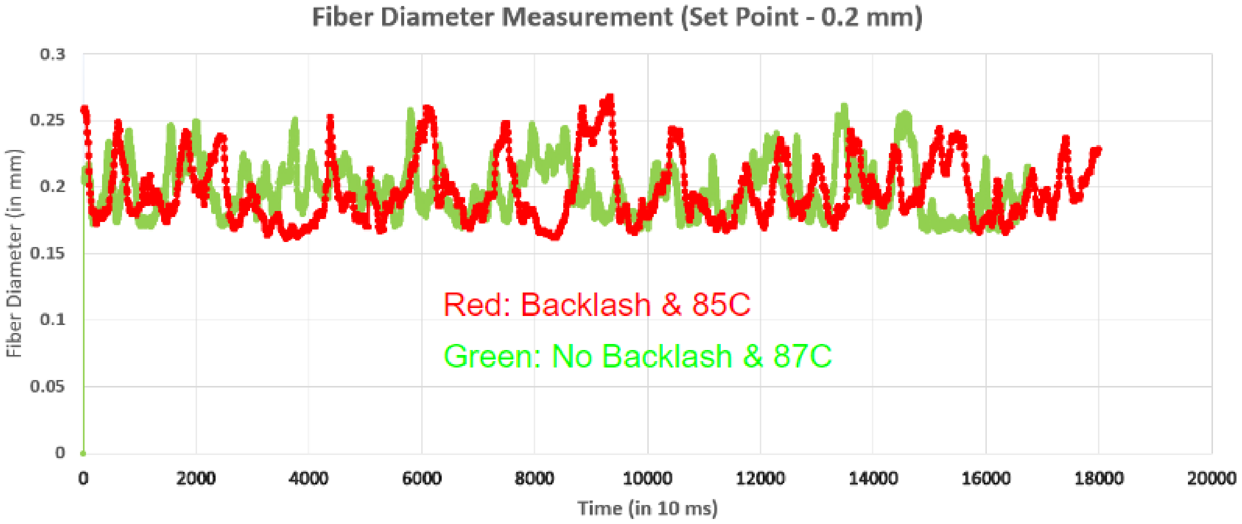


Figure 4.2. Fiber Diameter Run Chart (Combined Effects of Backlash and Temperature)

It can be evidently seen that some of the low-frequency components are reduced while some of the high-frequency components are back. This is further evident from the FFT plots as shown in Figure 4.3 and 4.4.

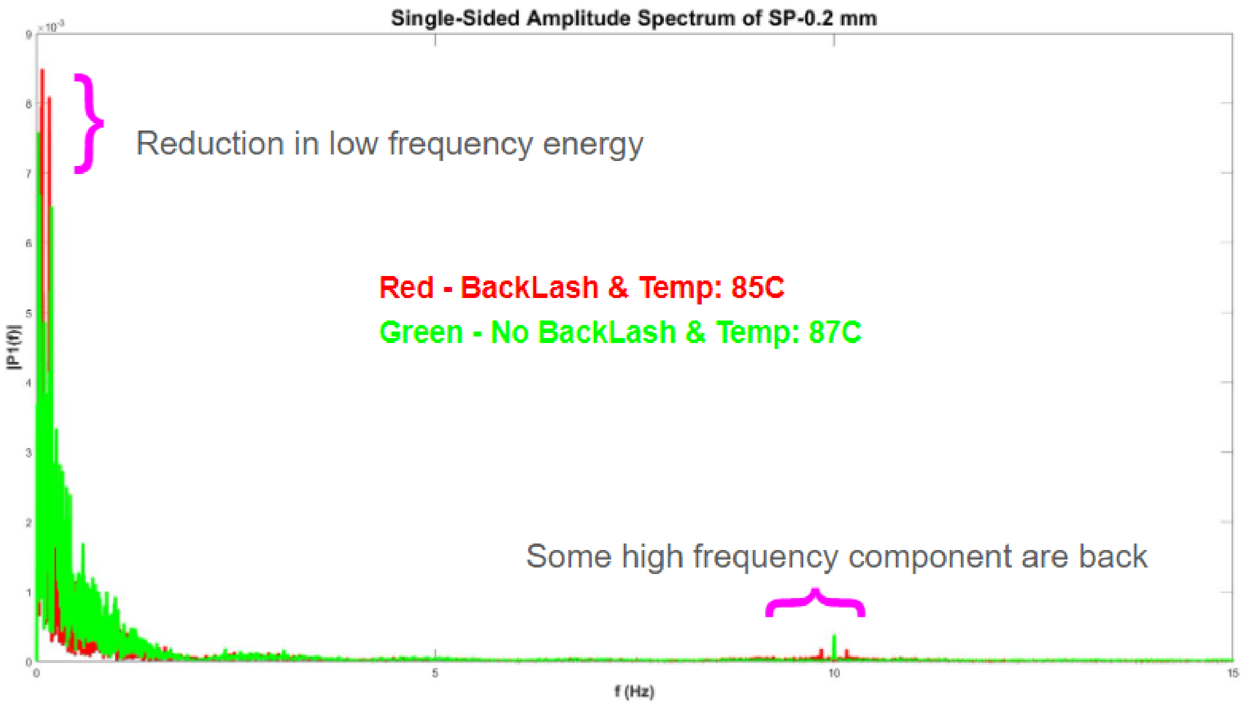


Figure 4.3. FFT of Fiber Diameter Run Chart (Combined Effects of Backlash and Temperature)

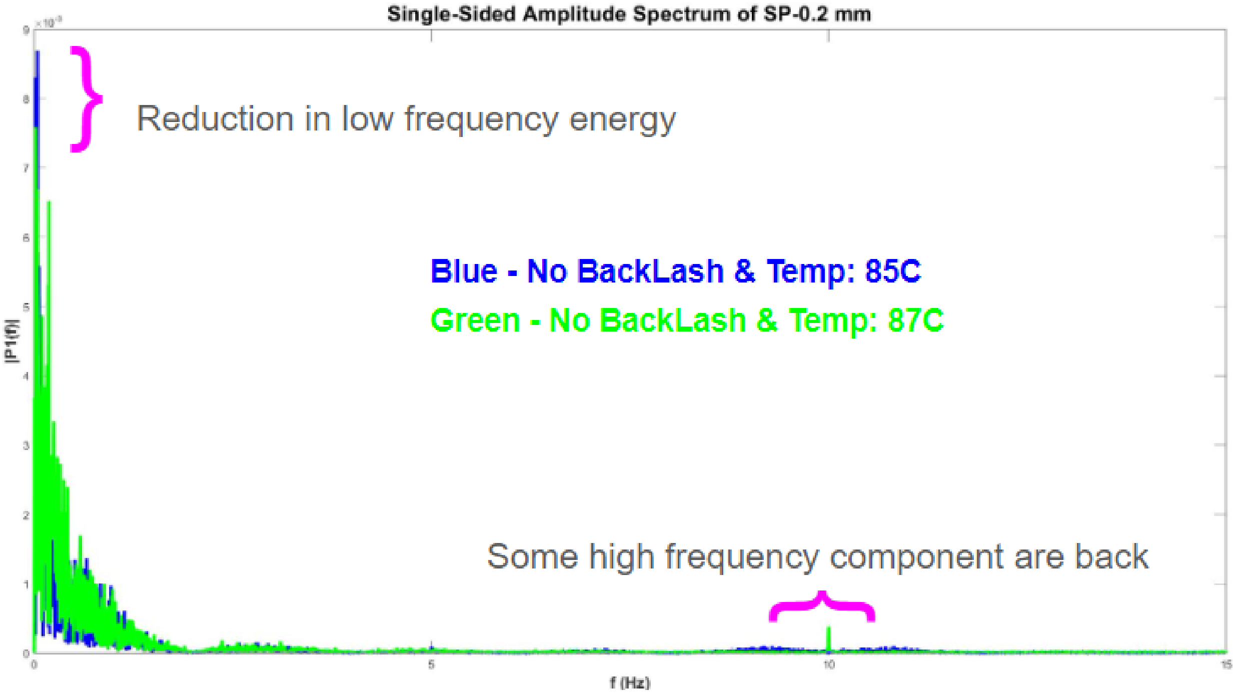
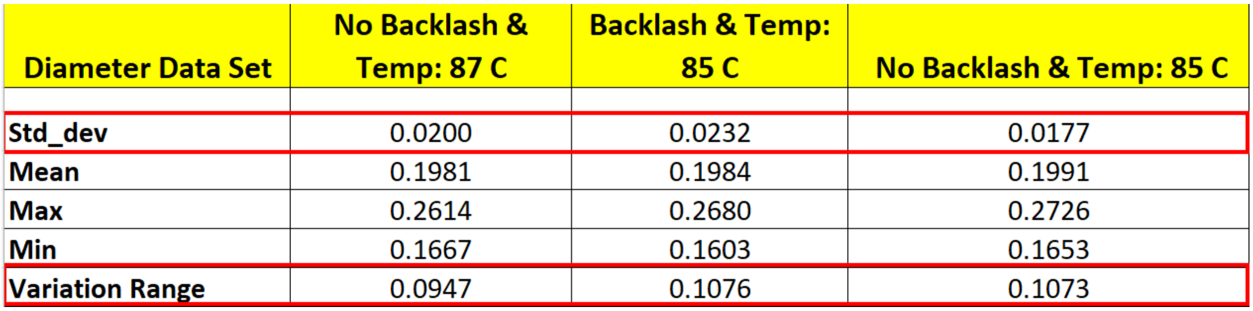


Figure 4.4. FFT of Fiber Diameter Run Chart (Effects of Temperature)

The main source of this high frequency component in the “No Backlash & Temp: 87℃” is hypothesized to be high variability in tension. Since the timing belt air-cooled system is different in the height and the lateral distance from the heater block, the roller length has been adjusted to bring back some tension that was lost in changing the height/lateral length of the platform. However, the hypothesis is that not the same level of tension was achieved as the leadscrew/pulley mass system. From this experiment, it was determined to add new functional requirements of a tension sensor and make the length and height of the roller support adjustable (see implementation in Figure 4.5). This would allow for the optimum tension to be discovered by a quick trial and test method.

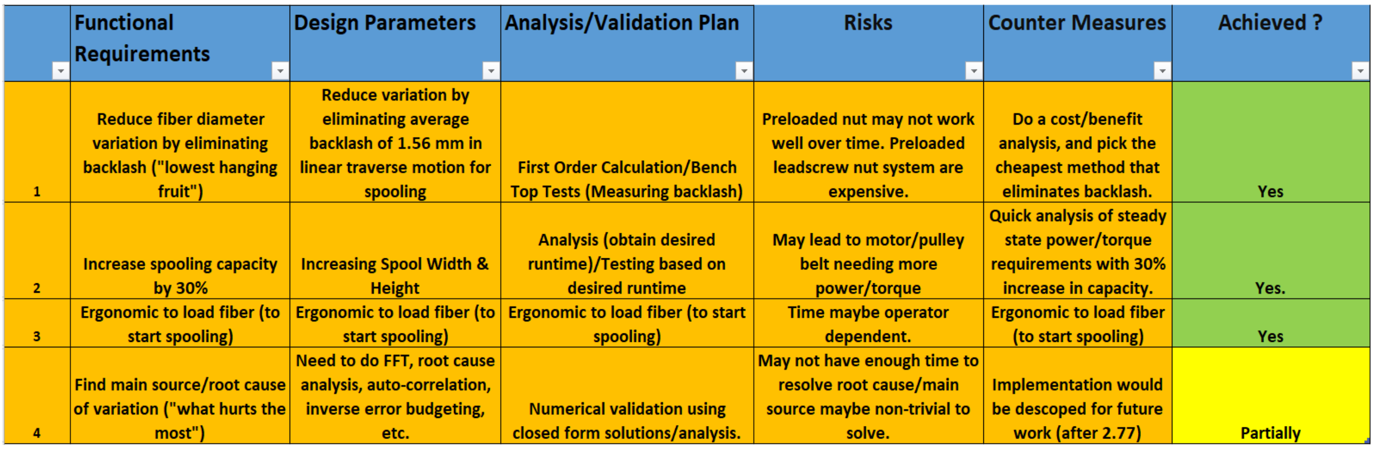
Table 4.1. Summary of the combined effects of optimizing temperature/eliminating backlash



Diameter Data Set	No Backlash & Temp: 87 C	Backlash & Temp: 85 C	No Backlash & Temp: 85 C
Std_dev	0.0200	0.0232	0.0177
Mean	0.1981	0.1984	0.1991
Max	0.2614	0.2680	0.2726
Min	0.1667	0.1603	0.1653
Variation Range	0.0947	0.1076	0.1073

It can be seen from table 4.1 above that the variation range of the “No Backlash Temp: 87℃” has the best variation but also the worst mean. The reduction in variation resulted from eliminating backlash and increasing the operating temperature to 87℃”. The worst mean resulted from using a PID controller that had not been tuned as well as possible (after switching out the lead screw system with the timing belt traverse system). In summary, FFT shows an 11% reduction in peak frequency energy, and fiber diameter variation reduced by 12%. The ergonomic design explained in Chapter 3 resulted in an improved spooling time by 91%, and increased spooling capacity by 144%. The summary of the FREDPARC table introduced in section 3.2 is shown in table 4.2. It concludes that all except the functional requirement #4 was fully met. Function requirement 4 was partially met, as the root cause of high variation was identified to be non-linearly dependent on heating chamber temperature and tension.

Table 4.2. Final FREDPARC Table



	Functional Requirements	Design Parameters	Analysis/Validation Plan	Risks	Counter Measures	Achieved ?
1	Reduce fiber diameter variation by eliminating backlash ("lowest hanging fruit")	Reduce variation by eliminating average backlash of 1.56 mm in linear traverse motion for spooling	First Order Calculation/Bench Top Tests (Measuring backlash)	Preloaded nut may not work well over time. Preloaded leadscrew nut system are expensive.	Do a cost/benefit analysis, and pick the cheapest method that eliminates backlash.	Yes
2	Increase spooling capacity by 30%	Increasing Spool Width & Height	Analysis (obtain desired runtime)/Testing based on desired runtime	May lead to motor/pulley belt needing more power/torque	Quick analysis of steady state power/torque requirements with 30% increase in capacity.	Yes.
3	Ergonomic to load fiber (to start spooling)	Ergonomic to load fiber (to start spooling)	Ergonomic to load fiber (to start spooling)	Time maybe operator dependent.	Ergonomic to load fiber (to start spooling)	Yes
4	Find main source/root cause of variation ("what hurts the most")	Need to do FFT, root cause analysis, auto-correlation, inverse error budgeting, etc.	Numerical validation using closed form solutions/analysis.	May not have enough time to resolve root cause/main source maybe non-trivial to solve.	Implementation would be descoped for future work (after 2.77)	Partially

4.4 Adjustable Pulley/Roller Design

It was discovered that while performing the experiments detailed in section 4.2 and 4.3 that temperature and tension are intrinsically connected in a non-linear fashion in the PLC FrED system. While higher temperatures (more than 83℃”) results in lower diameter variation, it causes a loss in tension due to loss in viscosity. Some of this loss in tension is counteracted by introducing an additional degree of freedom in the rollers/pulley. With this degree of freedom of the roller/pulley, the tension could be increased enough to be able to increase the heating chamber temperature to 87℃”.

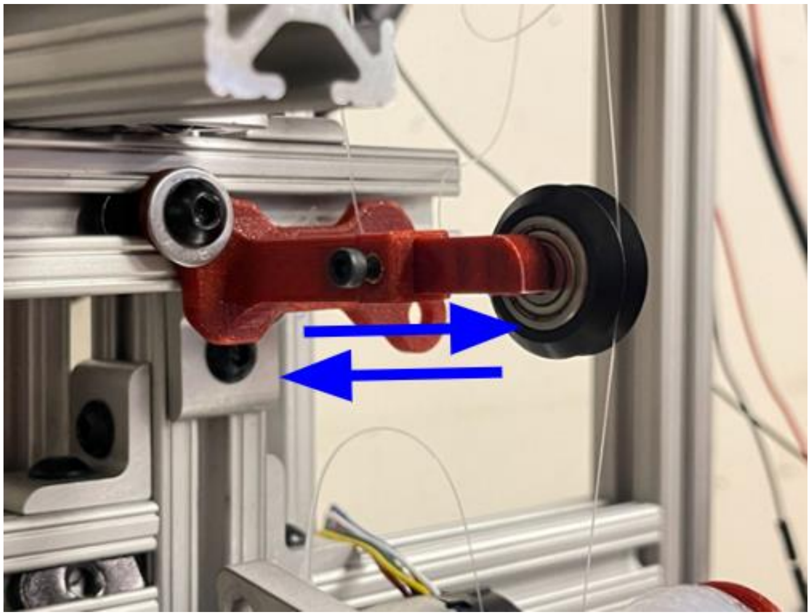


Figure 4.5. Adjustable roller to add/remove tension.

This research didn’t get to the point of installing a tensioner and put in additional degree of freedom/mechanisms and feedback loop to sense and prevent loss in tension with increasing temperature.

4.5 Mico-structure Thermal Dependencies

The micro-structure of the fiber was inspected using the Scanning Electron Microscope (SEM). At the micro-meter level, there was a clear distinction seen between fibers at 83℃” (Figure 4.6) and 87℃” (Figure 4.7).

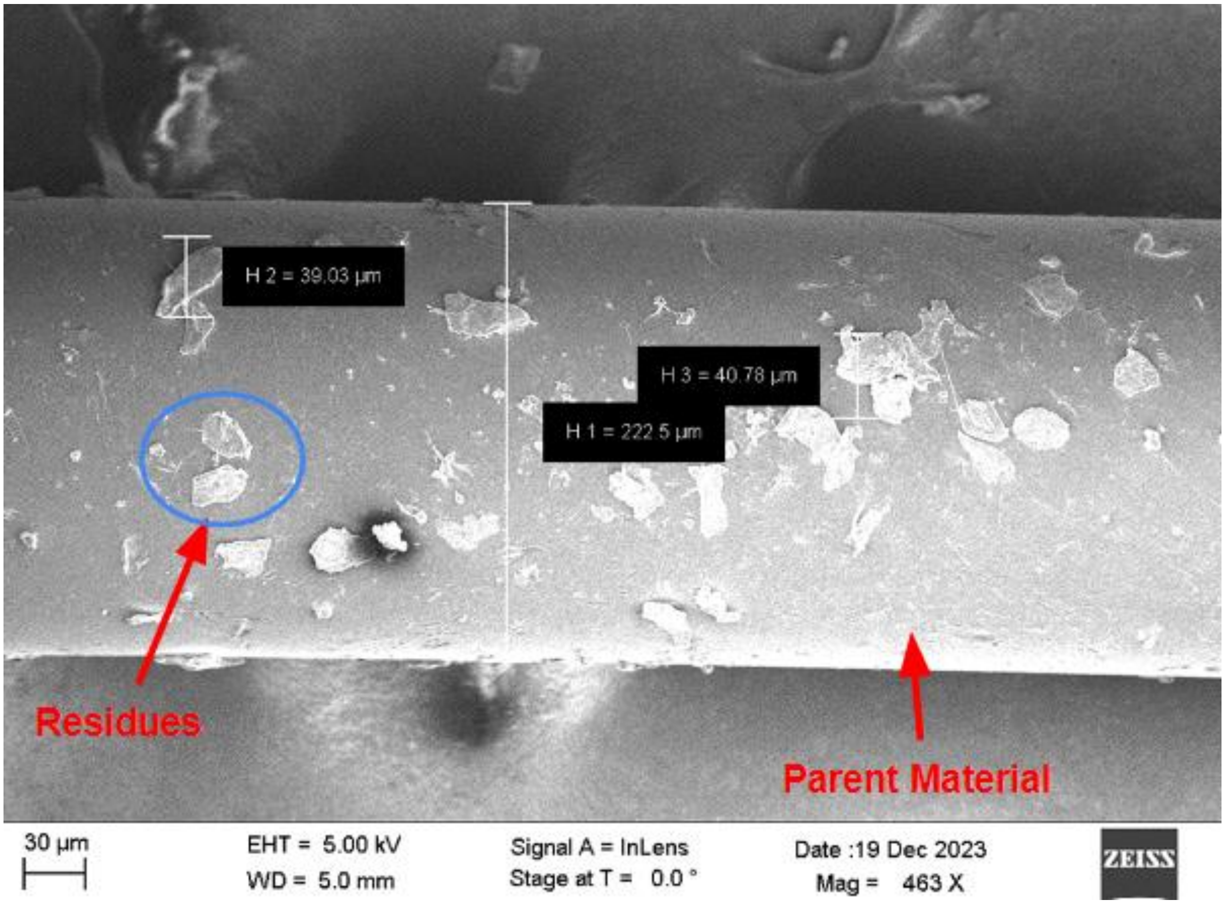


Figure 4.6. SEM Images of Fiber. Melt Temperature: 83 ℃*.*

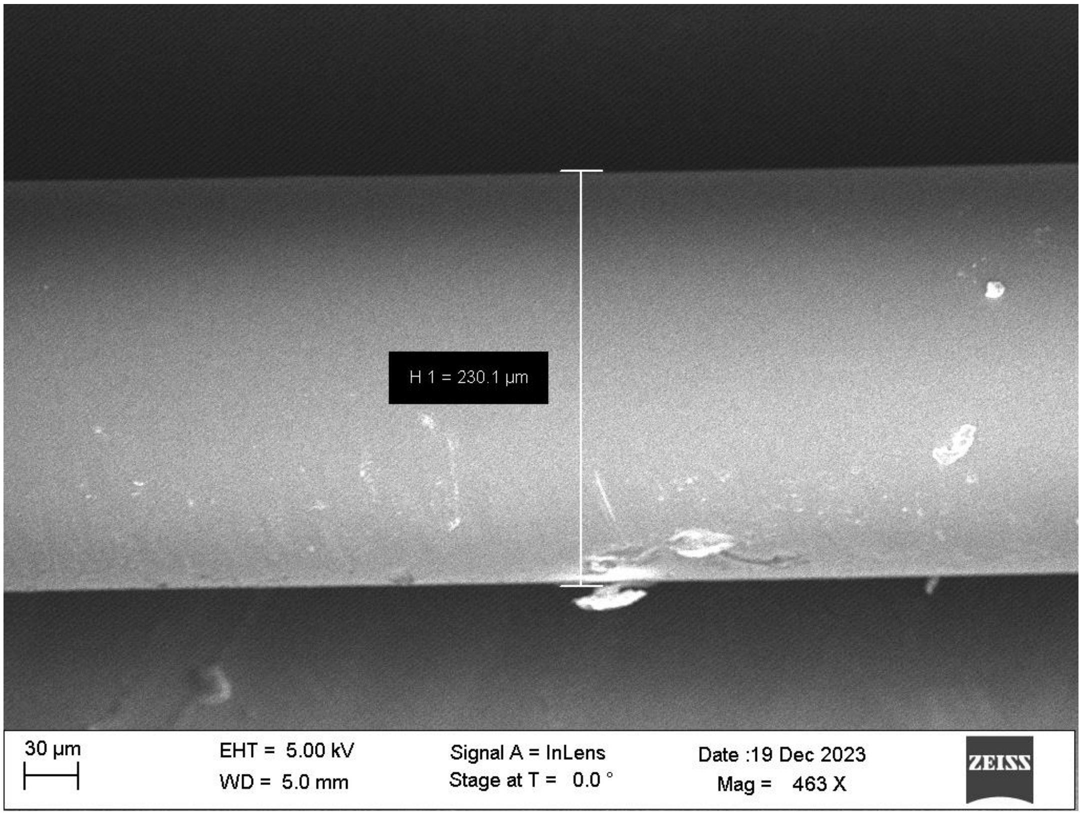


Figure 4.7. SEM Images of Fiber. Melt Temperature: 87 °C.

This distinction comes from the higher temperature making the preform material go to a more viscous/liquid state. This causes the residues as highlighted in figure melt away and become uniform with the parent material. This leads to less variation in the diameter measurement. Furthermore, another issue that became evident after putting the fiber under SEM: the fiber diameter has a deformation which comes two fibers sticking together and being split apart. Therefore, it is important to ensure the fibers don't get tangled and damaged when they are being measured.

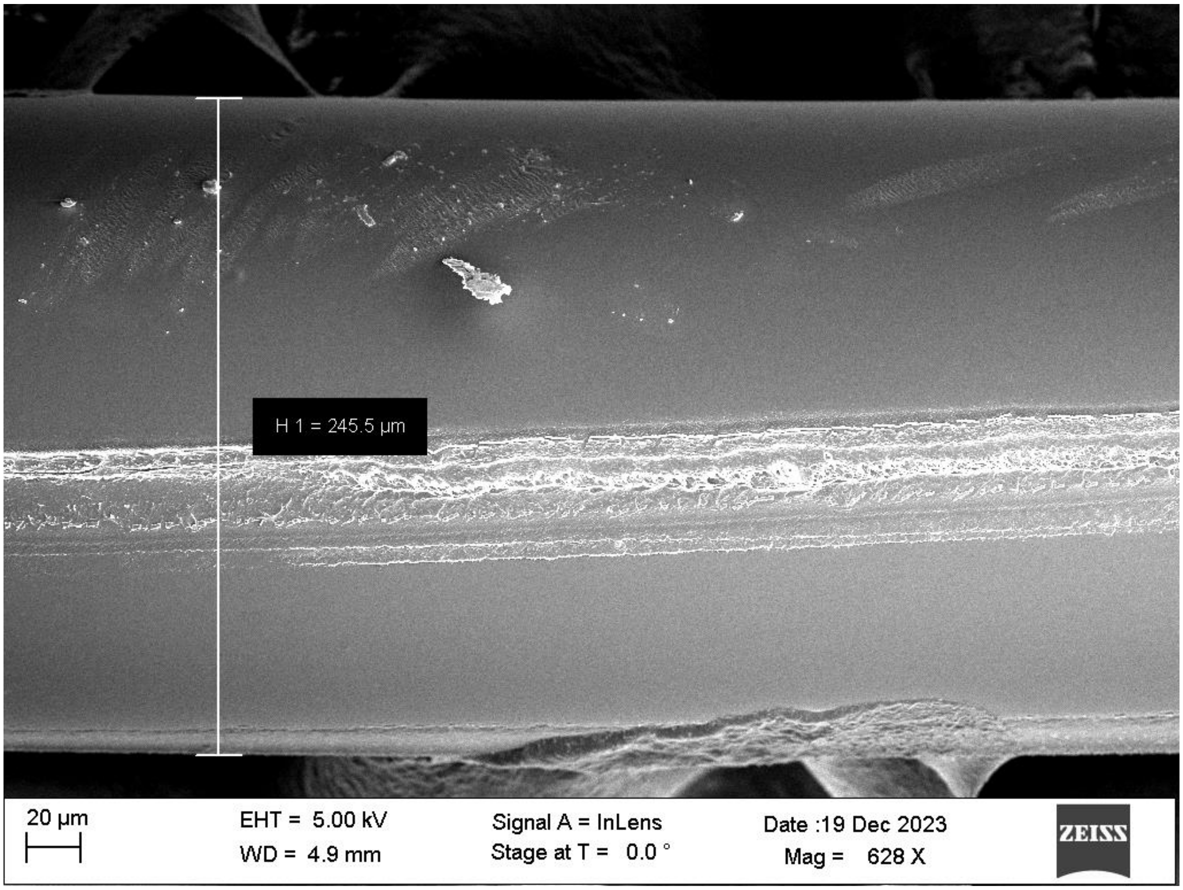


Figure 4.8. SEM Images of Damaged Fiber.

This page is intentionally left blank.

Chapter 5 PLC Electrical Systems

The electrical control architecture is built on an Allen-Bradley's CompactLogix 5380 industrial PLC. Rockwell 5069-L320ER PLC (Programmable Logic Controllers) is utilized. It is connected to 4 modules including DC Input module 5069-IB16F, DC Output module, 5069-OB16F, counter module 5069-HSC2x084, Analog Input module, 5069-IY4. The system is connected to laptop and the programming is done using Studio 5000. Both 12V power supply and 24V power supplies are used. The wiring diagram of the entire system is shown in Figure 5.1. Three different PID controls are used to control temperature control, DC motor speed control, and the overall fiber diameter control. The block diagram Figure 5.3 shows the control logic of FrED. This section summarizes some of the overlapped work that had been done with Zhang to build the electrical hardware (4). This section also points to the sources of some of the details that have been independently carried out by Zhang (4).

5.1 Electrical Hardware Architecture.

The overall wiring diagram of the FrED system is shown in Figure 5.1.

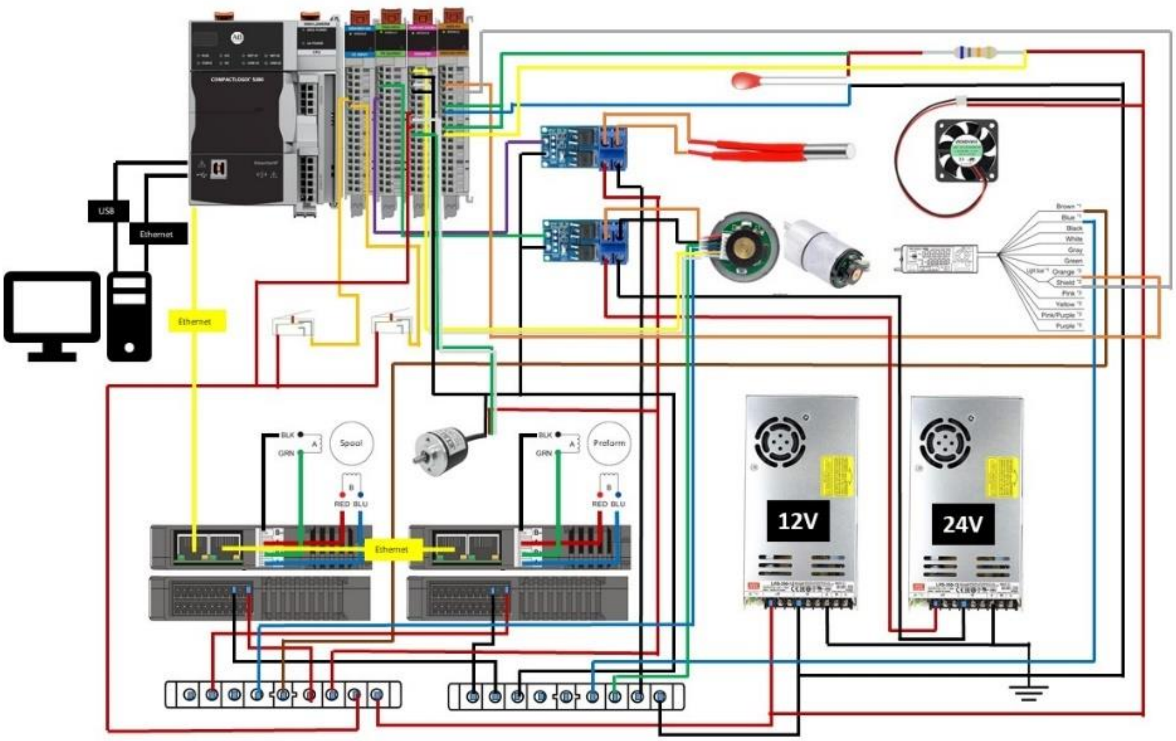


Figure 5.1. PLC FrED wiring diagram.

From the wiring diagram, it can be seen that the FrED system is a non-trivial electrical system. The main modules of the PLC hardware are shown in Figure 5.2

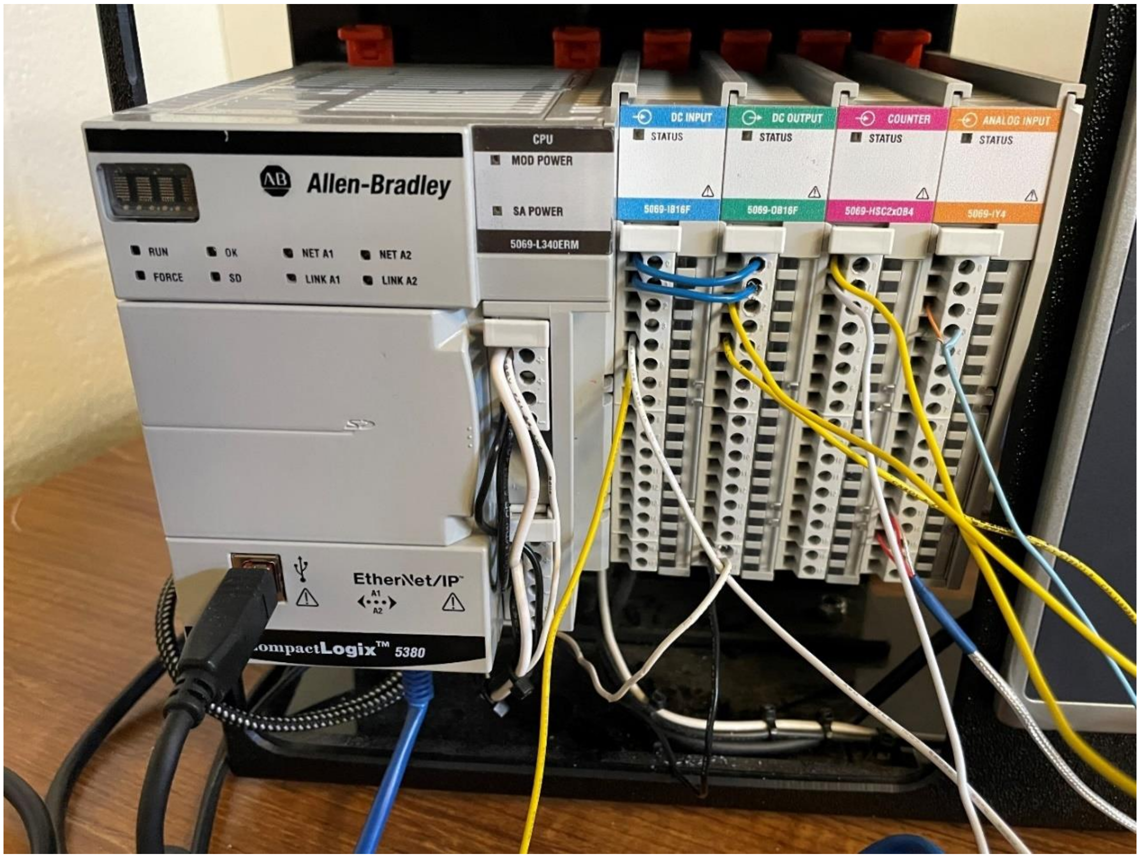


Figure 5.2. Allen Bradley 5380 Compact Logix Modules.

5.2 PID Controller Architecture

The FrED system is controlled by three PID (Proportional-Integral-Derivative) controllers implemented on Allen Bradley Rockwell 5069-L320ER/CompactLogix 5380 PLC (Programmable Logic Controllers). The PID control block diagram is shown below in Figure 5.3.

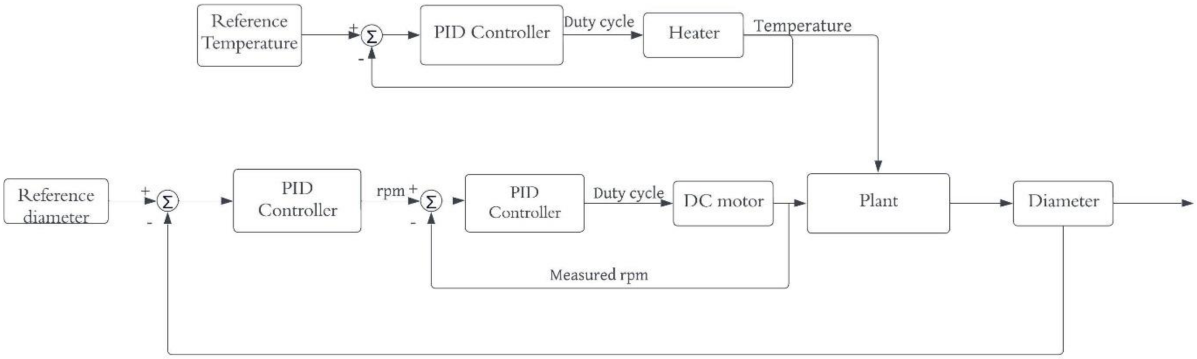


Figure 5.3. Three PID Controlled Loops Governing Overall FrED

There are three PID feedback loops in FrED. The temperature PID feedback loop is used to maintain the constant temperature of the heater block where the preform material melts to a viscous state. The loop is fed in a reference setpoint temperature and measured temperature and outputs a pulse width modulated (PWM) voltage to the drivers used to heat the heater block. The motor PID feedback loop is used to control the speed of the motor. The loop is fed in a reference setpoint motor speed and measured speed and outputs a pulse width modulated (PWM) voltage to the drivers used to power the motor. The diameter control PID loop is used to control the diameter of the fiber. The loop is fed in a reference setpoint fiber diameter and measured diameter and outputs a setpoint reference motor speed to the motor PID feedback loop. The control values (CV), which are the output of a PID loop, is given in eq (5.1) and eq (5.2). Note: these equations are utilized in all the PID loops described in this section.

$$CV_n = CV_{n+1} + K_P \Delta E + \frac{K_I}{60} E \Delta t + 60 K_D \frac{E_n - 2E_{n-1} + E_{n-2}}{\Delta t} \tag{5.1}$$

$$CVEU = CV \times \frac{CVEU_{Max} - CVEU_{Min}}{100} \tag{5.2}$$

The logic behind PID is based on the error (E) which is the difference between the setpoint values (temperature, fiber diameter of RPM of the spool motor) at time steps corresponding to $n, n-1, n-2$ ....

There are namely three parameters that govern the behavior of the PID loop. Namely the Proportionality parameter (K_P), the Integral parameter (K_I), and the Derivative parameter (K_D).

The tuned parameters of three PID loops are briefly detailed below:

Table 5.1. PID Parameters

PID Loop	KP	KI	KD
Temperature Feedback Loop	12.6	0.3	.075
Motor Feedback Loop	5.5	80	.01
Diameter Control Feedback Loop	2.5	90	.03

Although the PID feedback loop has been tuned by a heuristic method such as Zeigler Nichols, the performance of the temperature and motor PID loops is performing much better relative to the diameter PID loop (4). The Zeigler Nichols methods tuning process has been detailed in section 5.3.1.

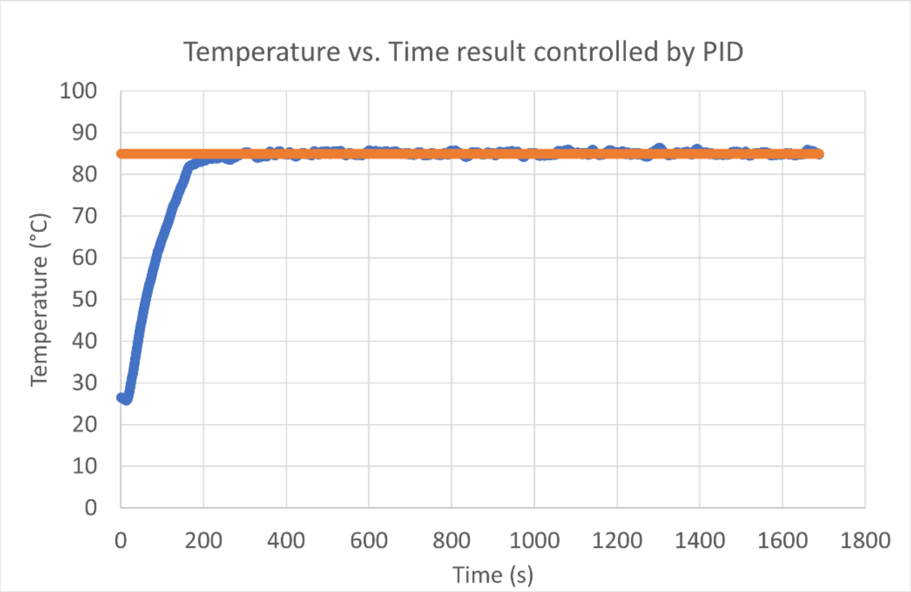


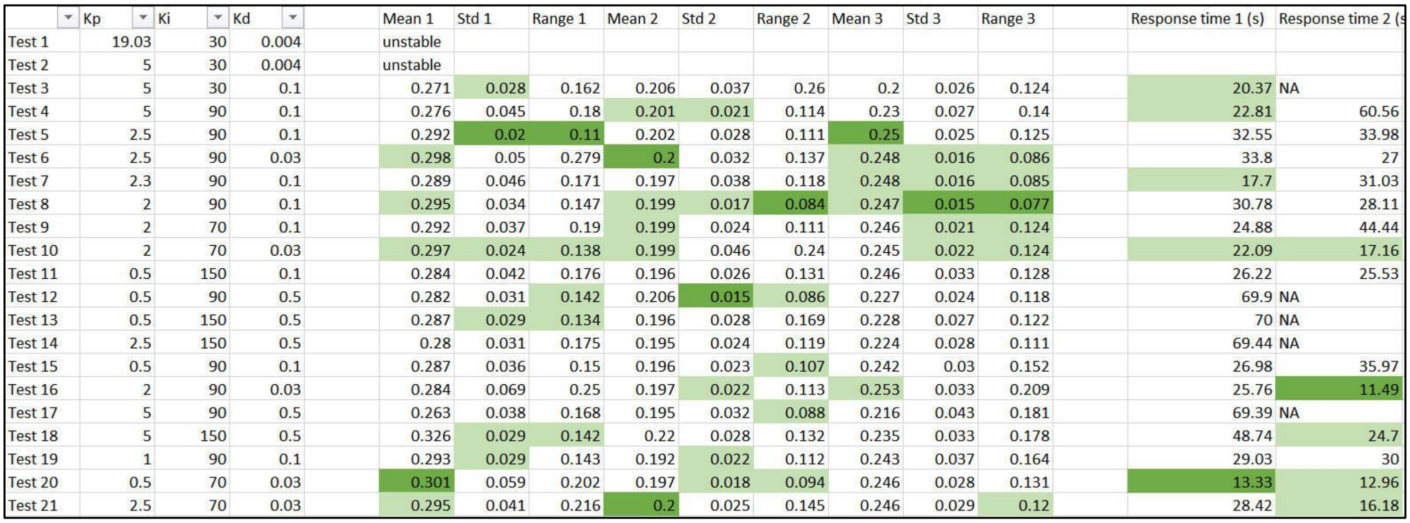
Figure 5.4. Temperature PID Control/Response Time

5.3 PID Heuristic Tuning Methods

5.3.1 Zeigler Nichols Tuning

Zhang performed detailed analysis of finding the optimal PID values using the Zeigler Nicholas tuning method (4). Given the FrED system had some limitations (explained more in Chapter 3) in the diameter range it could achieve (0.15 mm - 0.35 mm), Zhang experimentally attempted to find the optimal values for Spool DC motor control and overall diameter control (4).

Table 5.2. Table depicting the performance fiber diameter controller with various gain (4).



	Kp	Ki	Kd		Mean 1	Std 1	Range 1	Mean 2	Std 2	Range 2	Mean 3	Std 3	Range 3		Response time 1 (s)	Response time 2 (s
Test 1	19.03	30	0.004		unstable											
Test 2	5	30	0.004		unstable											
Test 3	5	30	0.1		0.271	0.028	0.162	0.206	0.037	0.26	0.2	0.026	0.124		20.37	NA
Test 4	5	90	0.1		0.276	0.045	0.18	0.201	0.021	0.114	0.23	0.027	0.14		22.81	60.56
Test 5	2.5	90	0.1		0.292	0.02	0.11	0.202	0.028	0.111	0.25	0.025	0.125		32.55	33.98
Test 6	2.5	90	0.03		0.298	0.05	0.279	0.2	0.032	0.137	0.248	0.016	0.086		33.8	27
Test 7	2.3	90	0.1		0.289	0.046	0.171	0.197	0.038	0.118	0.248	0.016	0.085		17.7	31.03
Test 8	2	90	0.1		0.295	0.034	0.147	0.199	0.017	0.084	0.247	0.015	0.077		30.78	28.11
Test 9	2	70	0.1		0.292	0.037	0.19	0.199	0.024	0.111	0.246	0.021	0.124		24.88	44.44
Test 10	2	70	0.03		0.297	0.024	0.138	0.199	0.046	0.24	0.245	0.022	0.124		22.09	17.16
Test 11	0.5	150	0.1		0.284	0.042	0.176	0.196	0.026	0.131	0.246	0.033	0.128		26.22	25.53
Test 12	0.5	90	0.5		0.282	0.031	0.142	0.206	0.015	0.086	0.227	0.024	0.118		69.9	NA
Test 13	0.5	150	0.5		0.287	0.029	0.134	0.196	0.028	0.169	0.228	0.027	0.122		70	NA
Test 14	2.5	150	0.5		0.28	0.031	0.175	0.195	0.024	0.119	0.224	0.028	0.111		69.44	NA
Test 15	0.5	90	0.1		0.287	0.036	0.15	0.196	0.023	0.107	0.242	0.03	0.152		26.98	35.97
Test 16	2	90	0.03		0.284	0.069	0.25	0.197	0.022	0.113	0.253	0.033	0.209		25.76	11.49
Test 17	5	90	0.5		0.263	0.038	0.168	0.195	0.032	0.088	0.216	0.043	0.181		69.39	NA
Test 18	5	150	0.5		0.326	0.029	0.142	0.22	0.028	0.132	0.235	0.033	0.178		48.74	24.7
Test 19	1	90	0.1		0.293	0.029	0.143	0.192	0.022	0.112	0.243	0.037	0.164		29.03	30
Test 20	0.5	70	0.03		0.301	0.059	0.202	0.197	0.018	0.094	0.246	0.028	0.131		13.33	12.96
Test 21	2.5	70	0.03		0.295	0.041	0.216	0.2	0.025	0.145	0.246	0.029	0.12		28.42	16.18

Zhang concluded that the best performing gains combinations are test 8, $K_P - 2$, $K_I - 90$ and $K_D - 0.1$for the best diameter accuracy and test 10, $K_P - 2$, $K_I - 70$ and $K_D - 0.003$ for the shortest response time (4). The figure below depicts the performance of the controller using the PID gains from test 8.

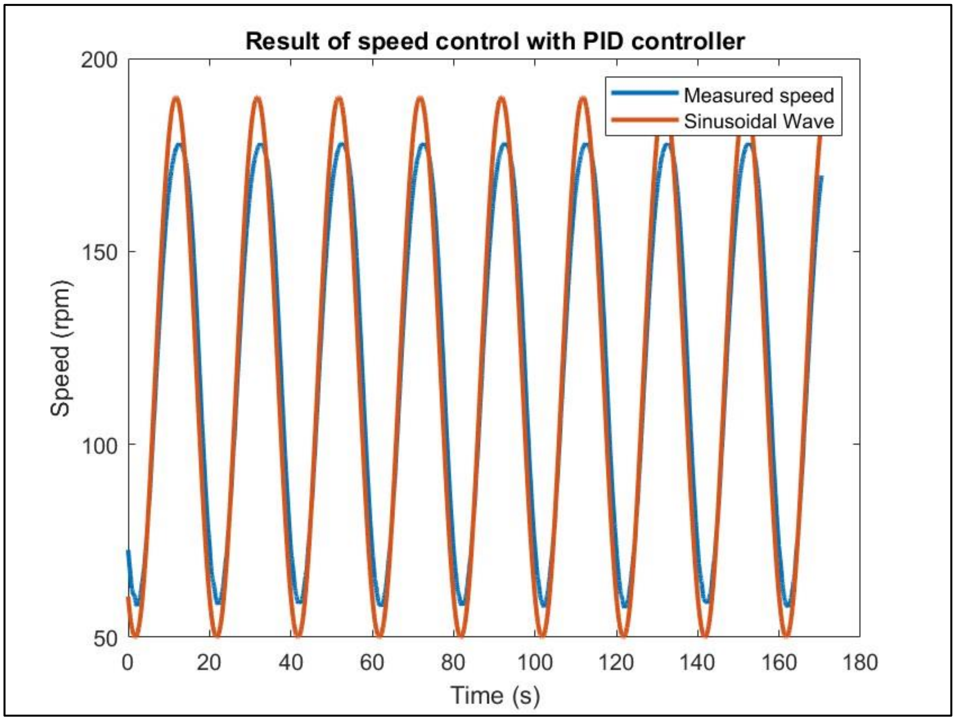


Figure 5.5. Spool DC Motor PID Speed Control (4)

Based on the experiments conducted the best performing gains combinations are test 8, which has $K_P = 2$, $K_I = 90$ and $K_D = 0.1$ for the best diameter accuracy and test 10 which has $K_P = 2$, $K_p = 70$ and $K_d = 0.003$ for the shortest response time (4). The figure below depicts the performance of the controller using the PID gains from test 8.

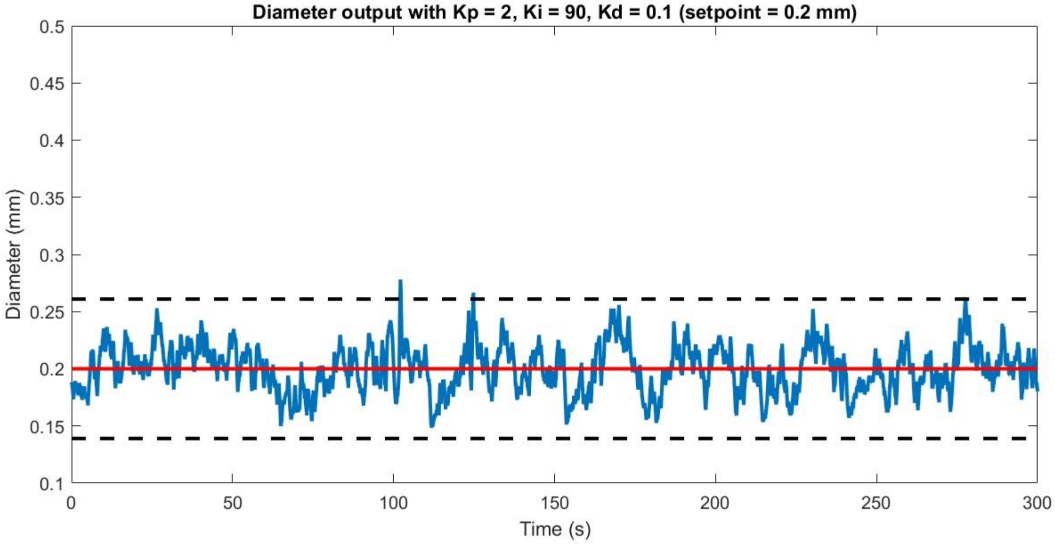


Figure 5.6. Diameter Control Setpoint: 0.2 mm (4)

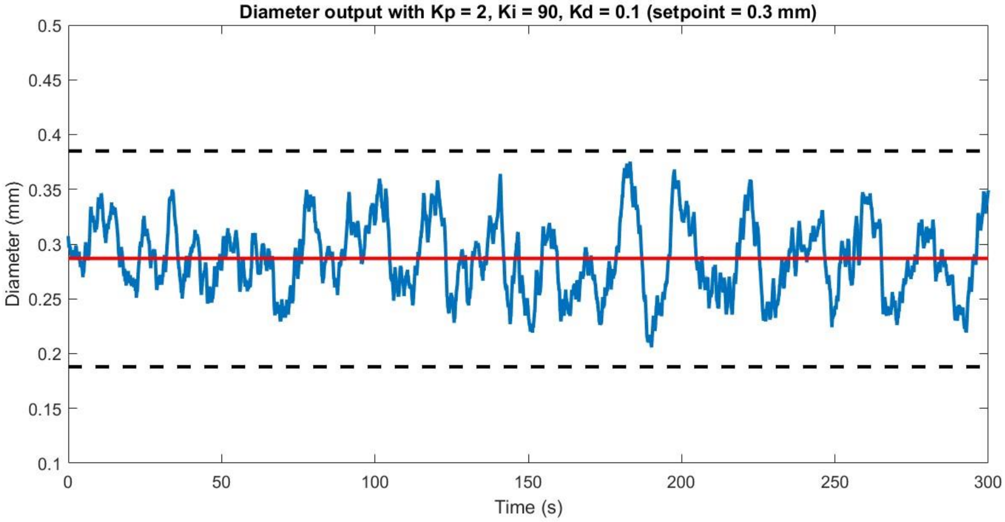


Figure 5.7. Diameter Control Setpoint: 0.3 mm (4)

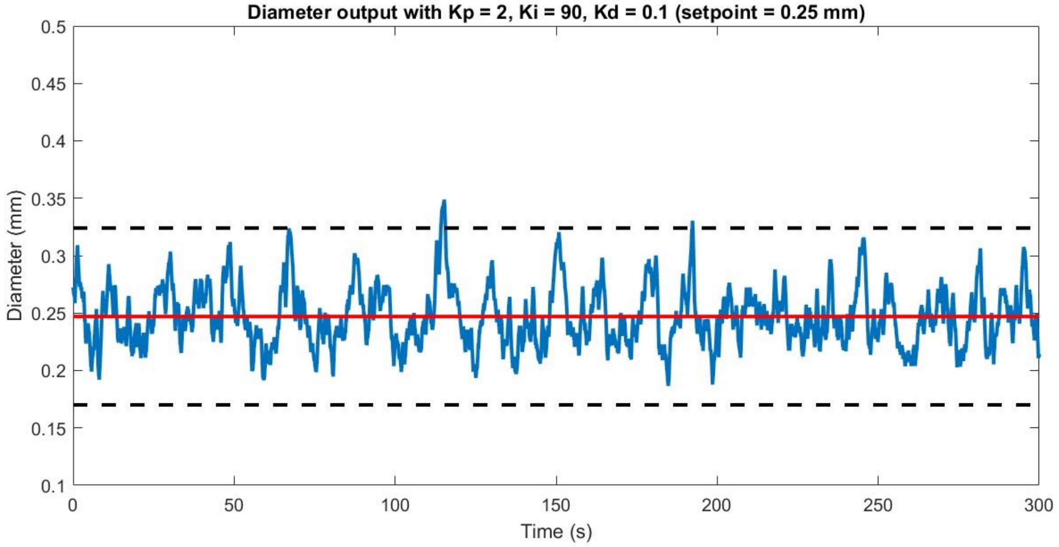


Figure 5.8. Diameter Control Setpoint: 0.25 mm (4)

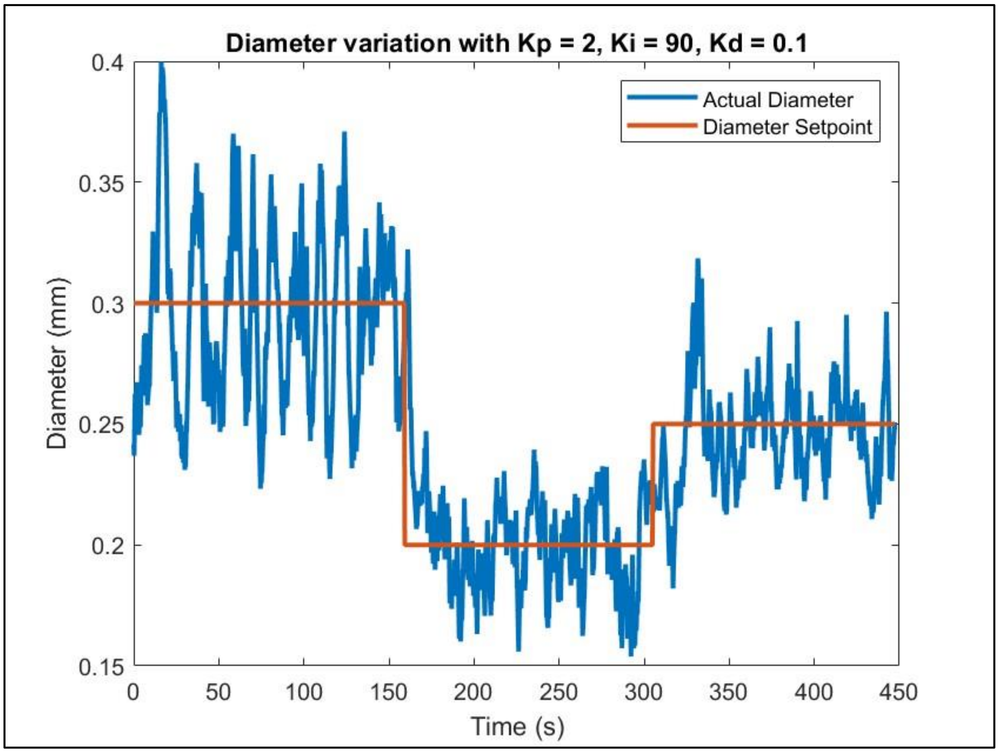


Figure 5.9. Diameter Control Various Setpoints (Zeigler Nichols Tuning) (4)

The performance of the controller tuned using the Zeigler Nichols method can be summarized in table 5.3.

Table 5.3. Table depicting the average, standard deviation, and range of the fiber diameter for each set point during 3 tests. (Zeigler Nicholas Tuning) (4)

Setpoint (mm)	Average (mm)	Standard Deviation (mm)	Range (mm)	Relative error (%)
0.30	0.287	0.033	0.170	4.3
0.25	0.247	0.026	0.162	1.2
0.20	0.200	0.020	0.130	0

5.3.2 Full Heuristic Tuning

Alternatively, a completely heuristic process developed on the following algorithm has been used to tune the PID parameters of the diameter.

i) Increase K_p until oscillation at a constant frequency occurs.

ii) Increase K_d while reducing K_p until almost all constant frequency oscillations are damped out. At this point, ensure K_d isn't so high that the response time when changing set points is too high.

iii) Make sure to remove any constant error by increasing K_i.

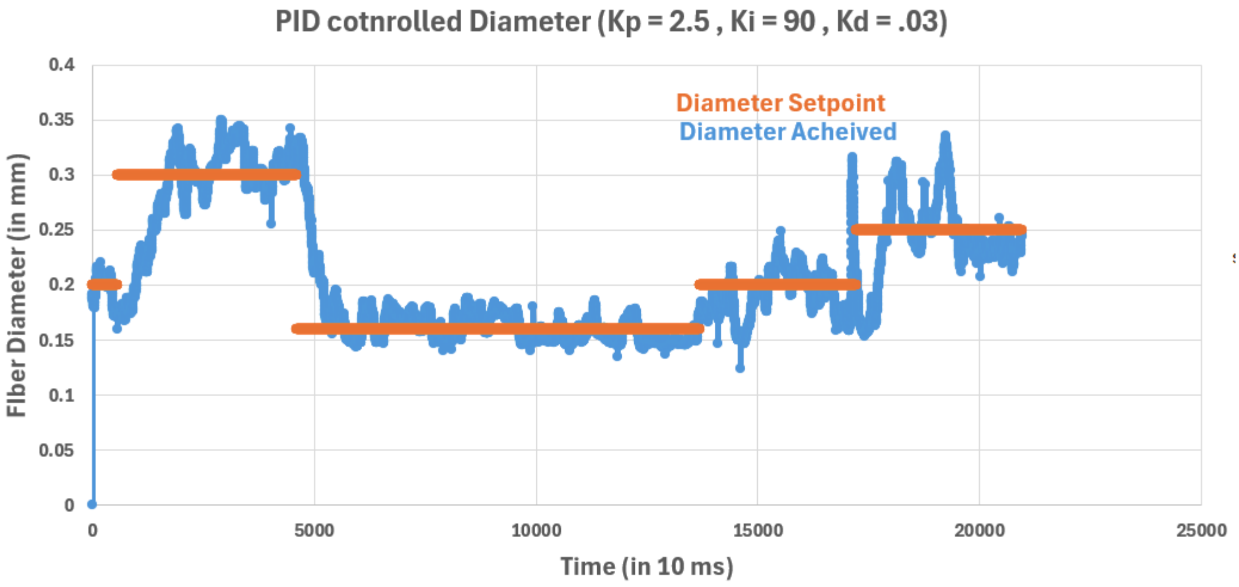


Figure 5.10. Diameter Control Various Setpoints (Full Heuristic Method)

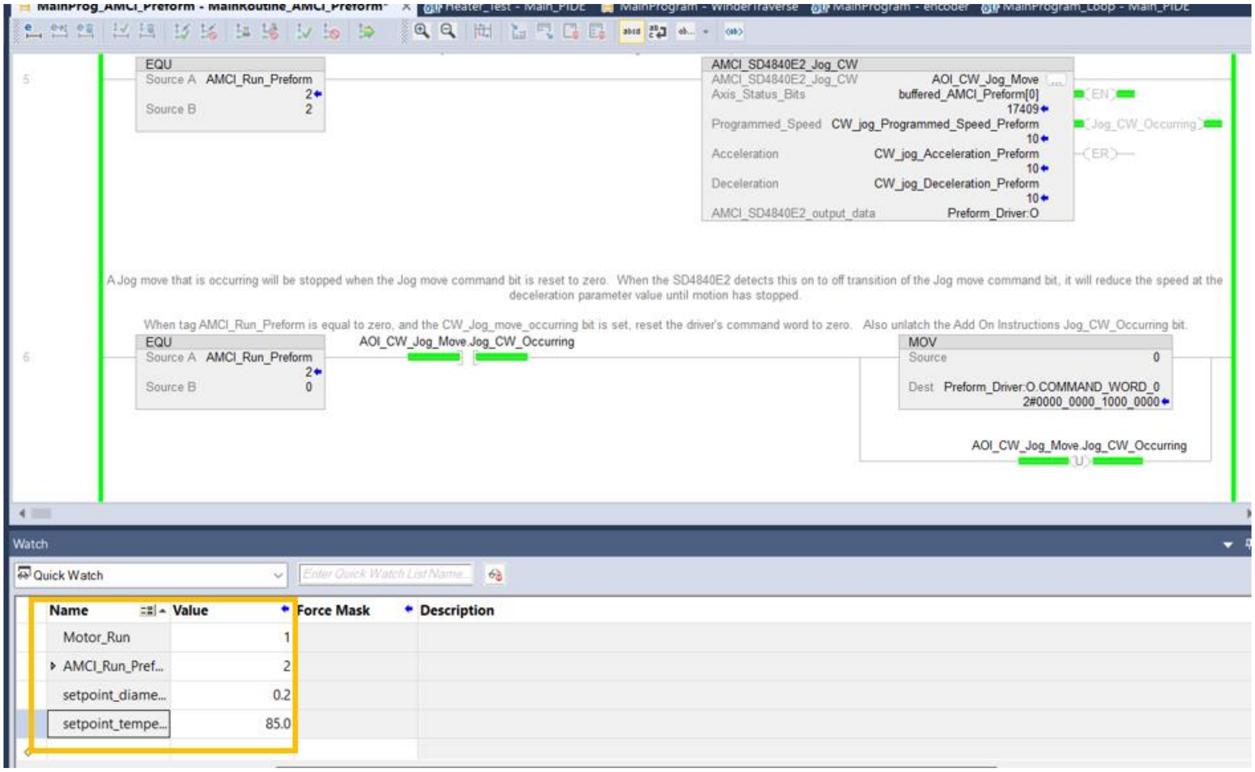
The tuning results of these two methods can be concluded to be comparable by comparing Figure 5.9 and Figure 5.10. In addition, Figure 5.10 shows the result of the tuning process which was also tuned at a higher operating temperature of 87C compared to the tuning process described in section 5.3.1. This may also be a factor in less reduced noise for the full heuristic tuning process. The reason behind this is with increased temperature low frequency components in the variation of the fiber diameter is reduced as concluded in Section 4. Table 5.3 can be compared with table 5.4 to compare the performance of Zeigler Nichols tuning method (as described in Section 5.3.1) and the full heuristic tuning method (as described in Section 5.3.2).

Table 5.4. Table depicting the average, standard deviation, and range of the fiber diameter for each set point during 3 tests. (Fully Heuristic Method)

Setpoint (mm)	Average (mm)	Standard Deviation (mm)	Range (mm)	Relative error (%)
0.30	0.302	0.017	0.100	0.667
0.25	0.245	0.020	0.120	2.00
0.20	0.200	0.010	0.045	0
0.16	0.161	0.0067	0.040	0.625

5.4 Studio 5000 Function Block/Ladder Logic Algorithms

This section highlights some of the key function block and ladder logic algorithms that were developed to run the FrED, establish PID control and subsequently pave path to establish learned control models. Function blocks are a much more intuitive and graphical way of writing PLC code and therefore it was used widely in this project to write programs. Even though there are hundreds of PLC programs that were made, imported, or reused in this research project, a couple key programs (written using function blocks) will be briefly discussed in this section. One of the key milestones in this project was to implement classical control (PID Loops as depicted in Figure 5.3). The three PID control programs are discussed in this section and depicted in Figure 5.14, 5.15, 5.21 and 5.24. The primary tags/control variables that are needed to control the PID control loop setups/programs are shown in Figure 5.11.


MainProg_AMCI_Preform - MainRoutine_AMCI_Preform*
Heater_Test - Main_PIDE
MainProgram - WinderTraverse
MainProgram - encoder
MainProgram_Loop - Main_PIDE
Watch
Quick Watch
Enter Quick Watch List Name...

```
5   EQU
    Source A  AMCI_Run_Preform
                             2
    Source B                 2

    AMCI_SD4840E2_Jog_CW
    AMCI_SD4840E2_Jog_CW                     AOI_CW_Jog_Move  (EN)
    Axis_Status_Bits              buffered_AMCI_Preform[0]
                                                    17409     (Jog_CW_Occurring)
    Programmed_Speed  CW_jog_Programmed_Speed_Preform
                                                       10     (ER)
    Acceleration          CW_jog_Acceleration_Preform
                                                       10
    Deceleration          CW_jog_Deceleration_Preform
                                                       10
    AMCI_SD4840E2_output_data           Preform_Driver:O
```

A Jog move that is occurring will be stopped when the Jog move command bit is reset to zero. When the SD4840E2 detects this on to off transition of the Jog move command bit, it will reduce the speed at the deceleration parameter value until motion has stopped.

When tag AMCI_Run_Preform is equal to zero, and the CW_Jog_move_occurring bit is set, reset the driver's command word to zero. Also unlatch the Add On Instructions Jog_CW_Occurring bit.

```
6   EQU                          AOI_CW_Jog_Move.Jog_CW_Occurring     MOV
    Source A  AMCI_Run_Preform                                        Source                                 0
                             2
    Source B                 0                                        Dest  Preform_Driver:O.COMMAND_WORD_0
                                                                                   2#0000_0000_1000_0000

                                                                      AOI_CW_Jog_Move.Jog_CW_Occurring
                                                                                     (U)
```

Name	Value	Force Mask	Description
Motor_Run	1		
▸ AMCI_Run_Pref...	2		
setpoint_diame...	0.2		
setpoint_tempe...	85.0		

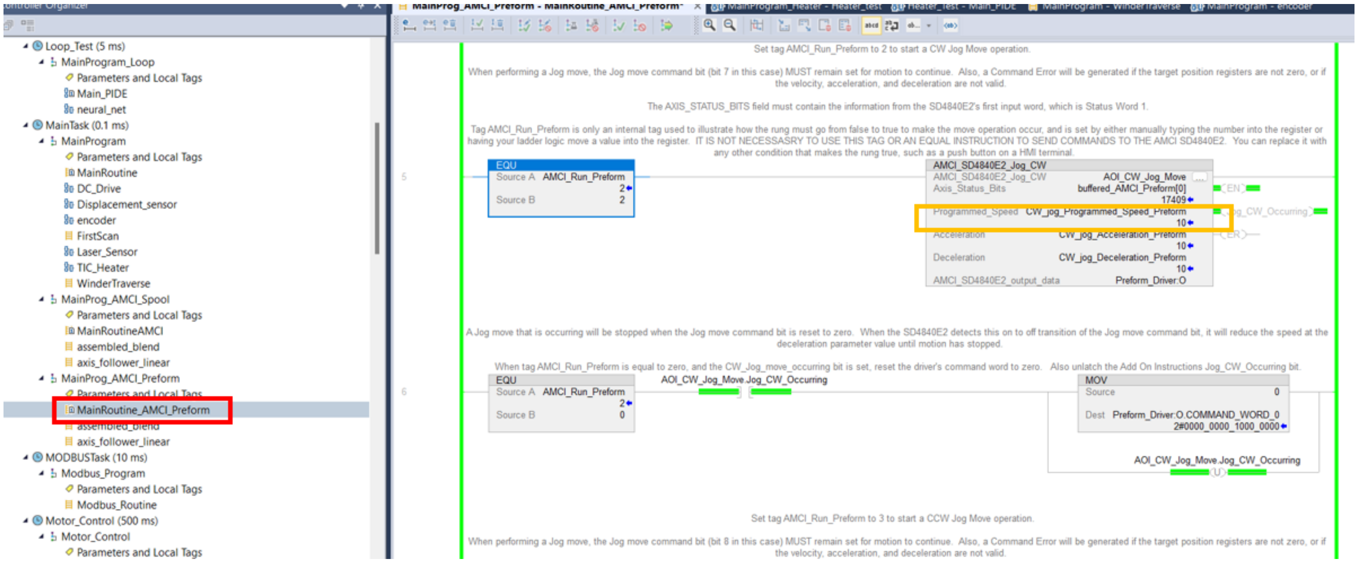
Figure 5.11. Important Tags for PID Control

Here switching “Motor Control” is a binary input variable and by inputting 1 or 0 will turn the DC Spool Motor on or off. Setting “AMCI Run Preform” variable to 2 makes the preform stepper motor turn clockwise causing the main gear to push the preform material into the heating chamber. The “setpoint_diameter” variable can be used to input the desired setpoint of the overall fiber PID control. It can be set from a range of 0.15 mm to 0.35 mm. The “setpoint_temperature” variable can be used to set the temperature ranging from 0C to 95C. This is the setpoint of the temperature PID control. It is advisable not to set the temperature beyond 100C as it will cause the heater chamber fixture (made from PLA 3D printed material) to soften and lose structural integrity.

5.4.1 Fiber Extrusion Process Algorithms

The various programs discussed in this section can be categorized to be responsible for the "Fiber Extrusion Process". The orange box represents important input or output tags/variables. The red box represents important objects/abstractions.

The preform stepper motor used the object called AMCI (Advanced Micro-Controls Inc.) to control the stepper motor that pushes the preform material into the heating chamber. The orange highlighted box here represents the speed of the motor/main gear 10 corresponds to 0.6 rpm and 5 corresponds to 0.3 rpm. It was experimentally shown that the PLC FrED is the most stable when this speed is between 0.3 rpm and 0.6 rpm. Over this speed, the fiber material tends to lose tension.



Set tag AMCI_Run_Preform to 2 to start a CW Jog Move operation.

When performing a Jog move, the Jog move command bit (bit 7 in this case) MUST remain set for motion to continue. Also, a Command Error will be generated if the target position registers are not zero, or if the velocity, acceleration, and deceleration are not valid.

The AXIS_STATUS_BITS field must contain the information from the SD4840E2's first input word, which is Status Word 1.

Tag AMCI_Run_Preform is only an internal tag used to illustrate how the rung must go from false to true to make the move operation occur, and is set by either manually typing the number into the register or having your ladder logic move a value into the register. IT IS NOT NECESSASRY TO USE THIS TAG OR AN EQUAL INSTRUCTION TO SEND COMMANDS TO THE AMCI SD4840E2. You can replace it with any other condition that makes the rung true, such as a push button on a HMI terminal.

EQU
Source A AMCI_Run_Preform 2
Source B 2

AMCI_SD4840E2_Jog_CW
AMCI_SD4840E2_Jog_CW AOI_CW_Jog_Move
Axis_Status_Bits buffered_AMCI_Preform[0] 17409
Programmed_Speed CW_jog_Programmed_Speed_Preform 10
Acceleration CW_jog_Acceleration_Preform 10
Deceleration CW_jog_Deceleration_Preform 10
AMCI_SD4840E2_output_data Preform_Driver:O

A Jog move that is occurring will be stopped when the Jog move command bit is reset to zero. When the SD4840E2 detects this on to off transition of the Jog move command bit, it will reduce the speed at the deceleration parameter value until motion has stopped.

When tag AMCI_Run_Preform is equal to zero, and the CW_Jog_move_occurring bit is set, reset the driver's command word to zero. Also unlatch the Add On Instructions Jog_CW_Occurring bit.

EQU
Source A AMCI_Run_Preform 2
Source B 0

AOI_CW_Jog_Move.Jog_CW_Occurring

MOV
Source 0
Dest Preform_Driver:O.COMMAND_WORD_0 2#0000_0000_1000_0000

AOI_CW_Jog_Move.Jog_CW_Occurring

Set tag AMCI_Run_Preform to 3 to start a CCW Jog Move operation.

When performing a Jog move, the Jog move command bit (bit 8 in this case) MUST remain set for motion to continue. Also, a Command Error will be generated if the target position registers are not zero, or if the velocity, acceleration, and deceleration are not valid.

Figure 5.12. Preform Stepper Motor Speed Control

The program shown in Figure 5.13 is used to capture the preform stepper motor speed. The program uses eq (5.3) to convert the frequency directly sensed by the encoder into rotational speed (in RPM).

$$RPM = \frac{(60)(Frequency)}{Line\ Count} \tag{5.3}$$

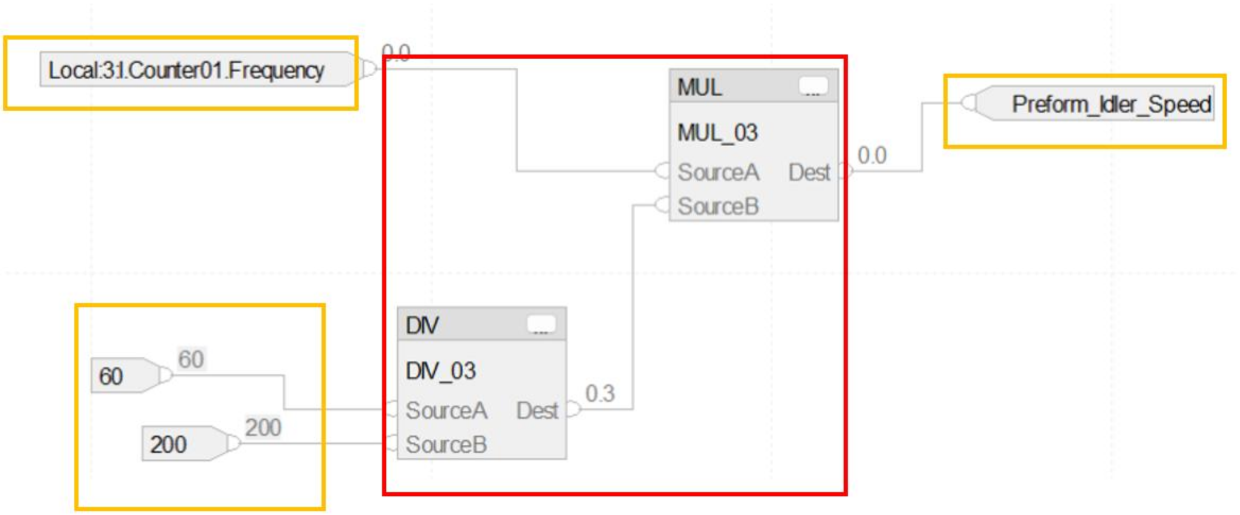


Figure 5.13. Preform Motor Encoder Speed Sensing

The PID loops (temperature, DC motor control, and fiber control) is implemented using the PIDE function in studio 5000. The PIDE functions use eq (5.1) and eq (5.2) to implement PID control. The PID control program for temperature is shown in Figure 5.14 and Figure 5.15.

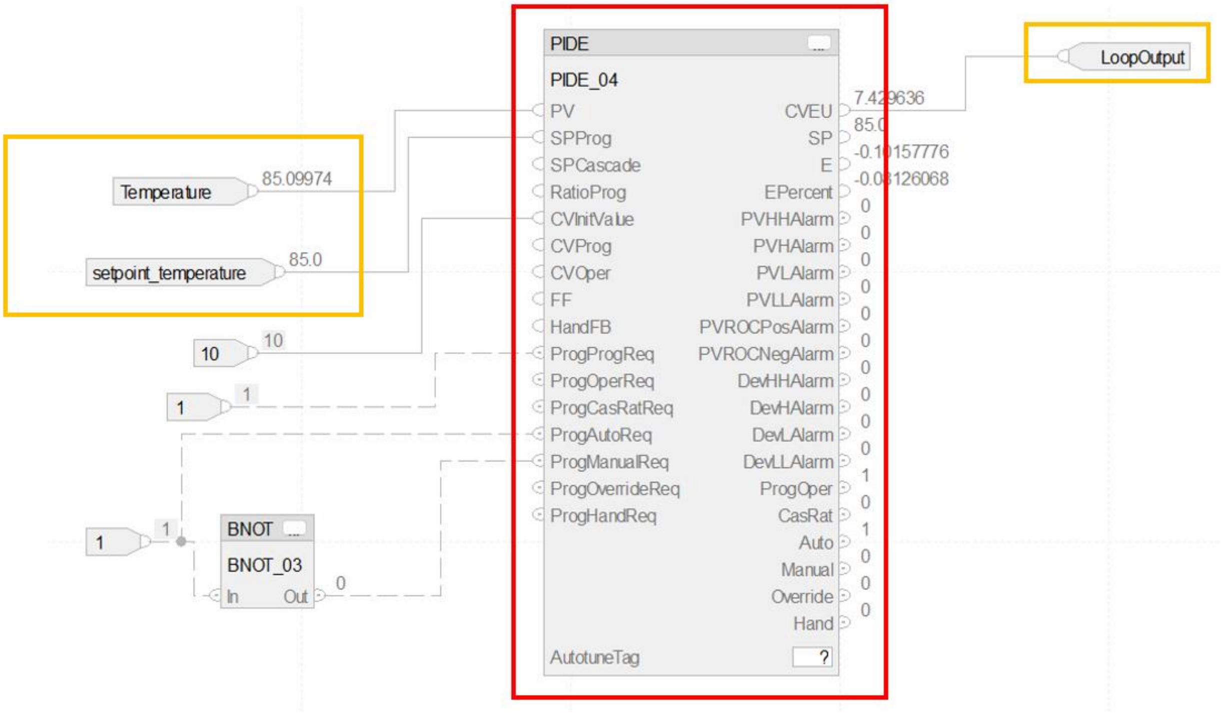


Figure 5.14. Temperature PID Loop

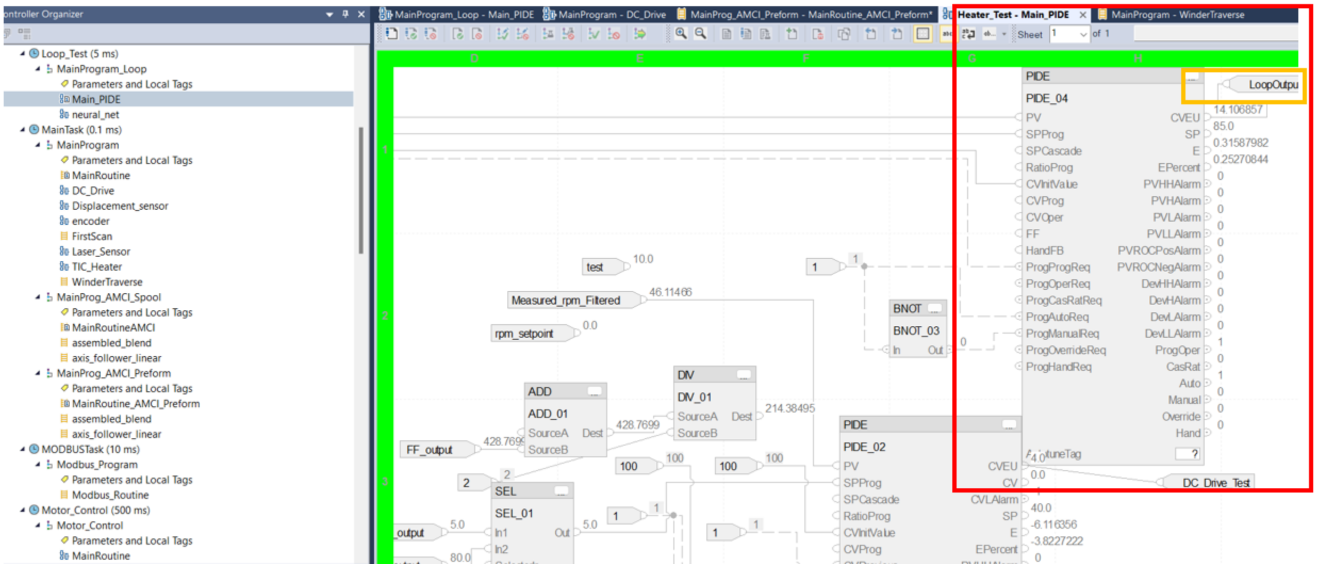


Figure 5.15. Temperature PID Loop 02

Figure XX shows the program that is used to detect temperature of the heating chamber. The sensing device is a thermistor which works on the principle that the resistance of the device changes with changing temperature. Therefore, if the device is calibrated properly, the temperature change can be detected in an electrical circuit due to the corresponding change in resistance. The thermistor used in this case, has resistance of 10k ohms at 25℃ and has B coefficient of 3950. The PLC, which is an analog system, cannot directly measure the resistance of the thermistor. Therefore, the voltage over the thermistor is measured instead. A voltage driver circuit is used by using a 68K resistor (R_0) in series with the thermistor. The resistance of the thermistor is then calculated using Eq (5.4).

$$R_T = \frac{V_{R_T} R_0}{V_{R_0}} \tag{5.4}$$

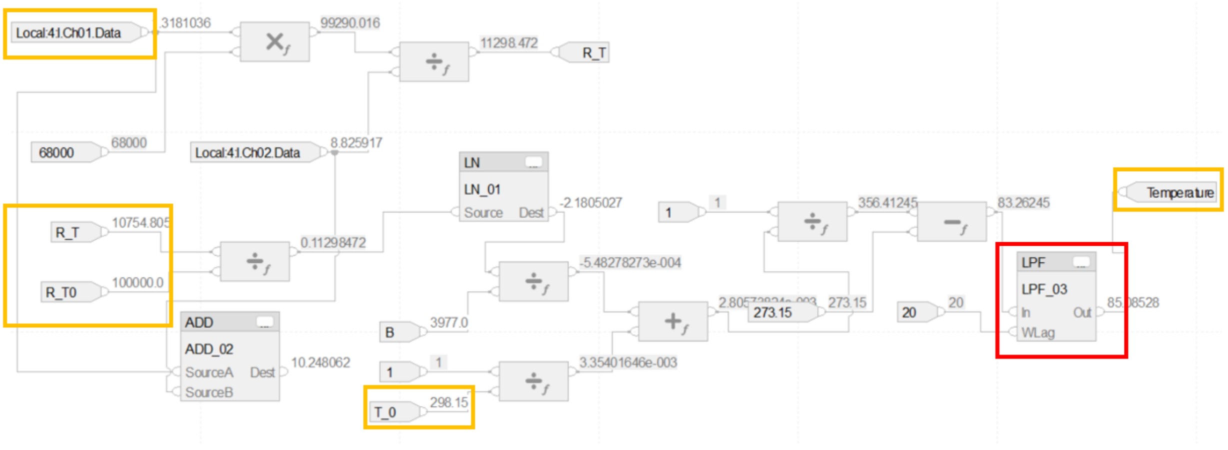


Figure 5.16. Temperature Sensing Program

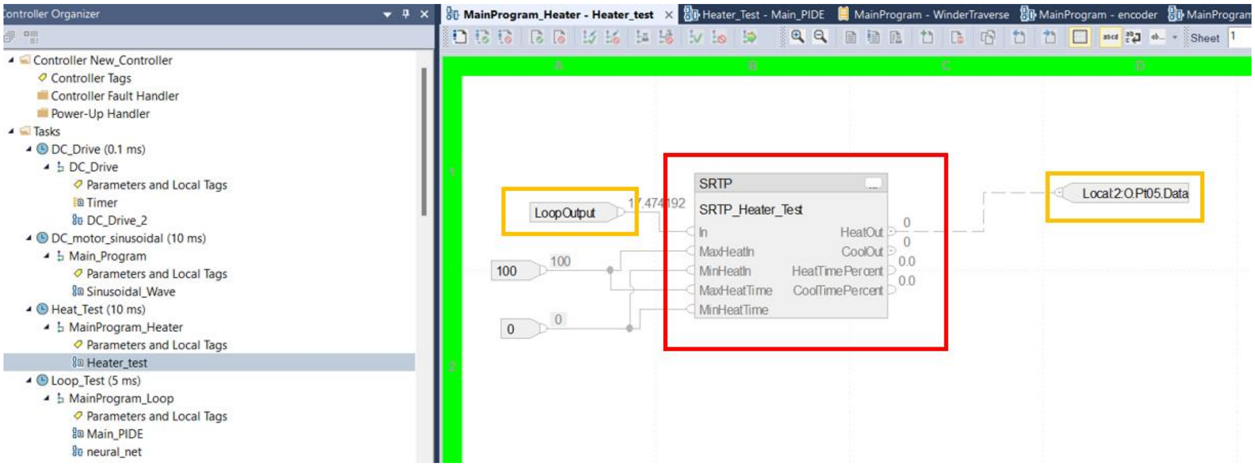


Figure 5.17. Temperature PWM/SRTP Program

5.4.2 Fiber In-Process Algorithms

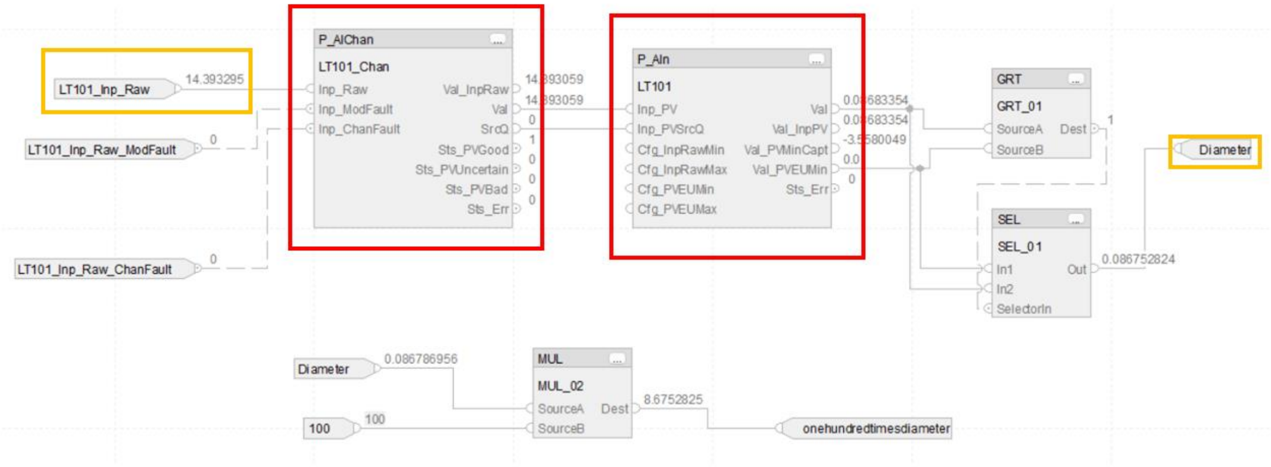


Figure 5.18. Fiber Diameter Sensing Program

5.4.3 Fiber Collection Process Algorithms

The fiber collection system has a 24V 50:1 Polulu 37D Metal Gearmotor with 64 CPR encoder, which is controlled by LGDehome high power motor controller. This motor drives the spooling capstan upon which the fiber wraps around. This pull also creates the tension needed to sustain the fiber draw process. The DC motor granularity is an interesting hardware limitation of the 5069-OB16F DC output module. The limitation of the communication frequency of this DC module limits the resolution of the duty cycle which limits the granularity of the speed change as shown in Figure 5.22. It can be seen that the duty cycle resolution of the DC motor is a function of the set cycle time. Ideally it would make sense to use the highest cycle time of 100 ms to have as much granularity as possible but these are certain disadvantages of using high cycle time (4). Firstly, a high cycle time introduces more noise in the response of the PWM controlled motor and raises the lowest speed the motor can run at (4). To find a good balance between these conflicting behaviors, a cycle time of 70 ms (1.4% duty cycle resolution) is used (4).

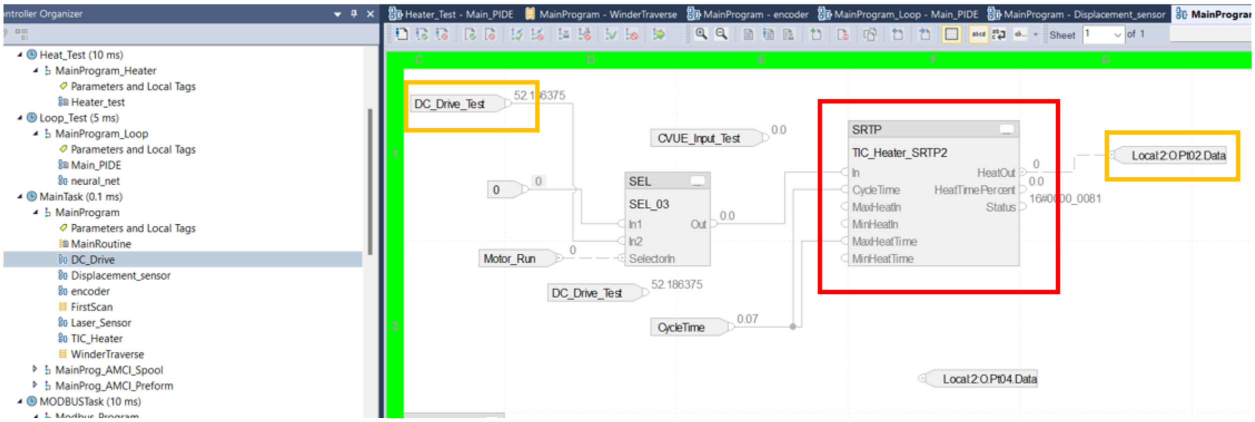


Figure 5.19. DC Spool Motor PWM/SRTP Program

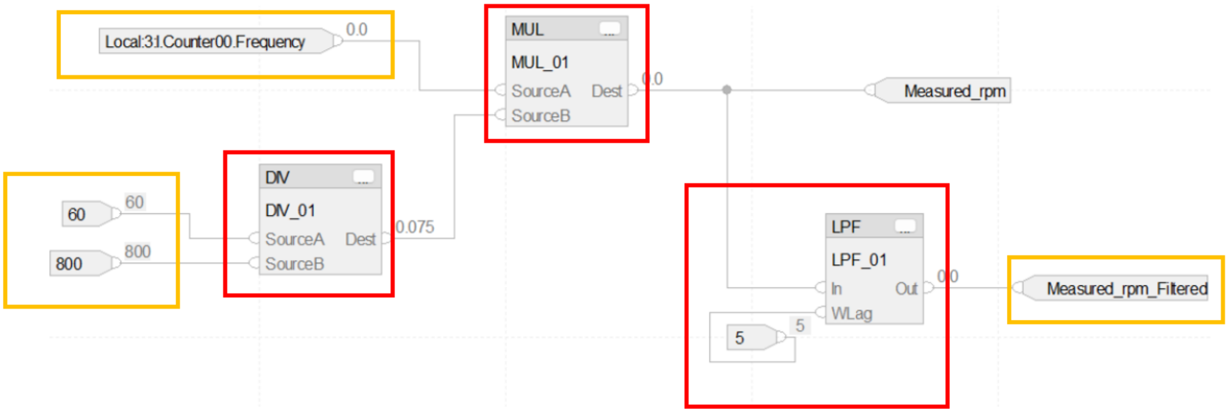


Figure 5.20. DC Spool Motor Sensing

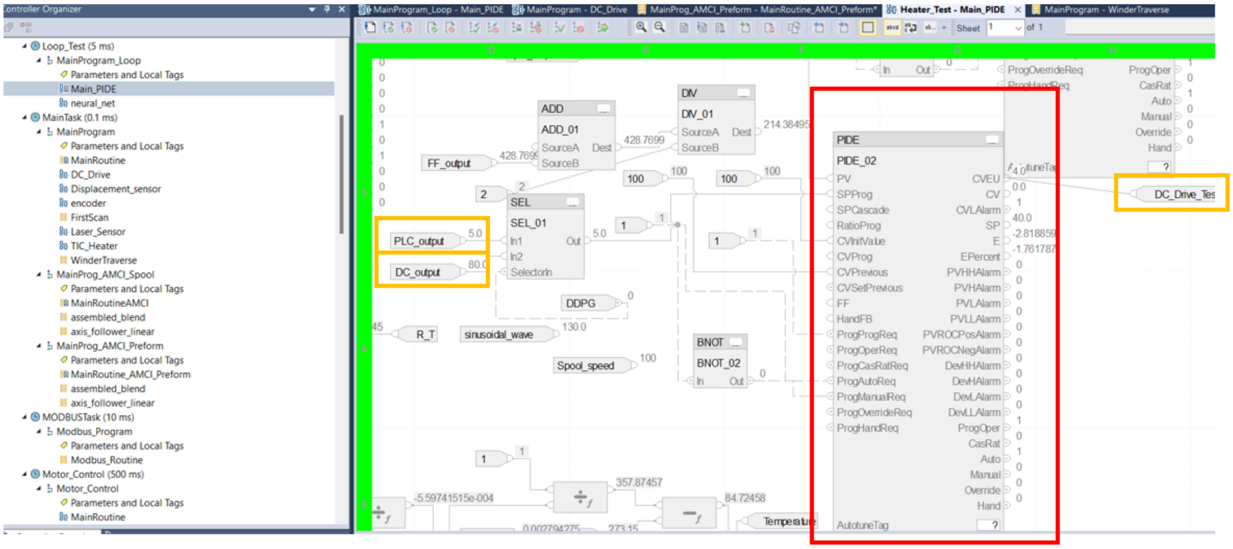


Figure 5.21. Motor PID Control

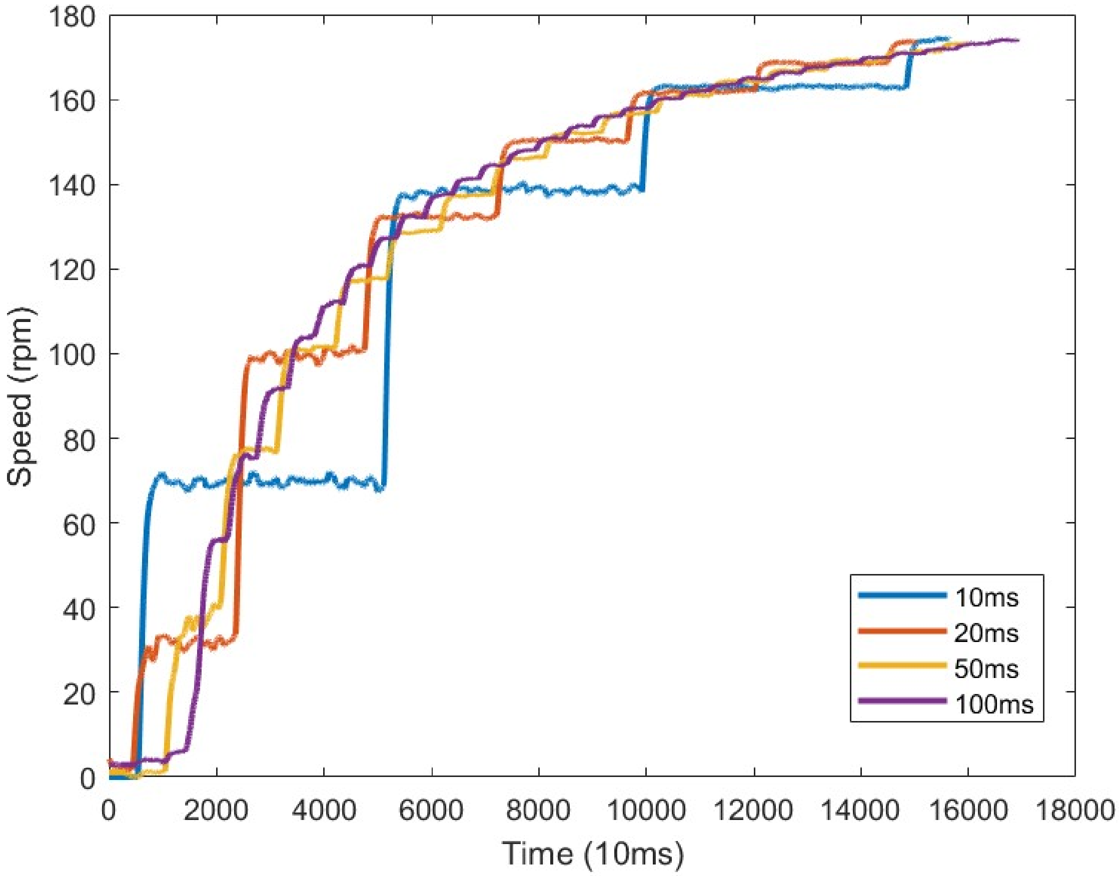
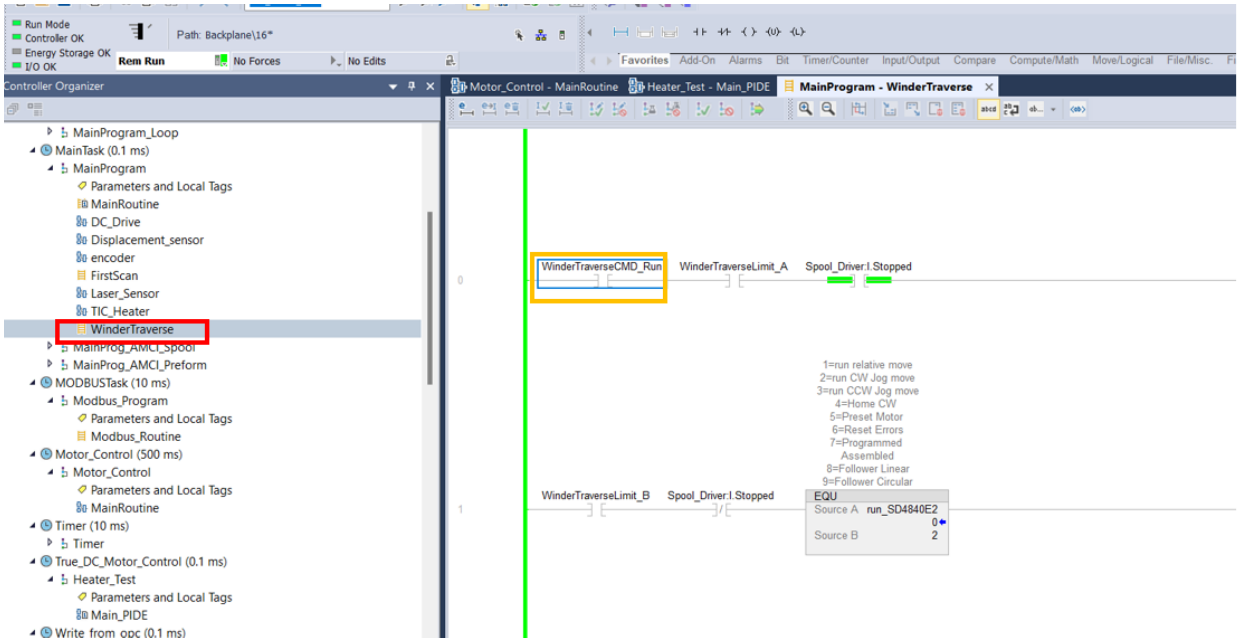


Figure 5.22. DC Spool Motor PWM Resolution (4)

In addition, a NEMA-17 Step motor driven by the second AMCI SD4840E2 Stepper Driver is used to drive the spooling platform moving back and forth (traverse motion). The “WindTraverseCMD_Run” gate is toggled to start/stop moving the spool platform back and forth.



1=run relative move
2=run CW Jog move
3=run CCW Jog move
4=Home CW
5=Preset Motor
6=Reset Errors
7=Programmed Assembled
8=Follower Linear
9=Follower Circular

Figure 5.23. Traverse Motion Algorithm

5.4.4 Fiber Control Algorithms

The fiber is controlled through a secondary PID loop which goes around the motor PID control loop as shown in Figure 5.3. The program (input, output and PIDE function) is shown in the function block program depicted in Figure 5.24. The PID tuning process for the fiber control is discussed in section 5.3.

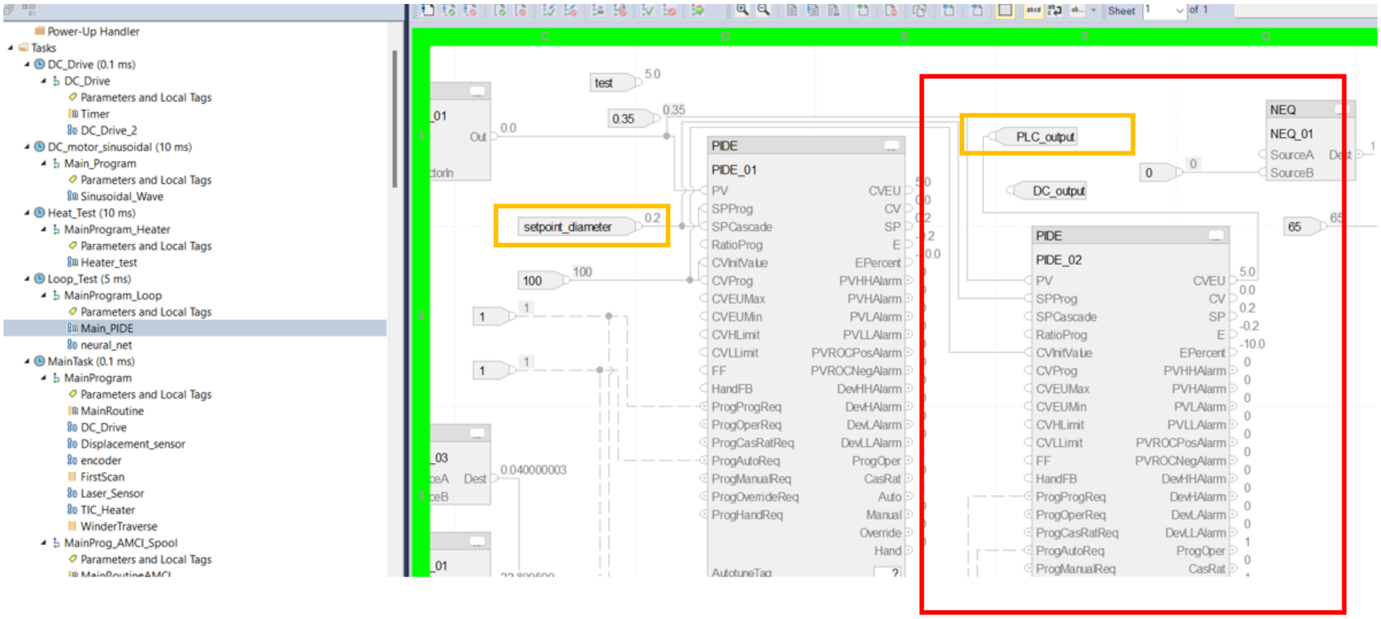


Figure 5.24. Fiber Diameter PID Control

5.4.5 Miscellaneous Process Algorithms

There are a couple of additional key programs that maybe worth documenting. Figure 5.25 shows the sinusoidal speed program that is used to generate sinusoidal speed input that is fed into the DC spool motor. This is needed to generate sinusoidal patterned setpoint for the motor PID control and obtain associated fiber diameter (in an open loop control). This dataset is important as it contains a lot of system dynamics properties that are then used to train the black box learned control algorithms (discussed in section 6.4).

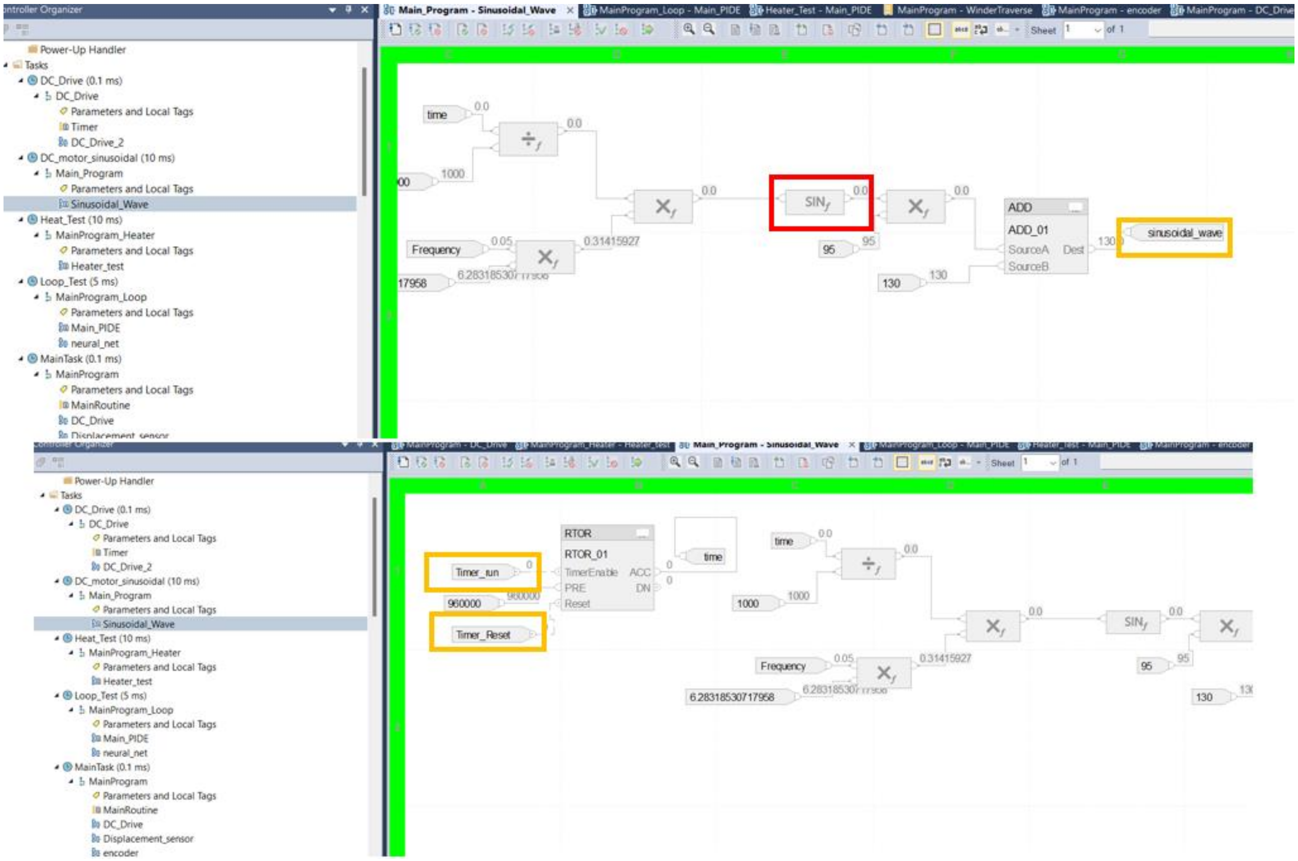
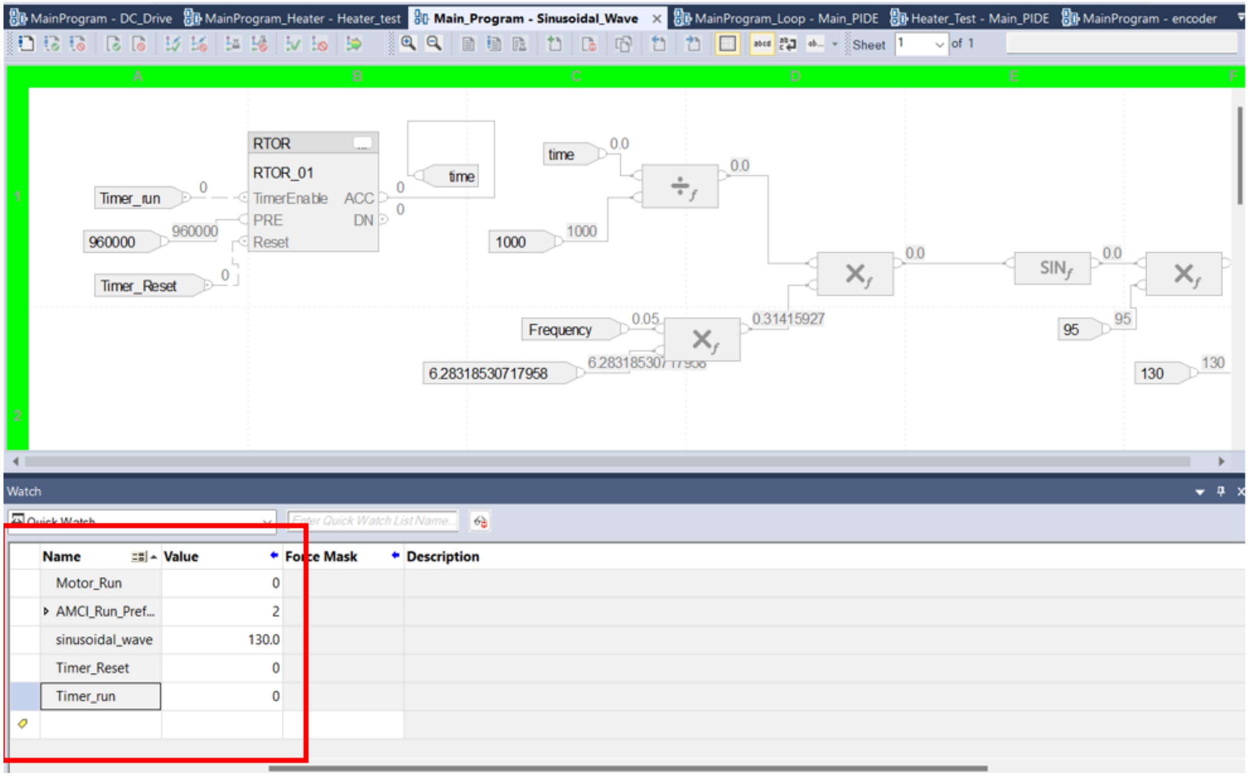


Figure 5.25. Sinusoidal Input for DC Motor Speed Control

Figure 5.26 shows the various tags associated with the sinusoidal speed program. The "frequency" parameter is set to the desired values and the "Motor_Run" is set to 1. The total time the sinusoidal program is desired to be run is set. Here it is set to 960000 ms or 16 mins. This turns on the spool DC motor. Then the "AMCI_Run_Preform" is set to 2 (to get the preform stepper motor to turn counterclockwise and push the preform material into the heaing chamber). Note: the traverse mechanism must also be turned on at this point to ensure even spooling of the fiber on the capstan (see Figure 5.23). Once the PLC FrED starts to spool in a stable manner, the "Timer_run" variable is set to 1 to start the sinusoidal speed program. Data must be collected using the proper trend (see Figure 5.27). After 16 mins, the system will automatically exit the sinusoidal program and run at a constant speed, so it important to step the data collection at around 15 mins or so. After 16 mins, to start another session, the "Timer_run" must be set to 0, then the "Timer_reset" must be set to 1. Then after waiting for a second or so the "Timer_reset" can be set to 0. Then from this point, when the "Timer_run" is set to 1, the sinusoidal program will start again, and another data set can be collected.



Name	Value	Force Mask	Description
Motor_Run	0		
AMCI_Run_Pref...	2		
sinusoidal_wave	130.0		
Timer_Reset	0		
Timer_run	0		

Figure 5.26. Import Tags for Sinusoidal Input for DC Motor Speed Control

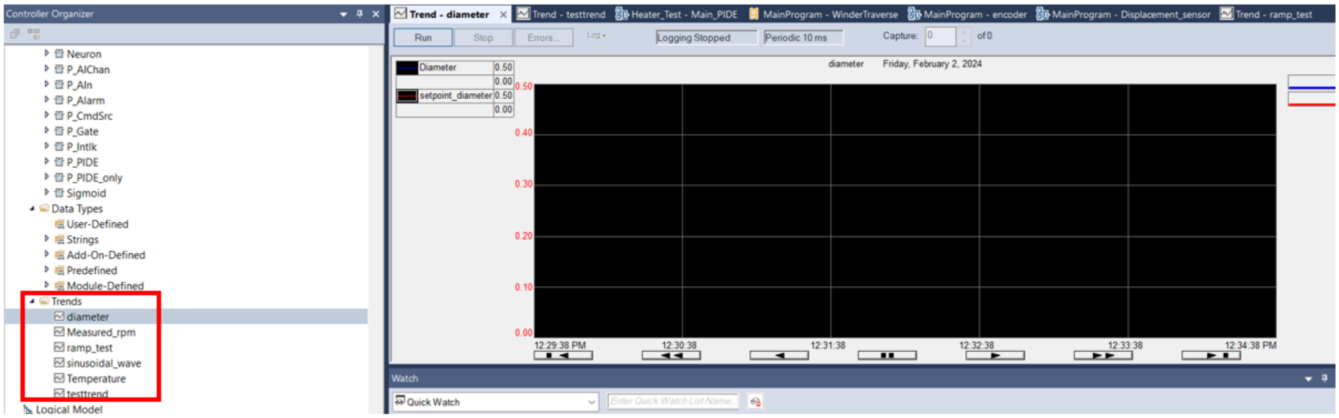


Figure 5.27. Various Custom Trends for Data Collection

5.5 OPC Client and Factory Talk Communication

The OPC Client and Factory Talk Communication forms the basis of establishing the structure needed to show that learned models can be used on PLCs to control manufacturing hardware. All the various subsystems from this thesis (mechanical design, learned model controller development, electrical hardware design, etc) come together to build the platform which could be tested. However, this communication methodology is the novel contribution which showed that learned models, while being on a Windows based workstation can effectively communicate with PLCs and a manufacturing platform (FrED). Having said that, my contributions to the develop the communication architecture is limited. That credit goes to the work done by Zhang (4). The discussion of the communication architecture work in this section serves as a reference to establish context to the original work discussed (done by me) in the other sections.

5.5.1 OPC UA Background

The Open Platform Communication Unified Architecture (OPC UA) is an open-source IEC62541 standard for data exchange between PLC, Linux, Windows, and various industrial devices that has been developed by the OPC Foundation. OPC Foundation was the first to standardize communication between various platforms such as industrial devices, PLC and Windows/Linux based client/servers. The standard is based on a client/server architecture, where the data packets are modeled as objects in the server and are presented to the clients as service. This communication architecture can be depicted in Figure 5.28.

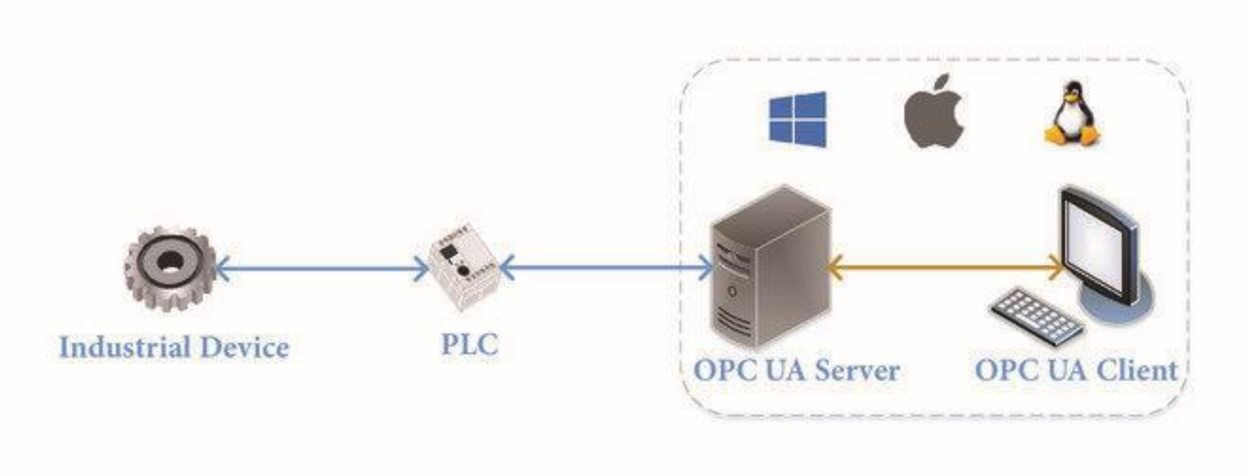


Figure 5.28. General OPCUA Client-Server Communication (4)

OPC utilizes both Unified Architecture Transmission Control Protocol (UA TCP) and Simple Object Access Protocol (SOAP)/Hypertext Transfer Protocol (HTTP) for communicating between OPC server and OPC client. OPC communication standards COM/DCOM (Distributed Component Model) for data exchange were based off Windows/Microsoft. With the need for cross-platform industrial automation, OPC UA standards were developed in 2008 to allow for cross-platform communication.

5.5.2 OPC UA Sever Configuration.

Even through OPCUA establishes the standard for communication between the PLC, FrED and the Windows based client (which in this case is the Windows based workstation), additional software is needed to configure act as the server. Factory Talk Linx is the data server that performs the communication service that allows the workstation to talk to FrED. Factory Talk Linx Gateway allows third party software (such as Excel or Studio 5000) to access/write the data to Factory Linx (4). The high-level summary of data communication/Factory Link ordering process is highlighted in Figure 5.29. Figure 5.30 and Figure 5.31 show the FactoryTalk Linx Administration console and Server configuration in FactoryTalk LinX Gateway respectively (4).

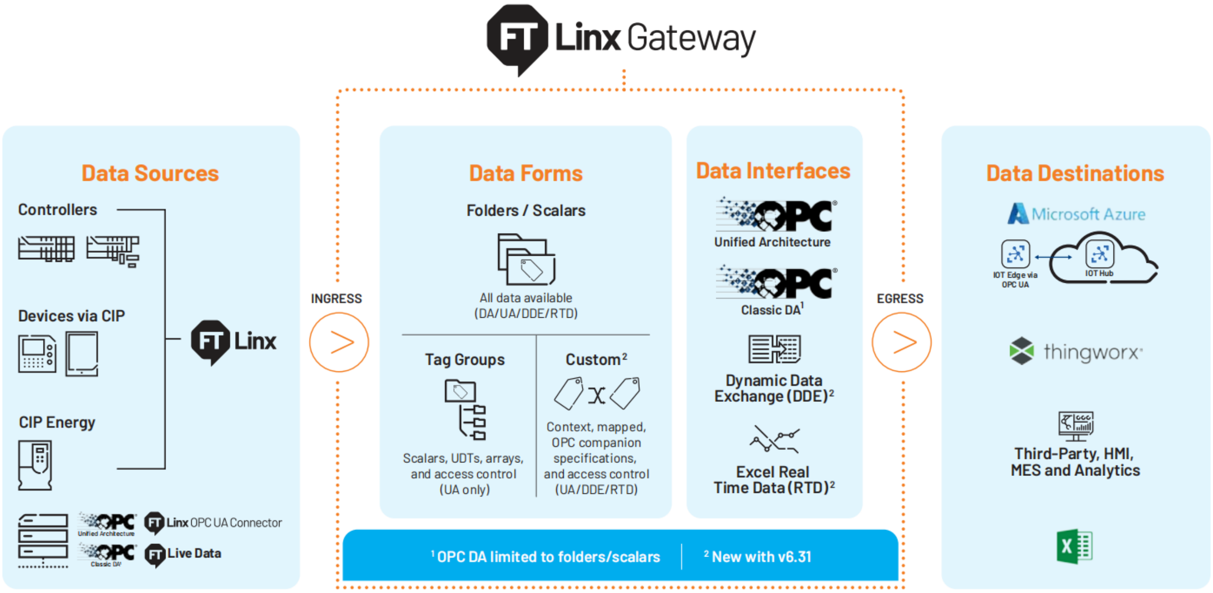
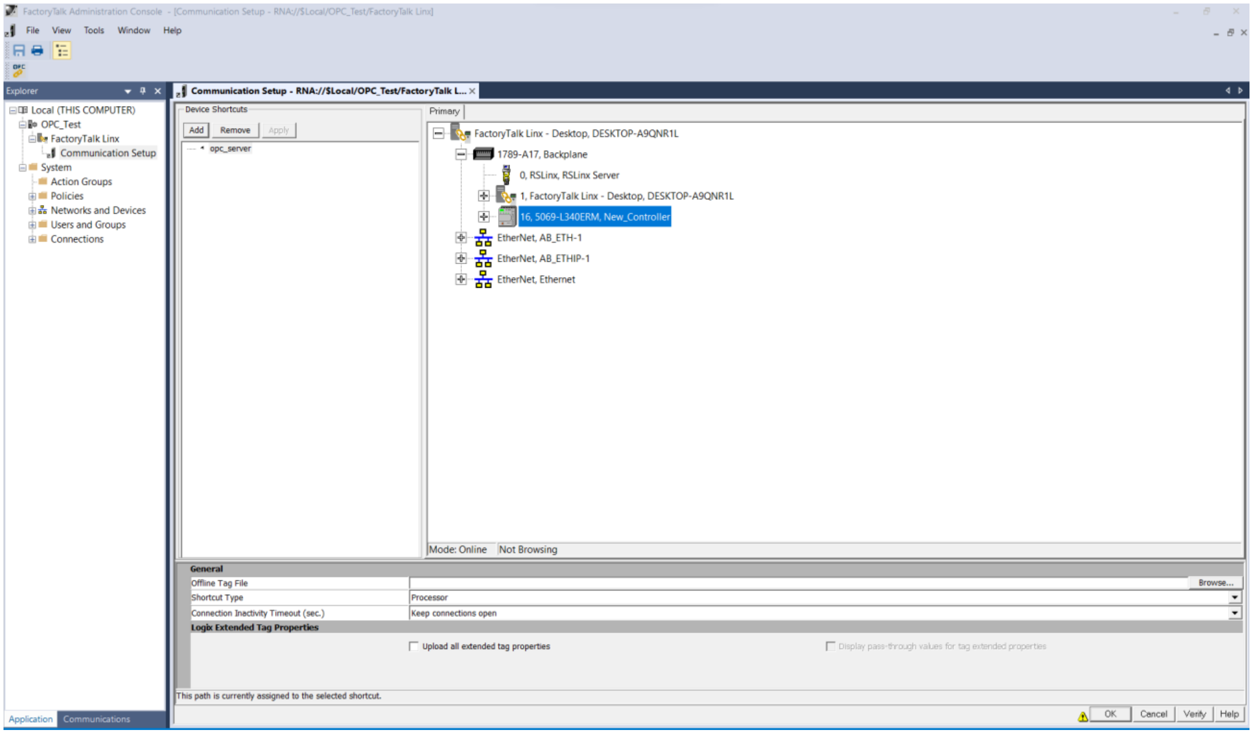


Figure 5.29. Data communication via FactoryTalk Linx ordering (4)



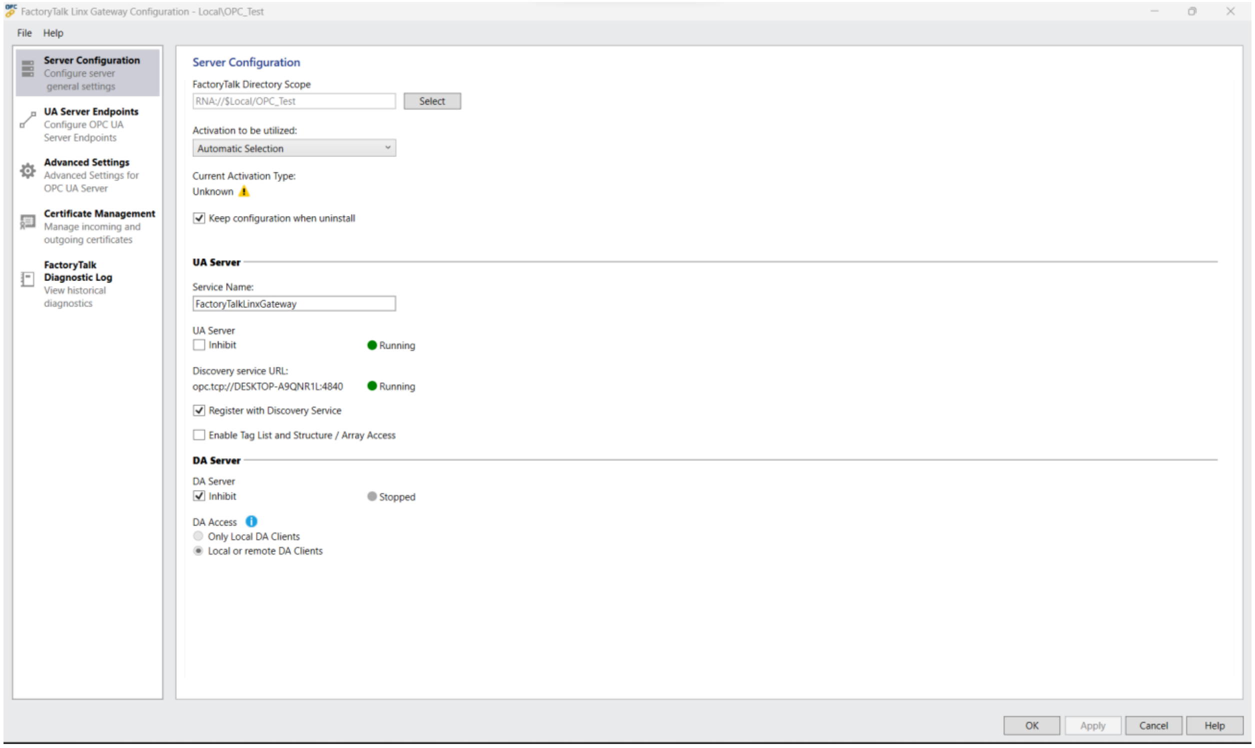
Offline Tag File	
Shortcut Type	Processor
Connection Inactivity Timeout (sec.)	Keep connections open

□ Upload all extended tag properties

□ Display pass-through values for tag extended properties

This path is currently assigned to the selected shortcut.

Figure 5.30. FactoryTalk Linx Administration Console



FactoryTalk Linx Gateway Configuration - Local\OPC_Test
File
Help
Server Configuration
Configure server general settings
UA Server Endpoints
Configure OPC UA Server Endpoints
Advanced Settings
Advanced Settings for OPC UA Server
Certificate Management
Manage incoming and outgoing certificates
FactoryTalk Diagnostic Log
View historical diagnostics
Select
OK
Apply
Cancel
Help

Server Configuration

FactoryTalk Directory Scope: RNA://$Local/OPC_Test

Activation to be utilized: Automatic Selection

Current Activation Type: Unknown

☑ Keep configuration when uninstall

UA Server

Service Name: FactoryTalkLinxGateway

UA Server
☐ Inhibit ● Running

Discovery service URL:
opc.tcp://DESKTOP-A9QNR1L:4840 ● Running

☑ Register with Discovery Service

☐ Enable Tag List and Structure / Array Access

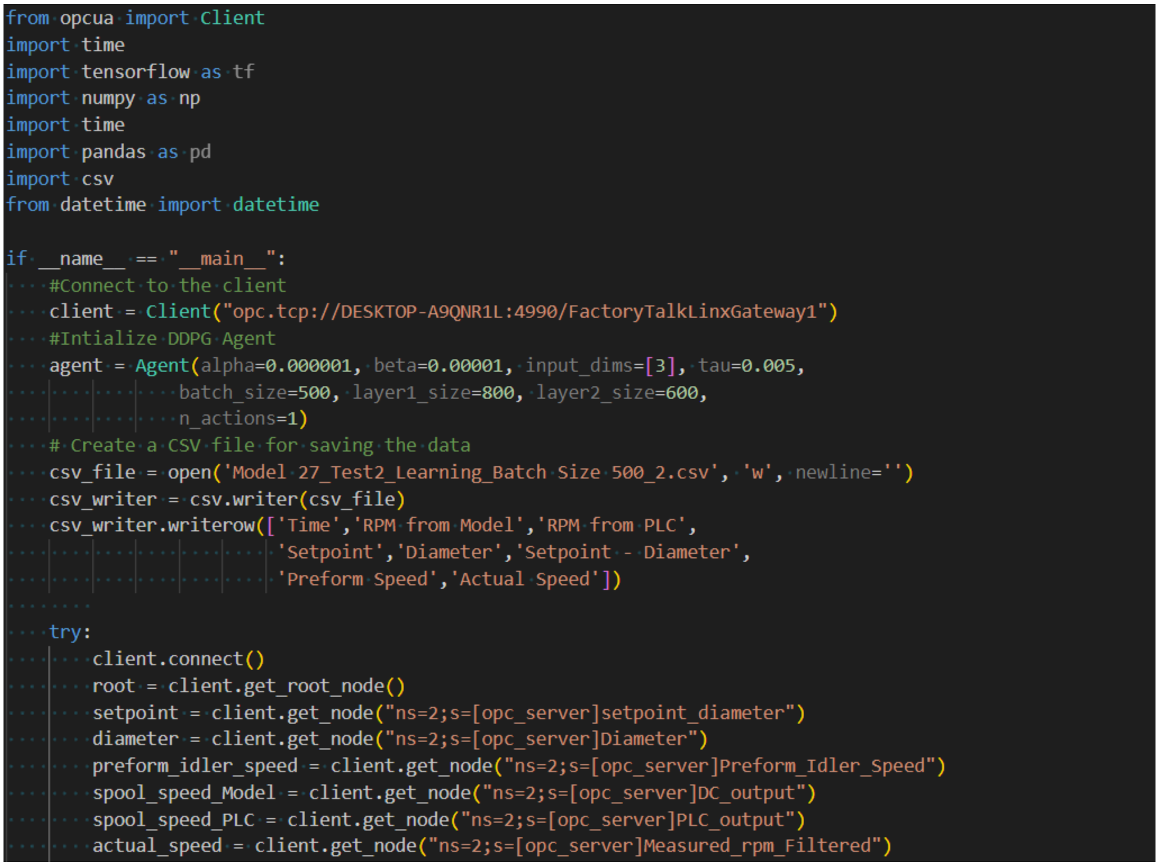
DA Server

DA Server
☑ Inhibit ● Stopped

DA Access
○ Only Local DA Clients
● Local or remote DA Clients

Figure 5.31. Server Configuration in FactoryTalk Linx Gateway

The first step in communicating with OPCUA communication framework is utilizing the open-source python package named “opcua”. Then the client class (as shown in Figure 5.32) needs to be initialized inside of a variable with the OPC UA server address. The *connect()* is then used to initiate communication session with the server. This function also performs the necessary handshakes, establishing present and future sessions for all interactions. The *get_root_node()* function serves the purpose of retrieving the root node within the server's address space. This root node represents the pinnacle of the hierarchy, providing access to all other nodes contained within the address space. On the other hand, the *get_node()* function is employed to retrieve a specific node within the OPC UA server's address space. These nodes typically store information related to various parameters that define the state of the FrED system. The snippet of code that performs this function is shown in Figure 5.32.



```
from opcua import Client
import time
import tensorflow as tf
import numpy as np
import time
import pandas as pd
import csv
from datetime import datetime

if __name__ == "__main__":
    #Connect to the client
    client = Client("opc.tcp://DESKTOP-A9QNR1L:4990/FactoryTalkLinxGateway1")
    #Intialize DDPG Agent
    agent = Agent(alpha=0.000001, beta=0.00001, input_dims=[3], tau=0.005,
                batch_size=500, layer1_size=800, layer2_size=600,
                n_actions=1)
    # Create a CSV file for saving the data
    csv_file = open('Model 27_Test2_Learning_Batch Size 500_2.csv', 'w', newline='')
    csv_writer = csv.writer(csv_file)
    csv_writer.writerow(['Time','RPM from Model','RPM from PLC',
                         'Setpoint','Diameter','Setpoint - Diameter',
                         'Preform Speed','Actual Speed'])
        
    try:
        client.connect()
        root = client.get_root_node()
        setpoint = client.get_node("ns=2;s=[opc_server]setpoint_diameter")
        diameter = client.get_node("ns=2;s=[opc_server]Diameter")
        preform_idler_speed = client.get_node("ns=2;s=[opc_server]Preform_Idler_Speed")
        spool_speed_Model = client.get_node("ns=2;s=[opc_server]DC_output")
        spool_speed_PLC = client.get_node("ns=2;s=[opc_server]PLC_output")
        actual_speed = client.get_node("ns=2;s=[opc_server]Measured_rpm_Filtered")
```

Figure 5.32. OPC UA Client Class Instantiation (4)

5.5.3 OPC UA Communication.

After successful configuration, and after the Fred system goes online, data from the sensors update their respective nodes on the OPC UA server. Another function in the client class named *get_value()* is utilized is utilized by the learned controller to store the observed states (parameters of the FrED environment) on a stack. These states then get fed into the learned control model to output the optimal speed of the spool motor according to the desired setpoint diameter. These parameter values are read in from tags in Studio 5000 which store the physical values representing the FrED system (like motor speed, temperature, fiber diameter, etc), the associated timestamp of when these parameters were read and the node identification number which uniquely identifies the node associated with each of the parameters. These parameters are depicted in Figure 5.34. This optimal speed is then fed into OPCUA server which uses the *"set_value()"* function to set new value to the tag in Studio 5000 which in turn controls the actual speed of the spool DC motor. This associated class code and the pictorial view of the movement of data is shown in Figure 5.35 and Figure 5.33 respectively. One of the key concerns regarding communication is the lag between the OPC server and client. If the lag time is on the order of the time needed for the learned controller to process the data, then there maybe some system performance lags that will lead to inaccurate results. Zhang showed that time lag due to subscription-based communication is on the order of 522 ms (4). This is depicted in Appendix A. The time needed to collect data for the learned model is around 355 ms. To ensure minimal disruption, it is suggested to tune the training time to match the communication lag time to ensure model behavior matches communication time and therefore no lag affects the physical behavior of the FrED system.

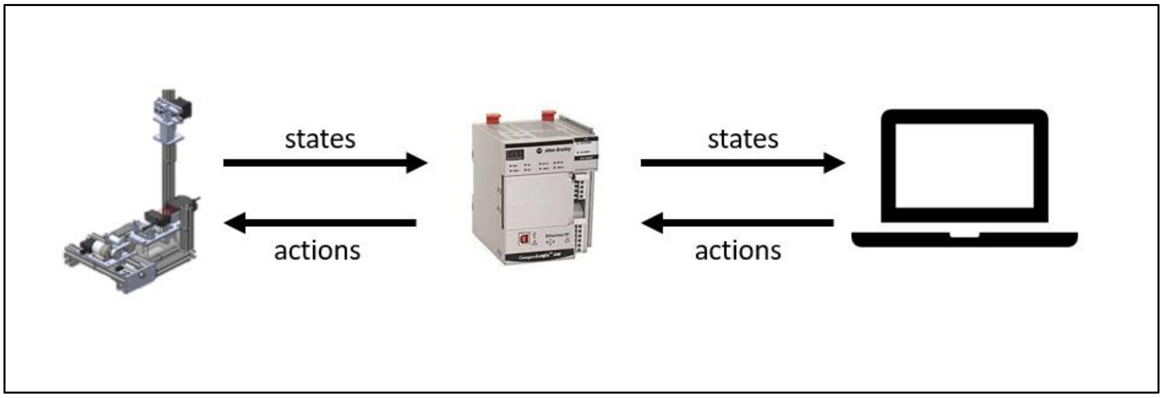
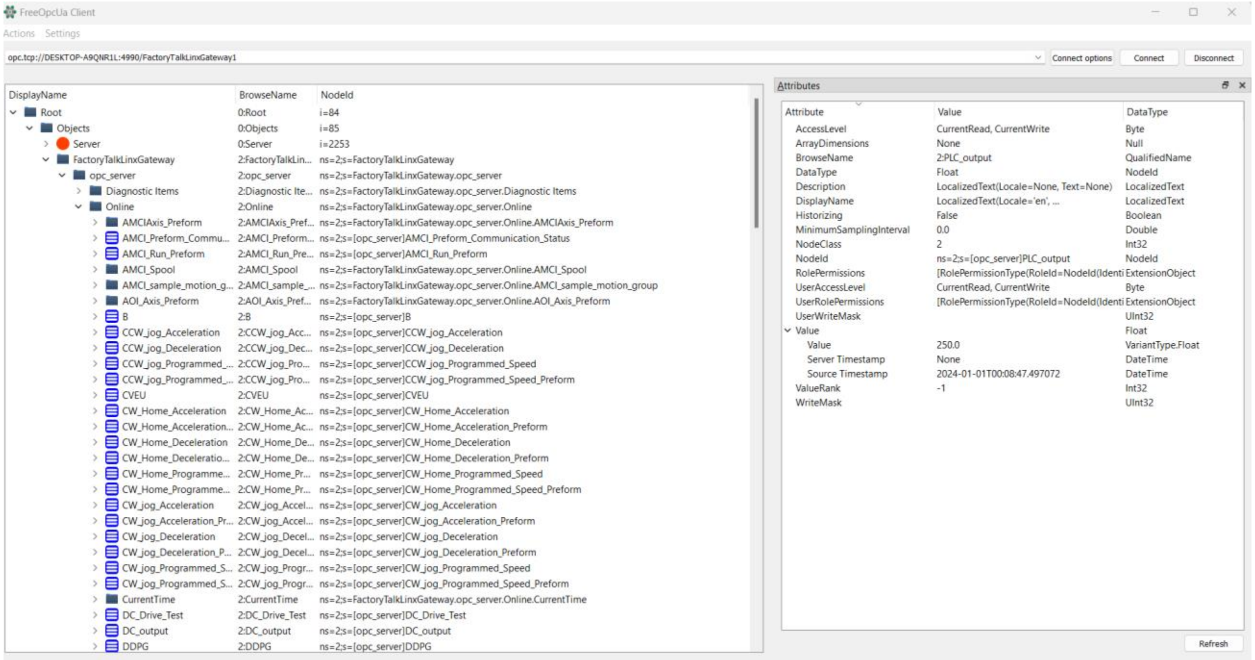


Figure 5.33. Data Transfer Methodology in PLC/FrED Frameworks (5)



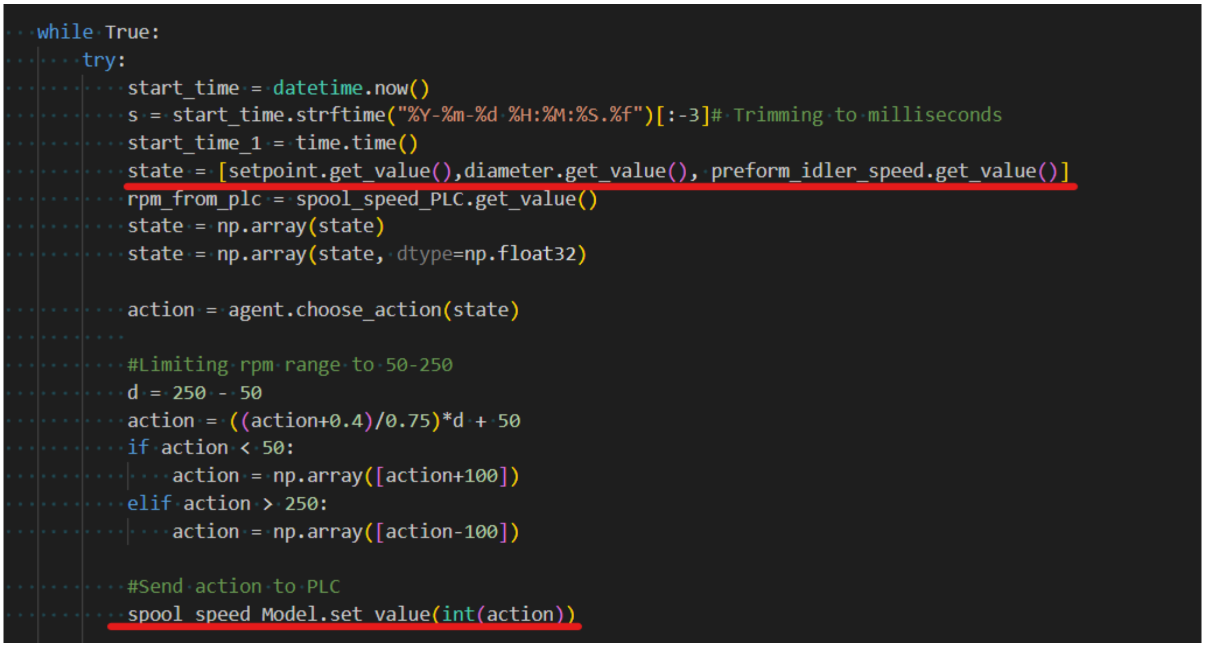
opc.tcp://DESKTOP-A9QNR1L:4990/FactoryTalkLinxGateway1

DisplayName	BrowseName	NodeId
Root	0:Root	i=84
Objects	0:Objects	i=85
Server	0:Server	i=2253
FactoryTalkLinxGateway	2:FactoryTalkLin...	ns=2;s=FactoryTalkLinxGateway
opc_server	2:opc_server	ns=2;s=FactoryTalkLinxGateway.opc_server
Diagnostic Items	2:Diagnostic Ite...	ns=2;s=FactoryTalkLinxGateway.opc_server.Diagnostic Items
Online	2:Online	ns=2;s=FactoryTalkLinxGateway.opc_server.Online
AMCIAxis_Preform	2:AMCIAxis_Pref...	ns=2;s=FactoryTalkLinxGateway.opc_server.Online.AMCIAxis_Preform
AMCI_Preform_Commu...	2:AMCI_Preform...	ns=2;s=[opc_server]AMCI_Preform_Communication_Status
AMCI_Run_Preform	2:AMCI_Run_Pre...	ns=2;s=[opc_server]AMCI_Run_Preform
AMCI_Spool	2:AMCI_Spool	ns=2;s=FactoryTalkLinxGateway.opc_server.Online.AMCI_Spool
AMCI_sample_motion_g...	2:AMCI_sample_...	ns=2;s=FactoryTalkLinxGateway.opc_server.Online.AMCI_sample_motion_group
AOI_Axis_Preform	2:AOI_Axis_Pref...	ns=2;s=FactoryTalkLinxGateway.opc_server.Online.AOI_Axis_Preform
B	2:B	ns=2;s=[opc_server]B
CCW_jog_Acceleration	2:CCW_jog_Acc...	ns=2;s=[opc_server]CCW_jog_Acceleration
CCW_jog_Deceleration	2:CCW_jog_Dec...	ns=2;s=[opc_server]CCW_jog_Deceleration
CCW_jog_Programmed_...	2:CCW_jog_Pro...	ns=2;s=[opc_server]CCW_jog_Programmed_Speed
CCW_jog_Programmed_...	2:CCW_jog_Pro...	ns=2;s=[opc_server]CCW_jog_Programmed_Speed_Preform
CVEU	2:CVEU	ns=2;s=[opc_server]CVEU
CW_Home_Acceleration	2:CW_Home_Ac...	ns=2;s=[opc_server]CW_Home_Acceleration
CW_Home_Acceleration...	2:CW_Home_Ac...	ns=2;s=[opc_server]CW_Home_Acceleration_Preform
CW_Home_Deceleration	2:CW_Home_De...	ns=2;s=[opc_server]CW_Home_Deceleration
CW_Home_Deceleratio...	2:CW_Home_De...	ns=2;s=[opc_server]CW_Home_Deceleration_Preform
CW_Home_Programme...	2:CW_Home_Pr...	ns=2;s=[opc_server]CW_Home_Programmed_Speed
CW_Home_Programme...	2:CW_Home_Pr...	ns=2;s=[opc_server]CW_Home_Programmed_Speed_Preform
CW_jog_Acceleration	2:CW_jog_Accel...	ns=2;s=[opc_server]CW_jog_Acceleration
CW_jog_Acceleration_Pr...	2:CW_jog_Accel...	ns=2;s=[opc_server]CW_jog_Acceleration_Preform
CW_jog_Deceleration	2:CW_jog_Decel...	ns=2;s=[opc_server]CW_jog_Deceleration
CW_jog_Deceleration_P...	2:CW_jog_Decel...	ns=2;s=[opc_server]CW_jog_Deceleration_Preform
CW_jog_Programmed_S...	2:CW_jog_Progr...	ns=2;s=[opc_server]CW_jog_Programmed_Speed
CW_jog_Programmed_S...	2:CW_jog_Progr...	ns=2;s=[opc_server]CW_jog_Programmed_Speed_Preform
CurrentTime	2:CurrentTime	ns=2;s=FactoryTalkLinxGateway.opc_server.Online.CurrentTime
DC_Drive_Test	2:DC_Drive_Test	ns=2;s=[opc_server]DC_Drive_Test
DC_output	2:DC_output	ns=2;s=[opc_server]DC_output
DDPG	2:DDPG	ns=2;s=[opc_server]DDPG

Attributes

Attribute	Value	DataType
AccessLevel	CurrentRead, CurrentWrite	Byte
ArrayDimensions	None	Null
BrowseName	2:PLC_output	QualifiedName
DataType	Float	NodeId
Description	LocalizedText(Locale=None, Text=None)	LocalizedText
DisplayName	LocalizedText(Locale='en', ...	LocalizedText
Historizing	False	Boolean
MinimumSamplingInterval	0.0	Double
NodeClass	2	Int32
NodeId	ns=2;s=[opc_server]PLC_output	NodeId
RolePermissions	[RolePermissionType(RoleId=NodeId(Identi	ExtensionObject
UserAccessLevel	CurrentRead, CurrentWrite	Byte
UserRolePermissions	[RolePermissionType(RoleId=NodeId(Identi	ExtensionObject
UserWriteMask		UInt32
Value		Float
Value	250.0	VariantType.Float
Server Timestamp	None	DateTime
Source Timestamp	2024-01-01T00:08:47.497072	DateTime
ValueRank	-1	Int32
WriteMask		UInt32

Figure 5.34. Online Tags read by OPCUA Client (4)



```
    while True:
        try:
            start_time = datetime.now()
            s = start_time.strftime("%Y-%m-%d %H:%M:%S.%f")[:-3]# Trimming to milliseconds
            start_time_1 = time.time()
            state = [setpoint.get_value(),diameter.get_value(), preform_idler_speed.get_value()]
            rpm_from_plc = spool_speed_PLC.get_value()
            state = np.array(state)
            state = np.array(state, dtype=np.float32)

            action = agent.choose_action(state)

            #Limiting rpm range to 50-250
            d = 250 - 50
            action = ((action+0.4)/0.75)*d + 50
            if action < 50:
                action = np.array([action+100])
            elif action > 250:
                action = np.array([action-100])

            #Send action to PLC
            spool_speed_Model.set_value(int(action))
```

Figure 5.35. Code Depicting Information transfer between learned model and OPCUA Server (5)

This page is intentionally left blank.

Chapter 6 Learned Controller Implementation

6.1 Learned Controller Motivations

While the PID based controllers perform very efficiently for one set of gain values, often in the real-world optical fiber manufacturing the environment, system and the input/output parameters often change. This maybe due to new diameter requirements, tension requirements, etc. The act of re-tuning through a trial-and-error process is very costly due to skilled technician labor needed and controller downtime needed. The act of tuning a PID controller also involves a lot of experimentation, as it involves the technician to adjust certain parameters (gains of the PID controller), observe the response of the controller and adjust until satisfactory performance is achieved. As a result of these inefficiencies, an alternative controller algorithm is an adaptive black-box model controller that can learn from its environment and the changing input parameters and modify the output parameters of the system as needed to achieve the desired result. This type of controller would not need any manual re-tuning as it would autonomously learn from the input/output parameters of the system and retrain the black-box model as needed to progressively approach the desired output. Due to the complicated nature of implementing such an adaptive controller, the task is broken down into major milestones/phases. While not all these phases were achieved in this research due to time and resource constraints, some of these phases were implemented on the test platform (FrED system + Rockwell Allen Bradley PLC). The phases are described as follows:

Phase 1: In this phase an optimized one-time gain setting of the PID controller is implemented on the training platform. Increase the range and reduce the variation of the fiber diameter as much as possible to ensure a wide range of data is available to evaluate the performance of the PID and any black-box model controllers built on top of that.

Phase 2: In this phase, a black box model controller is trained to perform optimally in a single environment. Usually the training is performed offline (not on the FrED/PLC environment), to take advantage of superior computing resources and limit random performances especially during the early phases of training. After the training, the black box model controller can optimally perform for changing input parameters of the system while the environment remains unchanged. This performance is ideally better than the PID controller in terms of setpoint change response time, reduced variation, etc.

Phase 3: In this phase, a black box model controller is sufficiently intelligent and adaptive enough where the training of the model can occur online (on the FReD/PLC environment). This phase relaxes the constraint of the environment being constant. Therefore, the black box model controller can fully adapt to both changing input parameters to the system and the environment within which this system functions. For instance, for the FrED system specifically, this controller would be able to perform optimally when both the input parameters (such as fiber diameter setpoint) and environment (heater temperature, fiber tension, spool traverse speed, external perturbations, etc) are changing.

This research implements phase 1 completely and phase 2 partially. That is, through this research, an optimal PID has been implemented (see section 5.3) and black box model controller has been implemented (see section 6.2-6.7). However, the black box model controller that has been implemented in this paper is not optimal relative to the PID controller due to limited time and data availability.

6.2 Training and Testing Dataset Collection Method

Various types of training and testing data were collected to train the black box model. The governing logic behind the types of data collected is that it is desirable to capture in the model the range of open loop dynamical behavior the FrED system. This contrasts with capturing specific static behaviors of the system where not a lot of information is learned for other types of static behaviors. For instance, to learn the dynamic behavior of the system, the open loop speed of the spool DC motor was varied according to chirp (6.2.2) and sinusoidal function (6.2.3). Note: by open loop it means no external PID loop exists around the diameter setpoint, but an internal PID loop does exist around the spool DC motor speed input (Refer to Figure 6.3 in section 6.2.2 for more details).

6.2.1 Random Step

In this data collection process, the random step function input of the motor speed is varied according to randomly changing the spool DC motor setpoints. This type of data is desirable to test the static performance of the black-box model controller. Both the spool motor response and the diameter response of the FrED system to the random step spool motor speed input are shown in Figure 6.1 and Figure 6.2.

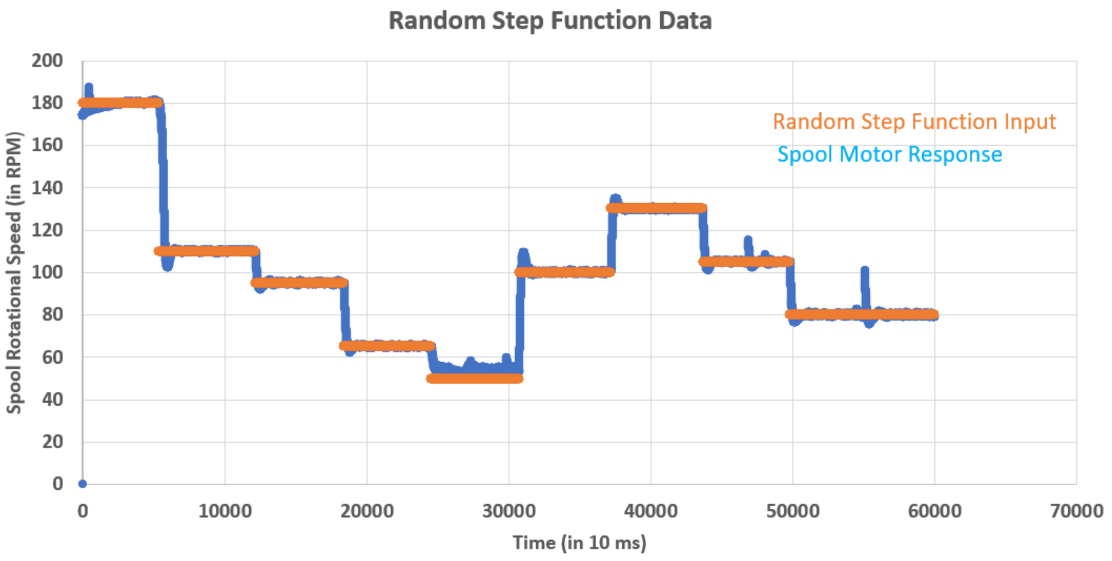


Figure 6.1. Random Step Speed Function Input (for DC Spool Motor).

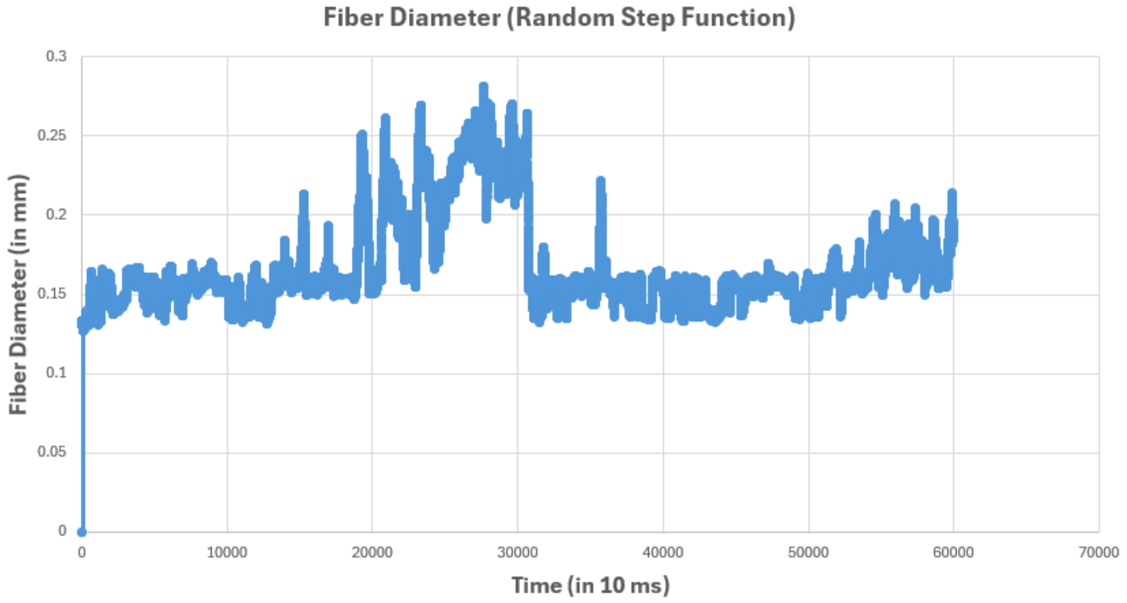


Figure 6.2. Diameter resulting from Random Step Speed Function Input (for DC Spool Motor).

6.2.2 Chirp

In this data collection process, the chirp function input of the motor speed is varied according to sinusoidal/decreasing amplitude (chirp) change of the spool DC motor setpoints. This type of data is desirable to train and learn the dynamical behavior of the FrED system. Both the spool motor response and the diameter response of the FrED system to the chirp spool motor speed input are shown in Figure 6.3 and Figure 6.4.

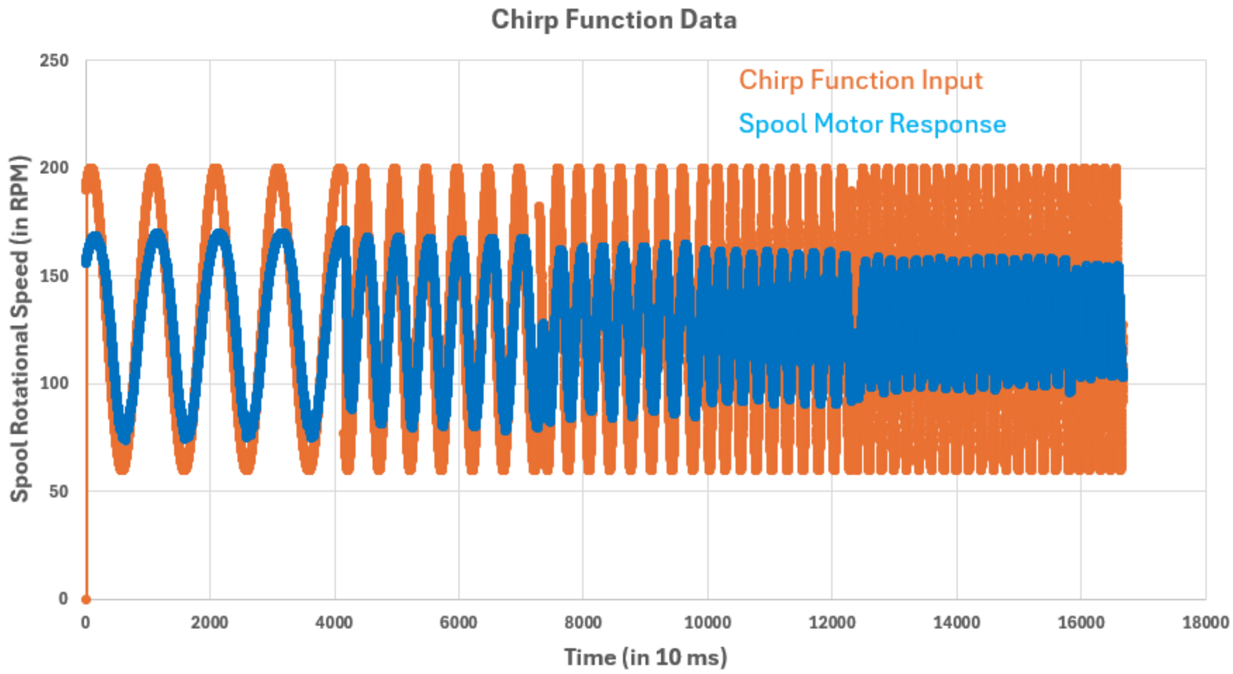


Figure 6.3. Chirp Speed Function Input (for DC Spool Motor).

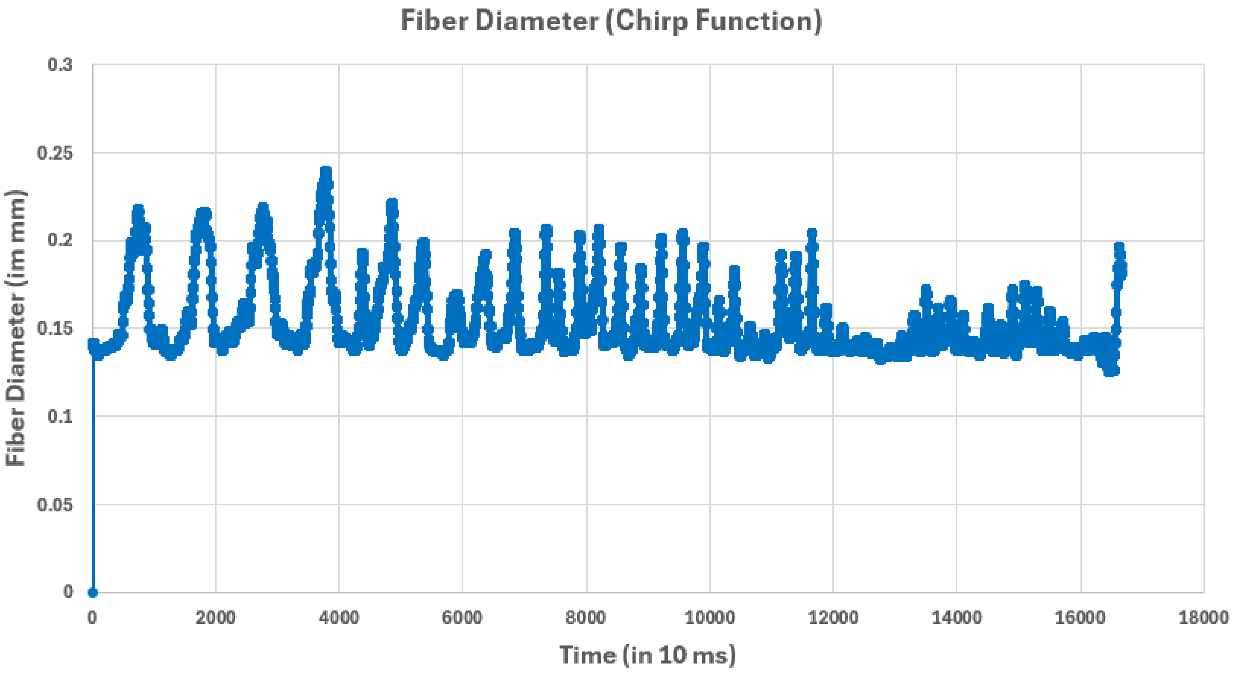


Figure 6.4. Diameter resulting from Chirp Speed Function Input (for DC Spool Motor).

6.2.3 Sinusoid

In this data collection process, the sinusoidal function input of the motor speed is varied according to sinusoidal/decreasing amplitude (chirp) change of the spool DC motor setpoints. This type of data is desirable to train and learn the dynamical behavior of the FrED system. Both the spool motor response and the diameter response of the FrED system to the sinusoidal spool motor speed input (of varied frequencies) are shown in Figure 6.5 – Figure 6.12. The spool motor speed input varied from 40 rpm – 220 rpm.

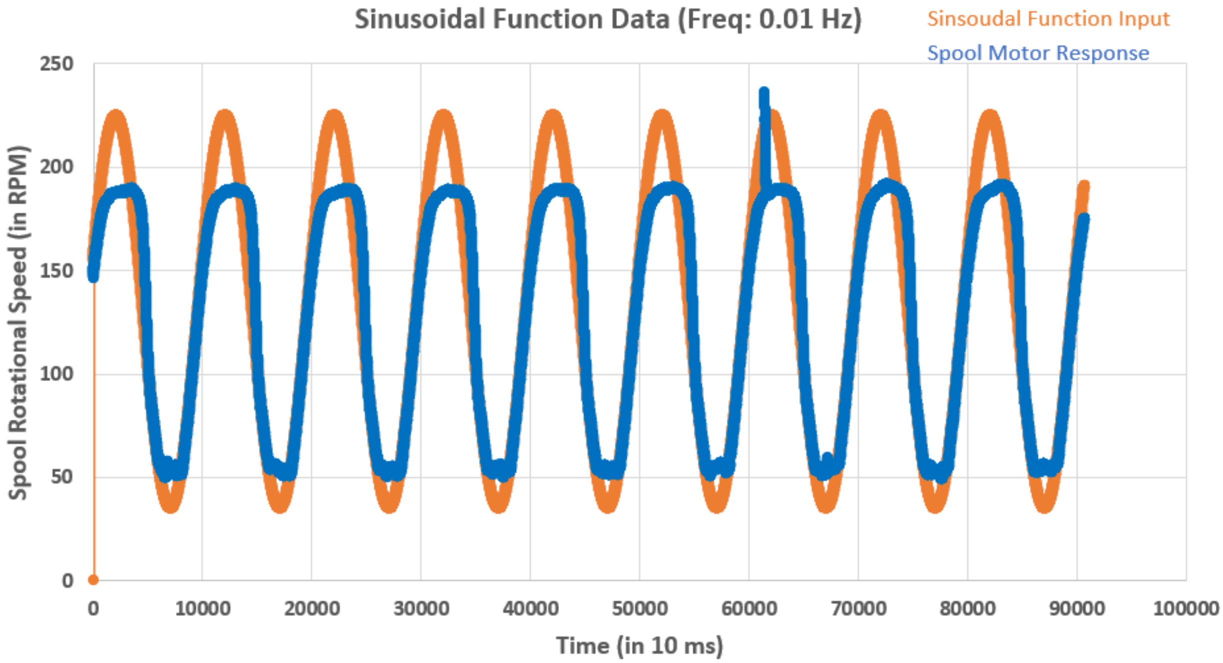


Figure 6.5. Freq: 0.01Hz Sinusoid Speed Function Input (for DC Spool Motor).

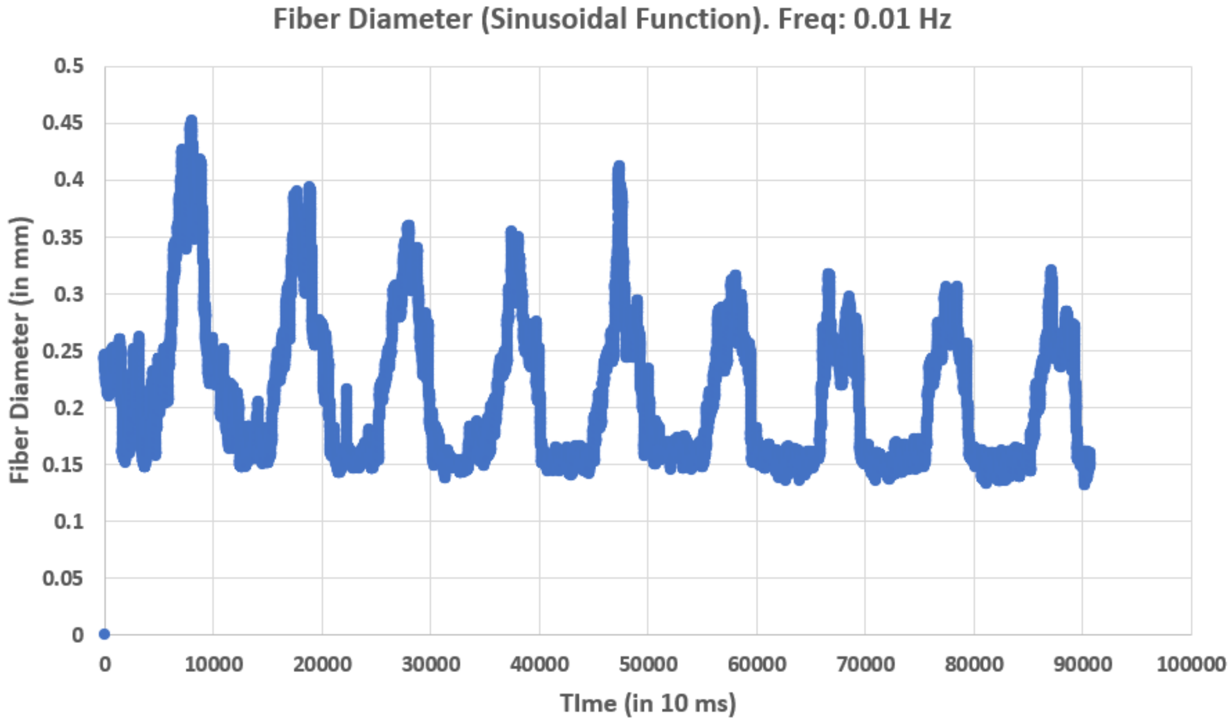


Figure 6.6. Diameter resulting from Freq: 0.01Hz Sinusoid Speed Function Input (for DC Spool Motor).

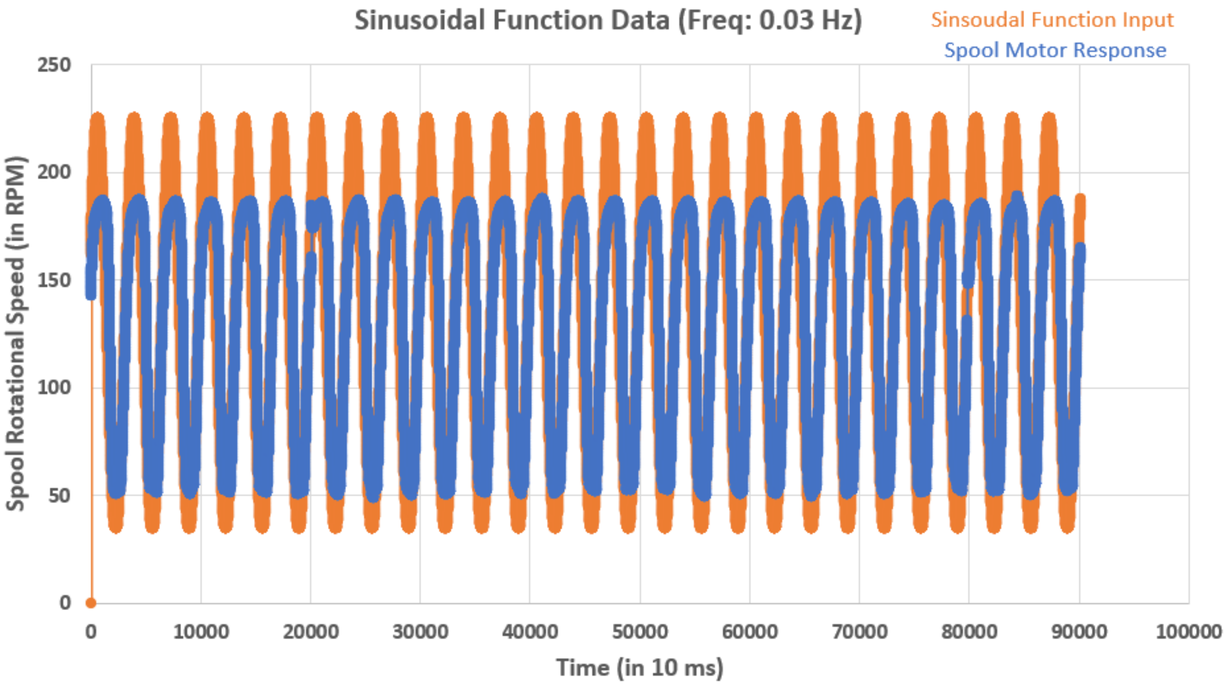


Figure 6.7. Freq: 0.03Hz Sinusoid Speed Function Input (for DC Spool Motor).

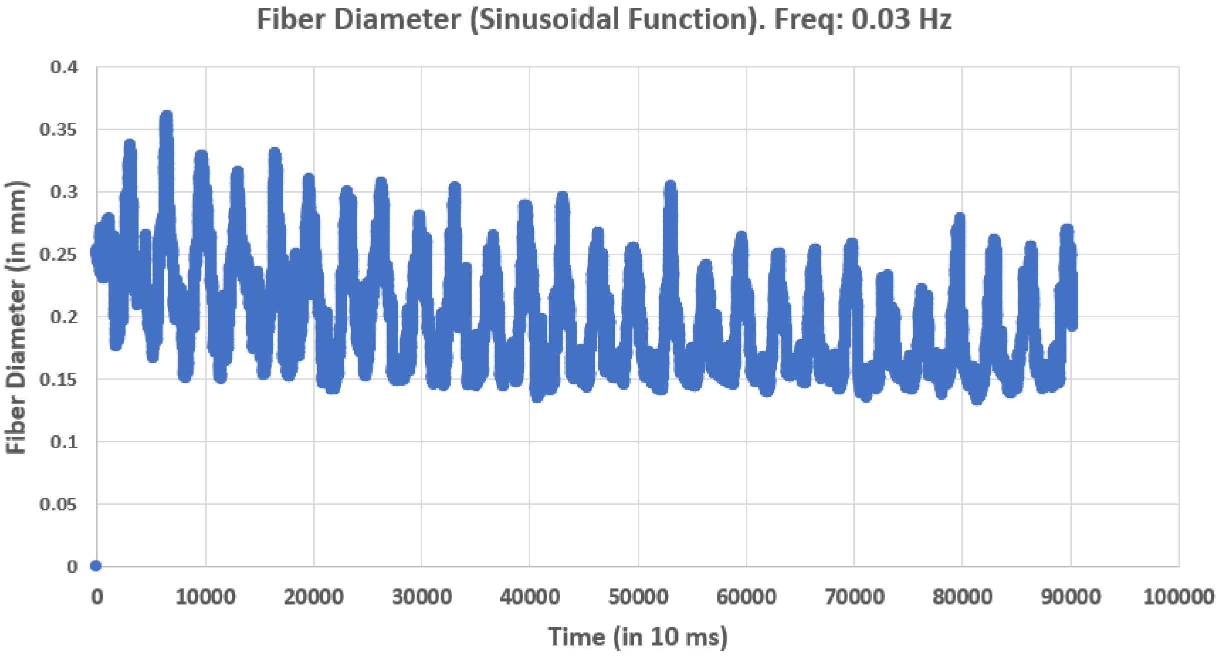


Figure 6.8. Diameter resulting from Freq: 0.03Hz Sinusoid Speed Function Input (for DC Spool Motor).

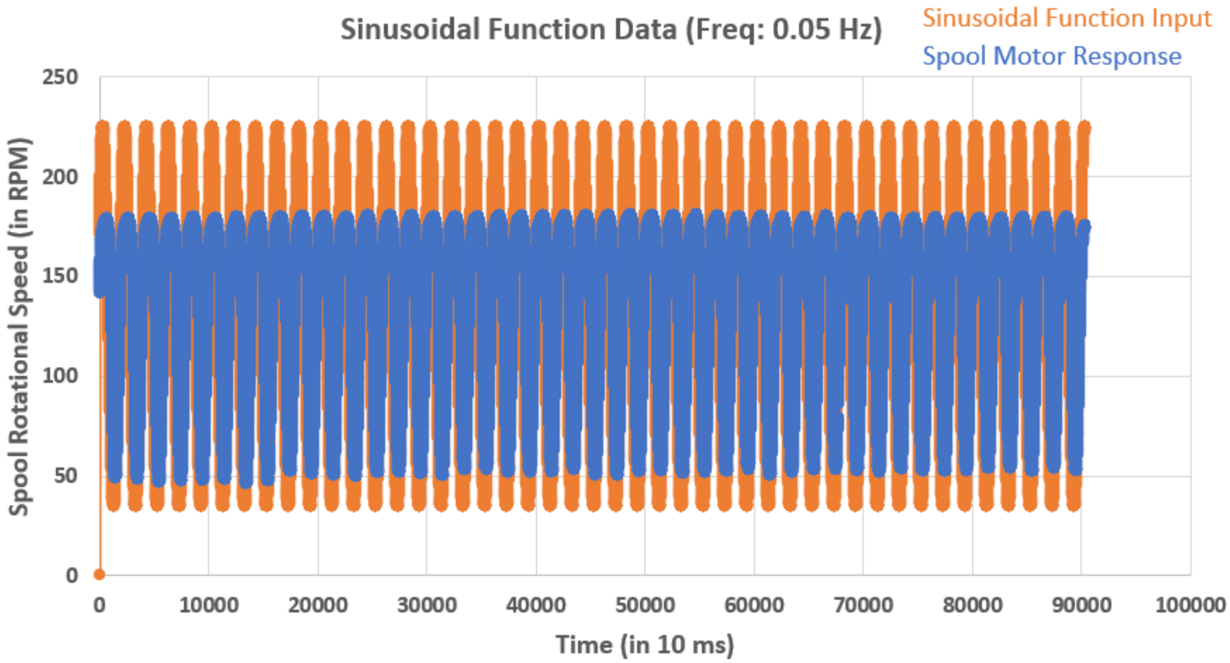


Figure 6.9. Freq: 0.05Hz Sinusoid Speed Function Input (for DC Spool Motor).

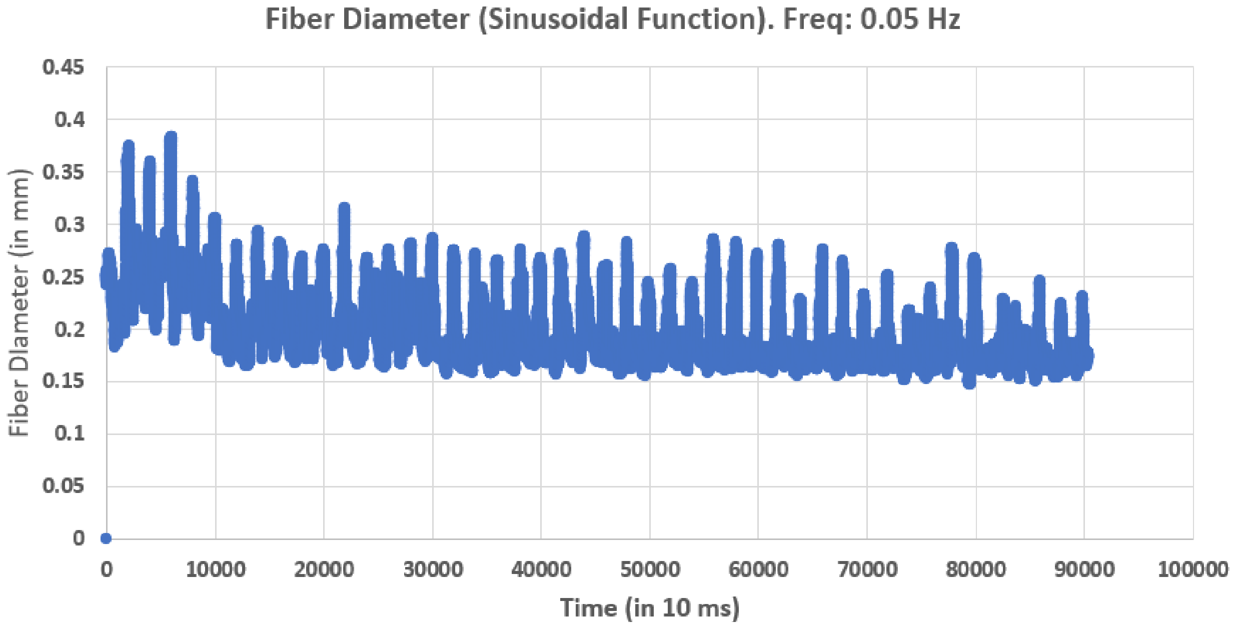


Figure 6.10. Diameter resulting from Freq: 0.05Hz Sinusoid Speed Function Input (for DC Spool Motor).

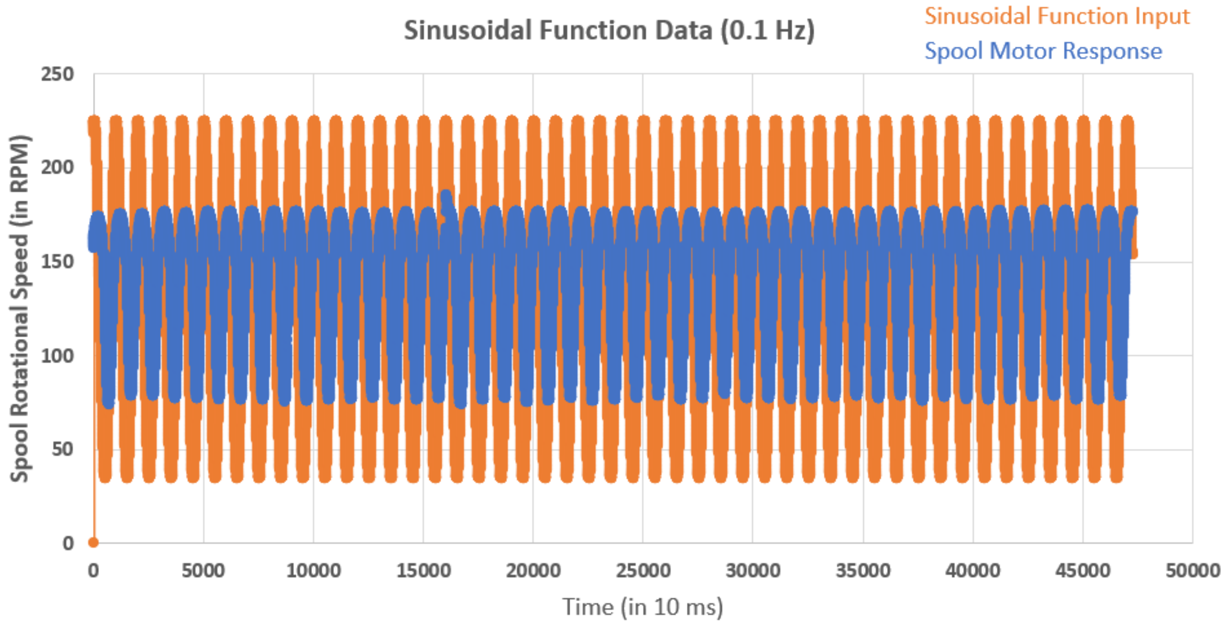


Figure 6.11. Freq: 0.1Hz Sinusoid Speed Function Input (for DC Spool Motor).

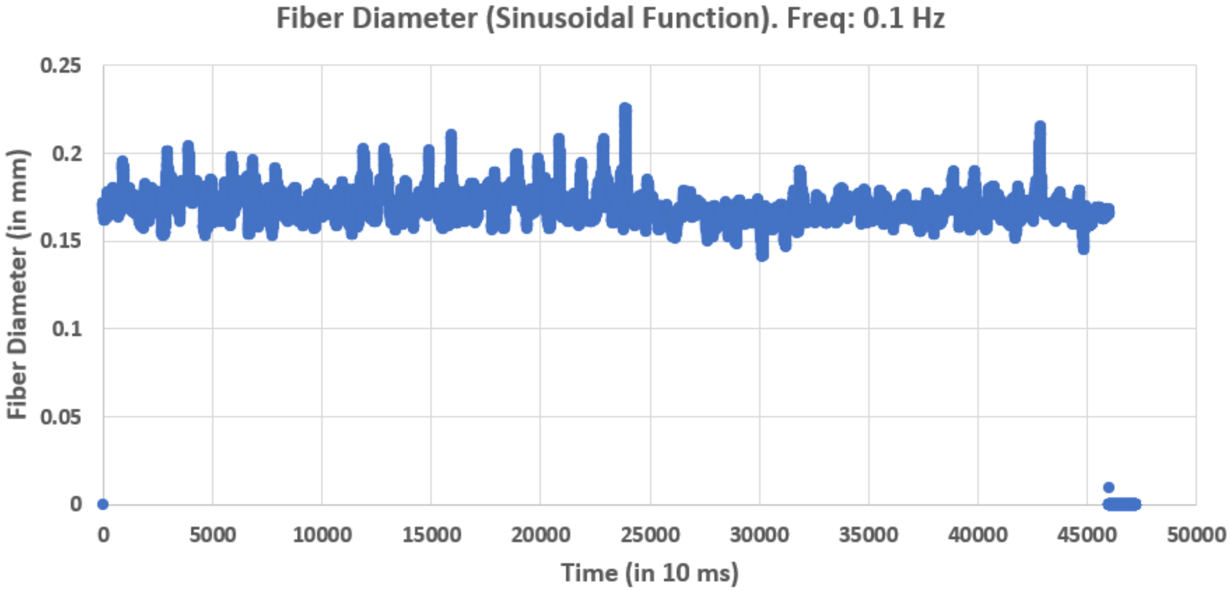


Figure 6.12. Diameter resulting from Freq: 0.1Hz Sinusoid Speed Function Input (for DC Spool Motor).

6.3 Deep Learning and Reinforcement Learning

6.3.1 Deep Learning and Reinforcement Learning Summary

Deep Learning (DL) is a subfield of machine learning that involves the use of artificial neural networks (NN) to model problems with large amounts of data. Deep neural networks with multiple layers (hence the term "deep") allow for accurate pattern extraction from raw data. On the other hand, the underlying principle of Reinforcement Learning (RL) involves two parties: the agent and the environment. The environment is an array of variables that best describe the system (which in this case is the FrED system) within which the RL model is functioning at any given time. The interaction of the agent and the environment is shown in Figure 6.13.

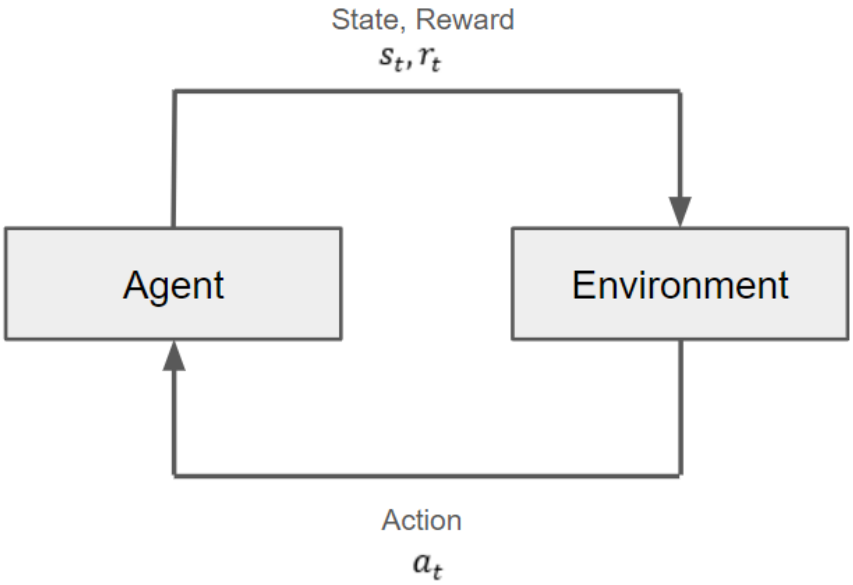


Figure 6.13. Blackbox model learned control diagram (5).

At every time step, t, the agent acts on the environment (induces change in the variables that describe the environment) according to a reward. The goal of the agent is to maximize the total reward for the total duration of the life of the model. The learned model that has been implemented on FrED is DDPG (Deep Deterministic Policy Gradient), which has features derived from both RL and DNN. Some terminologies associated with DDPG model architecture are discussed in section 6.3.2.

6.3.2 Deep Learning and Reinforcement Learning Terminology

This section discusses some key terminology that is needed to accurately describe the DDPG architecture in section 6.4.

States/Observations (s/o):

A state is a complete representation of the model representing the physical system, which in this case is FrED. An observation is a partial representation of this model. Both state and observation are represented through variables stored in matrix, array, or tensors in RL models. These variables for the FrED system may include motor speed, preform heater temperature, fiber diameter, etc.

Action Spaces:

Action spaces are a topological map of valid states the system can take on. Any point in this action space may have parameter values associated with a certain system state. These parameter values can either be discrete or continuous. For example, the FrED system may have certain discrete spool motor speed it could take on due to the limitations of the resolution of the PLC DC output module.

Policies:

Policies are rules or functions that define an action which modifies the state of the system. Policies can either be deterministic or stochastic. For the DDPG model implemented in this paper, the actions are deterministic and can be mathematically expressed in the following way,

$$a_t = \mu_\theta(s_t) \tag{6.1}$$

Here, policy is the agent's intelligence output a quantitative action based on the current state s_t and parameters of the policy θ (weights, biases, etc).

Trajectories:

Trajectories are compilation of states and actions in the environment within which the system resides in. This can be mathematically expressed in the following way,

$$\tau = (s_0\ , a_0\ , s_1, a_1, \ldots\ldots) \tag{6.2}$$

It is a timestamp of the states and actions that the system experiences in the environment it exists in. The system moves from one state to another as time goes from t to $t + 1$. This is mathematically expressed as the following,

$$s_{t+1} = f(s_t, a_t) \tag{6.3}$$

Rewards:

The reward function is crucial to aligning the behavior of the model, so it progressively approaches the desired outcome. Reward functions depend on the current state of the model, the action just taken and the next state of the model. This can be mathematically expressed the following way,

$$r_t = R(s_t\ , a_t\ , s_{t+1}) \tag{6.4}$$

Reward functions can often be expressed as a finite/infinite series sum. The finite-horizon undiscounted reward function is shown in eq (6.5). This is just the sum of the rewards obtained in a fixed time step.

$$R(\tau) = \textstyle\sum_{t=0}^{\infty} \gamma^t r_t \tag{6.5}$$

On the other hand, the infinite-horizon discounted return is shown in Eq (6.6). It is the sum of all the rewards with a discount factor, $\gamma\ \in (0,1)$.

$$R(\tau) = \textstyle\sum_{t=0}^{\infty} \gamma^t r_t \tag{6.6}$$

This type of reward function is a little easier to converge as the discount factor γ can be tuned based on performance to achieve convergence. The finite-horizon undiscounted reward function is easier to implement, therefore in this study this was the chosen reward function as a first step.

6.4 RL/DNN Architecture

6.4.1 RL Algorithm

The objective of the Reinforcement Learning (RL) algorithm is to implement a policy which maximizes the expected reward function when an agent acts according to it. Figure 6.14 shows the wide range of RL algorithms available to be used.

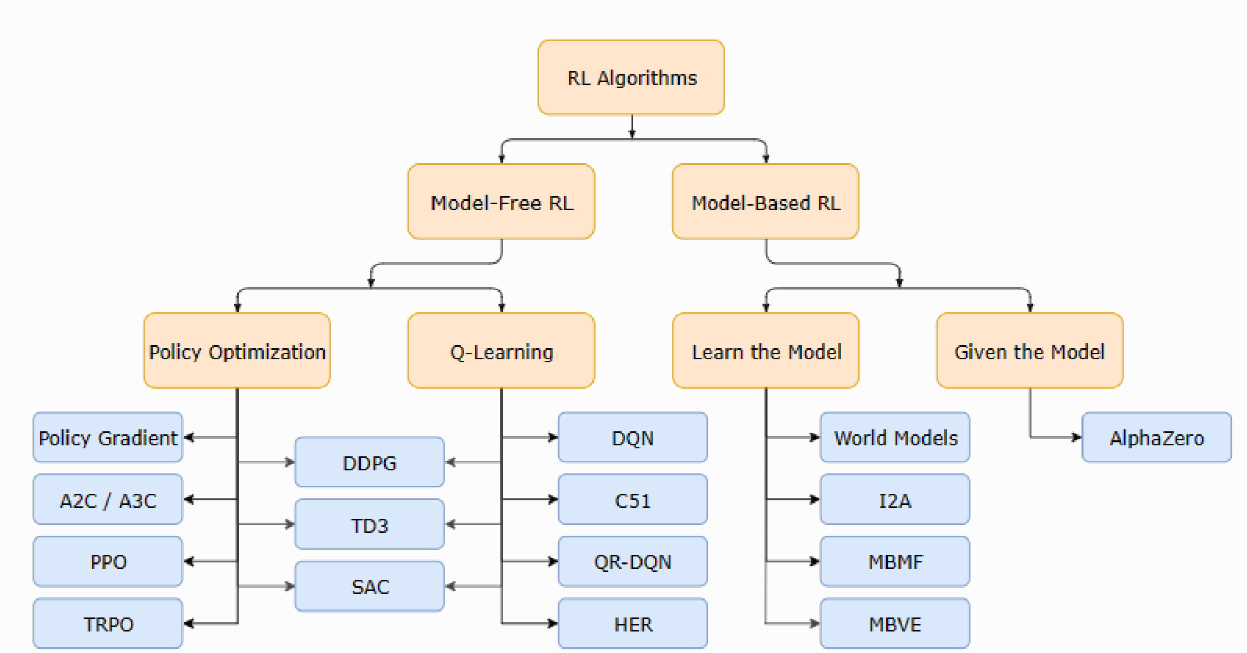


Figure 6.14. Taxonomy of ML Models available.

There have been some attempts in this research work to derive first principles-based Model-Based RL models. This reasoning behind utilizing such model-based RL is that it allows the agent to use an already established model to predict and act ahead. This not only increases the response time but also the efficiency of the RL model as much of the learning is done ahead of time using laws derived from first principles. The major downside of such that was also experienced while attempting to derive the first principles model is that while the agent performs in an optimal way per the theoretical model, it performs sub-optimally while deployed in the real environment. This is due to the complicated nature of the electro-mechanical device such as FrED that involves many coupled thermos-fluidic processes, electrical processes, etc, which are difficult to accurately model in the first place. Therefore, the model-based RL is better when the physical behavior of the system can be accurately predicted (even if partially) using theory/first principles.

On the other hand, there is a model-free approach which makes a black-box model by fully learning from the environment using the actor-critic architecture. The disadvantage of the model-free method is that it performs sub-optimally in the initial stages of learning and takes a lot more time and data to train to get it to a state that is comparable to a model-based RL. The major advantage of the model-based RL and the reason why it was chosen to be implemented in this thesis is that it is much easier to implement since no bias from the theoretical models feed into the RL model. This makes the model-free RL much easier to implement and with enough data/training much easier to converge.

6.4.2 Policy Optimization, and Q-Learning.

There are two ways to go about training the agents in a model-free RL to obtain optimal performance. The first are policy optimization algorithms. The policy for a given action a and state s for given parameter θ can be expressed per the eq (6.7)

$$\pi_\theta(a|s) \tag{6.7}$$

The performance of the policy can be optimized by maximizing the expected return of the performance objective function and optimize the gradient ascent as shown by Silver et al. through eq (6.8) and eq (6.9) (14)

$$J(\pi_\theta) = E[R(\pi_\theta)] \tag{6.8}$$

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\pi_\theta)|_{\theta_k} \tag{6.9}$$

The next step in using this algorithm is to obtain poly gradient expressions that are numerically computable. The derivation of the numerically executable is rather extensive and was omitted from this paper for sake of brevity.

$$\hat{g} = \frac{1}{|D|} \sum_{\tau \in D} \sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t) R(\tau) \tag{6.10}$$

Where the $D = \pi_{\theta_i}\ where\ i = 1, \ldots.., N$. *N* is the total number of trajectories.

The second approach is the Q-Learning algorithms. The basis of this approach relies on learning an approximator ($Q_\theta(s,a)$) that an optimal action value function, $Q^*(s,a)$ optimizes. The best of both worlds involves utilizing advantageous features of both policy optimization and Q-learning. DDPG (Deep Deterministic Policy Gradient) is one such algorithm that use a combination of both of these methods to simultaneously learn a deterministic policy and a Q-function by utilizing one to improve the other.

6.4.3 DDPG/SAC

Bellman equations are utilized to update this optimal action value function per eq (6.11).

$$Q^*\ (s,a) = \max_a \mathop{\mathrm{E}}_{s'\sim P}[r(s,a)\ + \gamma V^*(s')\ ] \tag{6.11}$$

Where $s'{\sim}P$ means the state at $t+1$ is sampled from a distribution P.

Another key equation is the Means Squared Bellman Error (MSBE) as show in eq (6.12)

$$\boldsymbol{L(\phi, D)} = \mathop{\mathrm{E}}_{\left(s,a,r,s',d\right)\sim D}\left[\left(\boldsymbol{Q_\phi(s,a)} - \left(\boldsymbol{r + \gamma(1-d)}\max_{a'}\boldsymbol{Q_\theta(s',a')}\right)\right)^{\boldsymbol{2}}\right] \tag{6.12}$$

Here $\boldsymbol{L(\phi, D)}$ describes how close the Bellman equations are satisfied. It is a measure of the optimal policy and therefore most of the Q-Learning algorithms such as DDPG seek to minimize this loss function. DDPGs also have two key features that enable this minimization. The first key is implementing target (Q' $(s,a|\theta^Q)$ and critic networks $\mu'(s|\theta^{\mu'})$). This allows Q-function to be more like the target. Using polyak averaging, the target network is updated once per main network update. This can be mathematically described using eq (6.13)

$$\theta^{Q'} \leftarrow \rho\theta^{Q} + (1-\rho)\theta^{Q'} \qquad (6.13)$$

Sangwoon showed that Long Short-Term Memory (LSTM) RNN can be utilized in the actor and critic network to capture the long-term dependencies in the sequential data (10). This helps to overcome the vanishing gradient problem that is usually present in the Vanilla Recurrent Neural Network. Some of the actor/critic network attempts as a first step to simply use RNNs and then as a next step utilize the more sophisticated LSTM networks to resolve the vanishing gradient issue.

The second key is the replay buffers which are a set of previous experiences. This buffer must be sized so the array of experiences stored is substantial and diverse enough to achieve stability and convergence. One of the significant challenges encountered when employing neural networks for reinforcement learning lies in the assumption made by optimization algorithms that samples are independently and identically distributed. This assumption becomes problematic when dealing with sequentially generated samples in processes such as fiber extrusion, where the extruded diameter at time step t is correlated with the diameter at time step $t-n$, where $n-1,2,3$ ....To address this issue, Lillicrap et al showed that a finite replay buffer could be used. The buffer stores transition values, with the oldest samples being discarded when the buffer is full. At each time step, the actor and critic functions are updated by uniformly sampling a minibatch from the replay buffer. This approach proves beneficial for DDPG, which is an off-policy algorithm. This is because a large replay buffer allows the algorithm to learn from uncorrelated transitions.

In a manner like Q-learning algorithms, Lillicrap et al., updates actor and critic functions (3). However, in DDPG, these updates employ soft updates rather than direct weight copying from target networks. Specifically, a copy of the actor and critic networks, denoted as is created. The weights of these target networks are then updated to track the learned networks over time. The other element of this the Soft Actor Critic (SAC). A key aspect of the Soft Actor-Critic (SAC) algorithm is the incorporation of entropy regularization. In this approach, the policy is trained to optimize a balance between expected return and entropy, where entropy serves as a metric for the randomness inherent in the policy. This concept is closely tied to the exploration-exploitation trade-off: by enhancing entropy, the policy engages in more exploration, potentially expediting learning in subsequent stages. Additionally, this regularization prevents the policy from prematurely settling into undesirable local optima, thereby promoting a more robust and effective learning process.

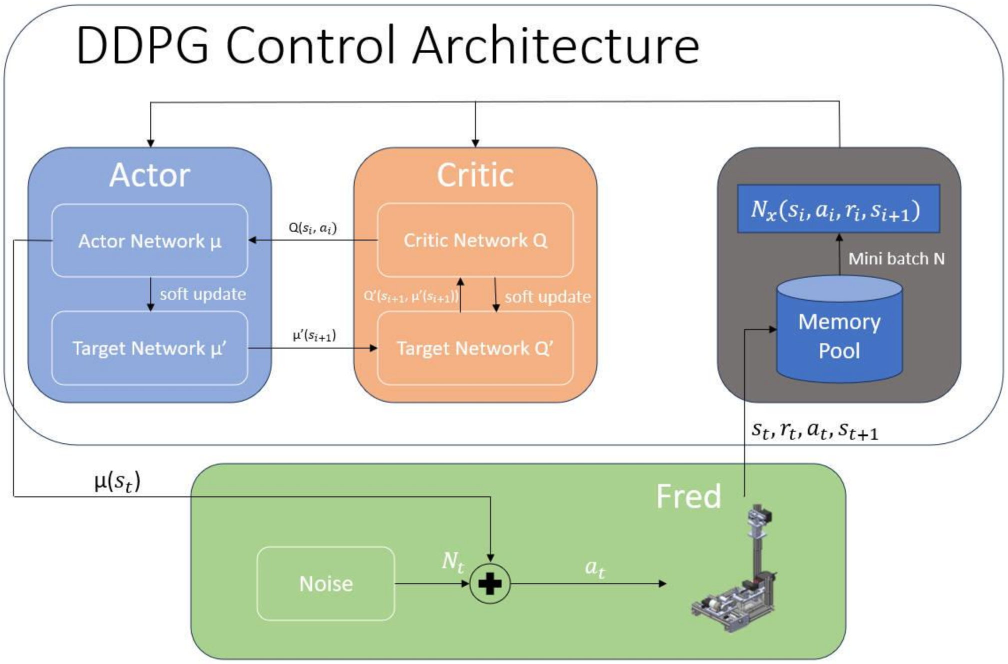


Figure 6.15. DDPG Control Architecture (5)

The DDPG algorithm based of the work done by , Lillicrap et al., has been implemented in this study to regulate the diameter of glue extruded by FREd is outlined as follows (3):

Algorithm 2: DDPG

```
1: Randomly initialize critic network $Q(s, a|\theta^{Q})$ and actor $\mu(s|\theta^{\mu})$ with weights $\theta^{Q}$ and $\theta^{\mu}$.
2: Initialize target network Q' and $\mu'$ with weights $\theta^{Q'} < \theta^{Q}, \theta^{\mu'} < \theta^{\mu}$
3: Initialize replay buffer $R$
```

```
for epoch = 1, M do
    Initialize a random process N for action exploration using the Ornstein-Uhlenbeck process
    Receive initial observation state $s_1$
    for t=1, T do
        Select action $a_t = clip(\mu(s_t|\theta^{\mu}) + N_t, \alpha_{Low}, \alpha_{High})$ according to the current policy and exploration noise
        Execute action $a_t$ and observe reward $r_t$ and observe new state $s_{t+1}$
        Store the transition $(s_t, a_t, r_t, s_{t+1})$ in R
        Sample a random minibatch of N transitions $(s_t, a_t, r_t, s_{t+1})$ from R
        Set $y_i = r_i + \gamma Q'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
        Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i\left(y_i - Q(s_i, a_i|\theta^{Q})\right)^2$
        Update the actor policy using the sampled policy gradient:
            $\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^{Q})|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s_i}$
        Update the target networks:
            $\theta^{Q'} \leftarrow \rho\theta^{Q} + (1-\rho)\theta^{Q'}$
            $\theta^{\mu'} \leftarrow \rho\theta^{\mu} + (1-\rho)\theta^{\mu'}$
    end for
end for
```

6.4.4 Twin Delayed DDPG

One of the disadvantages of the DDPG algorithm as discussed in section 6.4.3 is that it is too sensitive to the tuning of the hyperparameter. For instance, in conventional DDPG logic, the Q-function tends to overestimate the Q-values, which leads to policy breaking especially if the hyperparameters aren't tuned exactly right. There are three enhancements to the conventional DDPG algorithm that are discussed in this section. Firstly, instead of using one Q function, this twin delayed DDPG algorithm uses two Q functions and uses the better one (as defined in the eq (6.14))). Two Means Squared Bellman Error (MSBE) loss functions are depicted in eq (6.14) (14).

$$\boldsymbol{L(\phi_i, D)} = \underset{(s,a,r,s',d)\sim D}{\mathrm{E}}\left[\left(\boldsymbol{Q_{\phi_i}(s,a)} - \left(\boldsymbol{r} + \boldsymbol{\gamma(1-d)}\max_{a'}\boldsymbol{Q_{\theta}(s',a')}\right)\right)^{2}\right] \qquad (6.14)$$

$$where\ i = 1{,}2$$

Another enhancement is the introduction of clipped noise as shown in eq (6.15). This noise in the target action makes the Q-function less sensitive to abrupt changes in the policy (15).

$$a(s) = clip(\mu(s_t|\theta^{\mu}) + clip(N_t, -c, c), \alpha_{Low}, \alpha_{High})$$

In addition, another enhancement is the delaying of that the policy and the target are updated less frequently than the Q-function. This is implemented by putting in a conditional (line 14) as shown in the Algorithm 3 pseudocode near the end of this section. This algorithm is based on the work done by Fujimoto et al. (15).

Algorithm 3: Twin Delayed DDPG

```
1: Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^{\mu})$ with weights $\theta^Q$ and $\theta^{\mu}$.
2: Initialize target network Q' and $\mu'$ with weights $\theta^{Q'} < \theta^Q, \theta^{\mu'} < \theta^{\mu}$
3: Initialize replay buffer R
4: for epoch = 1, M do
5:     Initialize a random process N for action exploration using the Ornstein-Uhlenbeck process
6:     Receive initial observation state $s_1$
7:     for t=1, T do
8:         Select action $a_t = clip(\mu(s_t|\theta^{\mu}) + N_t, \alpha_{Low}, \alpha_{High})$ according to the current policy and exploration noise
9:         Execute action $a_t$ and observe reward $r_t$ and observe new state $s_{t+1}$
10:        Store the transition $(s_t, a_t, r_t, s_{t+1})$ in R
11:        Sample a random minibatch of N transitions $(s_t, a_t, r_t, s_{t+1})$ from R
12:        Compute target actions, $a(s) = clip(\mu(s_t|\theta^{\mu}) + clip(N_t, -c, c), \alpha_{Low}, \alpha_{High})$
```

```
        Set $y_i = r_i + \gamma Q_i'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
        if  j mod policy-delay = 0 then
            Update critic by minimizing the loss:$L = \frac{1}{N}\sum_i\left(y_i - Q(s_i, a_i|\theta^Q)\right)^2$
                Update the actor policy using the sampled policy gradient:
                    $\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$
            Update the target networks:
                    $\theta^{Q'} \leftarrow \rho\theta^Q + (1-\rho)\theta^{Q'}$
                    $\theta^{\mu'} \leftarrow \rho\theta^\mu + (1-\rho)\theta^{\mu'}$
    end for
end for
```

6.5 DDPG Hyperparameter Tuning Process

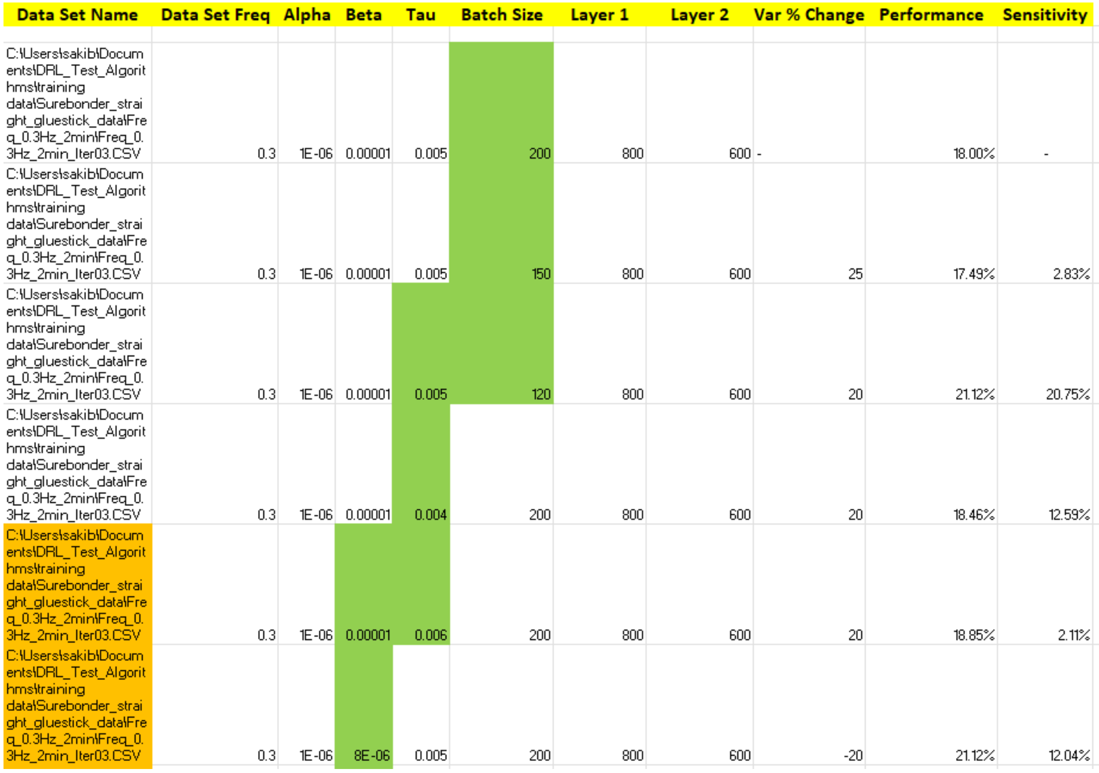
According to Patrick, model performance and convergence is highly sensitive to the following hyperparameters that were concluded to be optimal (5). In this section a DOE (Design of Experiment)/ sensitivity study is done to find the optimal set of parameters with a reasonable convergence time. The optimized set of parameters, according to Patrick, is shown in table 6.1.

Table 6.1: DDPG Pseudo-Optimized Parameters (Patrick (5))

Parameter	**Value**
Alpha	0.000001
Beta	0.00001
Tau	0.005
Batch Size	120
Layer 1 Size	200
Layer 2 Size	800
OU Sigma / Speed / Decay rate	0.15 / 0.2 / 1e-2

The highest accuracy value Patrick obtained from the hyperparameter values defined in table 6.1 is 18.4% (5). As the near-term goal of this research is to utilize an offline trained DDPG model to respond to changing system dynamics (mainly diameter setpoint), the optimized parameters here are chosen without taking convergence time into account as opposed to Patrick (5). With this goal in mind, a DOE study was conducted to build on top of Patrick's work on optimizing hyperparameter (5). In the DOE study, the hyperparameters were systematically tweaked (on the order of about 20%-25%) to see the sensitivity of the accuracy/prediction value to each hyperparameter. The DOE study was organized in the manner shown in table 6.2.

Table 6.2: Hyperparameter DOE Study



Data Set Name	Data Set Freq	Alpha	Beta	Tau	Batch Size	Layer 1	Layer 2	Var % Change	Performance	Sensitivity
C:\Users\sakib\Documents\DRL_Test_Algorithms\training data\Surebonder_straight_gluestick_data\Freq_0.3Hz_2min\Freq_0.3Hz_2min_Iter03.CSV	0.3	1E-06	0.00001	0.005	200	800	600	-	18.00%	-
C:\Users\sakib\Documents\DRL_Test_Algorithms\training data\Surebonder_straight_gluestick_data\Freq_0.3Hz_2min\Freq_0.3Hz_2min_Iter03.CSV	0.3	1E-06	0.00001	0.005	150	800	600	25	17.49%	2.83%
C:\Users\sakib\Documents\DRL_Test_Algorithms\training data\Surebonder_straight_gluestick_data\Freq_0.3Hz_2min\Freq_0.3Hz_2min_Iter03.CSV	0.3	1E-06	0.00001	0.005	120	800	600	20	21.12%	20.75%
C:\Users\sakib\Documents\DRL_Test_Algorithms\training data\Surebonder_straight_gluestick_data\Freq_0.3Hz_2min\Freq_0.3Hz_2min_Iter03.CSV	0.3	1E-06	0.00001	0.004	200	800	600	20	18.46%	12.59%
C:\Users\sakib\Documents\DRL_Test_Algorithms\training data\Surebonder_straight_gluestick_data\Freq_0.3Hz_2min\Freq_0.3Hz_2min_Iter03.CSV	0.3	1E-06	0.00001	0.006	200	800	600	20	18.85%	2.11%
C:\Users\sakib\Documents\DRL_Test_Algorithms\training data\Surebonder_straight_gluestick_data\Freq_0.3Hz_2min\Freq_0.3Hz_2min_Iter03.CSV	0.3	1E-06	8E-06	0.005	200	800	600	-20	21.12%	12.04%

Patrick determined "the optimal batch size to allow for the model to learn the complete system dynamics between the onset of a change in the spool motor speed and an observable change in the diameter is 10.5 seconds or 1050 data points. If we were to use a batch size of 1050 data points a single calculation would take 90 milliseconds on a Razer Blade 15 laptop with an Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz   2.59 GHz processor which was deemed as too long (5)." It was observed from the DOE study that increasing batch size doesn't necessarily increase model performance. This maybe to due overfitting errors that creep with large batch sizes which may lead to loss in ability for generalized prediction. The optimum set of parameters while training and testing the model on sinusoid data is summarized in table 6.3. Running the model with this set of parameters resulted in the highest accuracy (21.12 %) and lowest sensitivity (12.04%). This model was run on AMD Ryzen 7 5825U with Radeon Graphics (2.00 GHz Processor).

6.6 Data Quality Analysis

The amount of data the DDPG model is trained on is only one of the factors that affect the accuracy of the model. The other key characteristic of the data is that must be of high quality, devoid of noise and rich in system dynamics information. For instance, Figure 6.16 shows examples of low-quality data that resulted from sinusoidal spool speed input (as discussed in section 6.2.3). What makes this data low quality is the fact that due to the low frequency nature of the sinusoidal speed input, the diameter doesn't evenly span all the possible values and tends to stay at the lower limit of 0.15 mm for a long time. The accuracy for diameter data shown in Figure 6.16 is ~ 14% and the DDPG result is shown in Figure 6.17. On the other hand, a high-quality diameter data is when a sinusoidal spool speed input frequency of 0.3 Hz is used as shown in Figure 6.18. What makes this data high quality is the fact that the diameter equally spans all the setpoint ranging from 0.15 mm to 0.26 mm evenly. This allows the DDPG model to learn more of the system dynamics. The accuracy for diameter data shown in Figure 6.18 is ~ 21% and the DDPG result is shown in Figure 6.19.

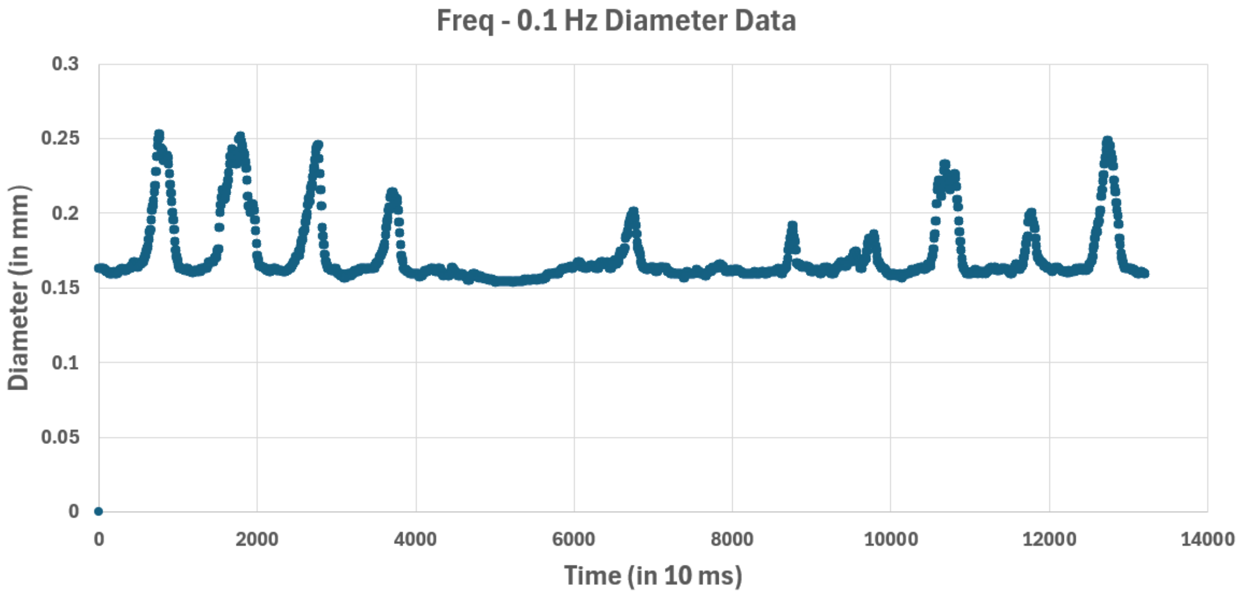
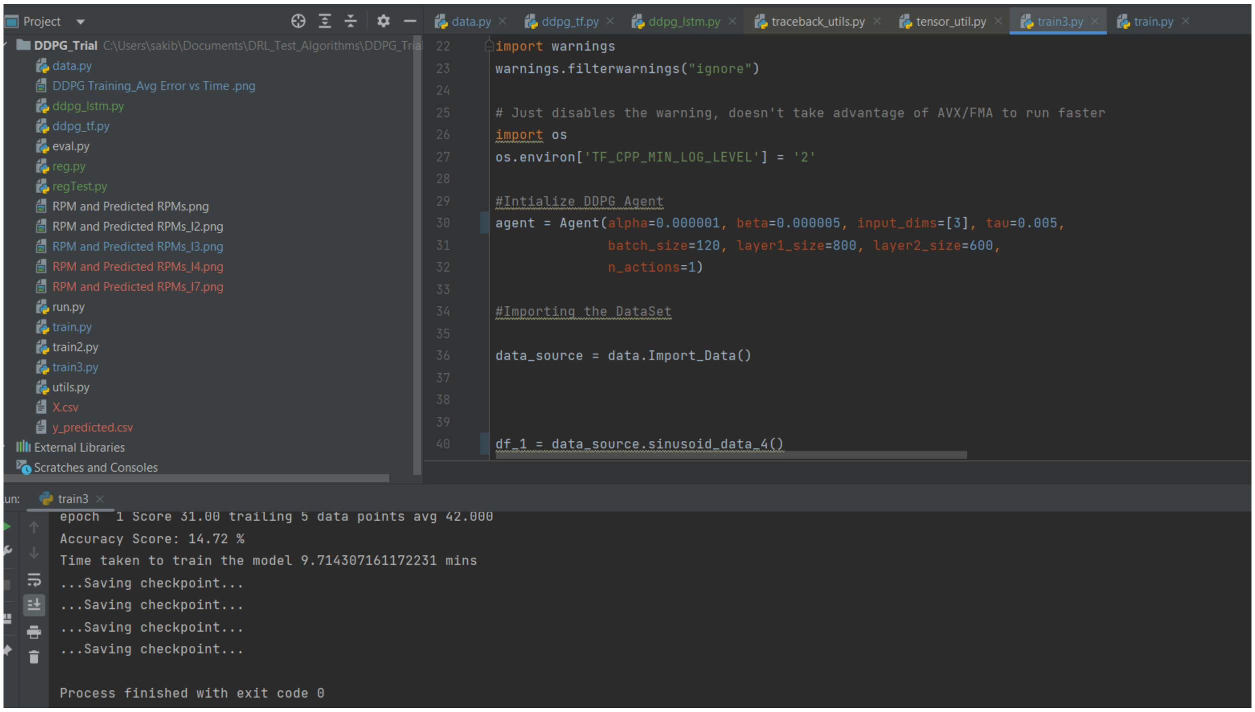


Figure 6.16. Freq – 0.1 Hz Diameter Data (Sinusoidal Spool Speed Input)



```
import warnings
warnings.filterwarnings("ignore")

# Just disables the warning, doesn't take advantage of AVX/FMA to run faster
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

#Intialize DDPG Agent
agent = Agent(alpha=0.000001, beta=0.000005, input_dims=[3], tau=0.005,
              batch_size=120, layer1_size=800, layer2_size=600,
              n_actions=1)

#Importing the DataSet

data_source = data.Import_Data()



df_1 = data_source.sinusoid_data_4()
```

```
epoch  1 Score 31.00 trailing 5 data points avg 42.000
Accuracy Score: 14.72 %
Time taken to train the model 9.714307161172231 mins
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...

Process finished with exit code 0
```

Figure 6.17. Freq – 0.1 Hz Diameter Data DDPG Model Accuracy (Sinusoidal Spool Speed Input)

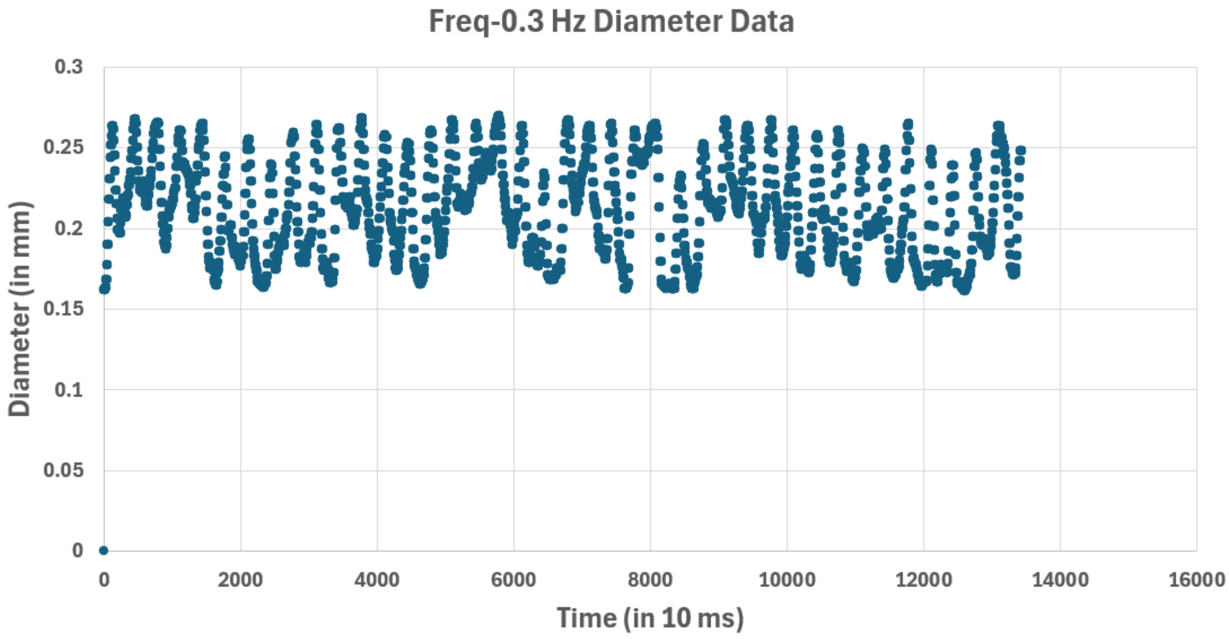
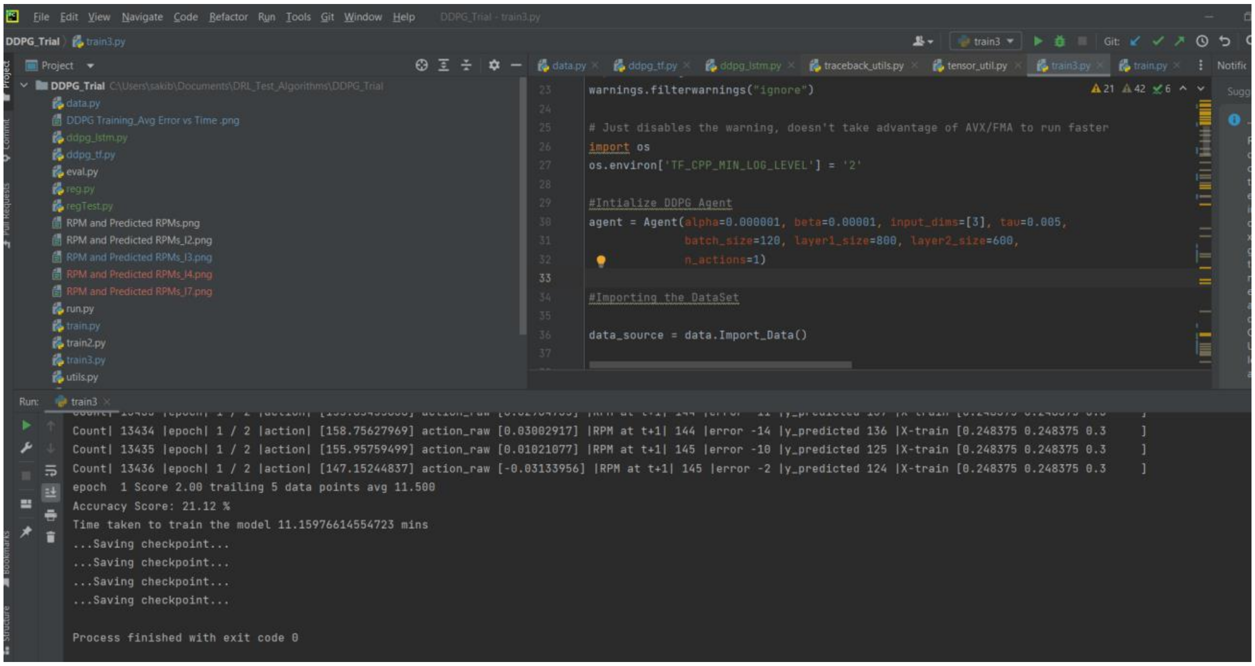


Figure 6.18. Freq – 0.3 Hz Diameter Data (Sinusoidal Spool Speed Input)


File Edit View Navigate Code Refactor Run Tools Git Window Help
DDPG_Trial - train3.py
DDPG_Trial
train3.py
Project
DDPG_Trial C:\Users\sakib\Documents\DRL_Test_Algorithms\DDPG_Trial
data.py
DDPG Training_Avg Error vs Time .png
ddpg_lstm.py
ddpg_tf.py
eval.py
reg.py
regTest.py
RPM and Predicted RPMs.png
RPM and Predicted RPMs_l2.png
RPM and Predicted RPMs_l3.png
RPM and Predicted RPMs_l4.png
RPM and Predicted RPMs_l7.png
run.py
train.py
train2.py
train3.py
utils.py
data.py
ddpg_tf.py
ddpg_lstm.py
traceback_utils.py
tensor_util.py
train3.py
train.py
train3
Git:
Run:
train3

```
warnings.filterwarnings("ignore")

# Just disables the warning, doesn't take advantage of AVX/FMA to run faster
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

#Intialize DDPG Agent
agent = Agent(alpha=0.000001, beta=0.00001, input_dims=[3], tau=0.005,
              batch_size=120, layer1_size=800, layer2_size=600,
              n_actions=1)

#Importing the DataSet

data_source = data.Import_Data()
```

```
Count| 13434 |epoch| 1 / 2 |action| [158.75627969] action_raw [0.03002917] |RPM at t+1| 144 |error -14 |y_predicted 136 |X-train [0.248375 0.248375 0.3     ]
Count| 13435 |epoch| 1 / 2 |action| [155.95759499] action_raw [0.01021077] |RPM at t+1| 145 |error -10 |y_predicted 125 |X-train [0.248375 0.248375 0.3     ]
Count| 13436 |epoch| 1 / 2 |action| [147.15244837] action_raw [-0.03133956] |RPM at t+1| 145 |error -2 |y_predicted 124 |X-train [0.248375 0.248375 0.3     ]
epoch  1 Score 2.00 trailing 5 data points avg 11.500
Accuracy Score: 21.12 %
Time taken to train the model 11.15976614554723 mins
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...

Process finished with exit code 0
```

Figure 6.19. Freq – 0.3 Hz Diameter Data DDPG Model Accuracy (Sinusoidal Spool Speed Input)

6.7 DDPG Offline Performance

The results of the DDPG model that was run with the optimized parameters (summarized in table 6.3) are discussed in this section.

Table 6.3: Optimized Hyperparameter (from DOE Study)

Parameter	**Value**
Alpha	0.000001
Beta	0.000008
Tau	0.005
Batch Size	200
Layer 1 Size	800
Layer 2 Size	600
OU Sigma / Speed / Decay rate	0.15 / 0.2 / 1e-2

The actual vs predicted RPM plot is shown in Figure 6.2. This model was trained and tested on sinusoidal data with a frequency of 0.3 Hz. This model had an accuracy of 21.12% which is higher than the 18.4% reported by Patrick (5).

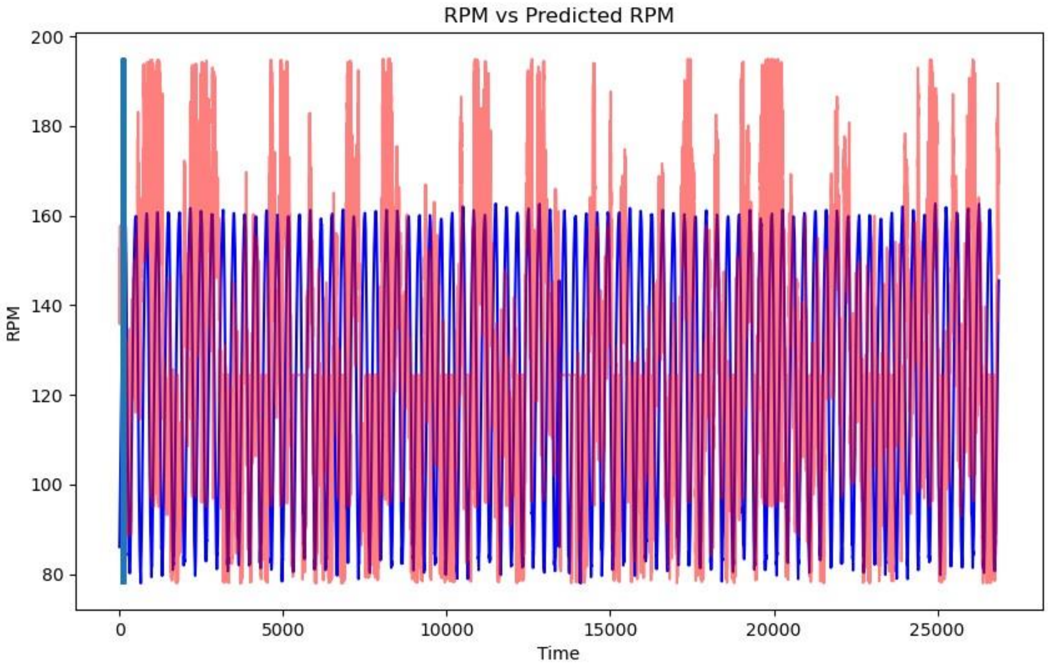


Figure 6.20. Freq – 0.3 Hz Diameter Data DDPG Model Accuracy (Sinusoidal Spool Speed Input)

6.8 DDPG Online Performance

Due to limited resources, the optimal DDPG model obtained from the DOE study in section 6.5 was not implemented on the actual PLC hardware. This section as a result re-summarizes the implementation of the pseudo-optimized model as discussed by Patrick (5). Furthermore, the results are compared to the fully heuristic model in addition to Zeigler Nichol's as described in Zhang (4). To obtain a fair comparison of the optimized parameters, the DDPG model was trained on the same set of 120,000 data points which was described by Patrick (5). The DDPG online performance showed some promise when compared to both Ziegler Nichols and the fully heuristic PID controllers. The DDPG model was tested by changing the diameter setpoints from 0.2 mm to 0.25 mm. The performance of the DDPG model is show in

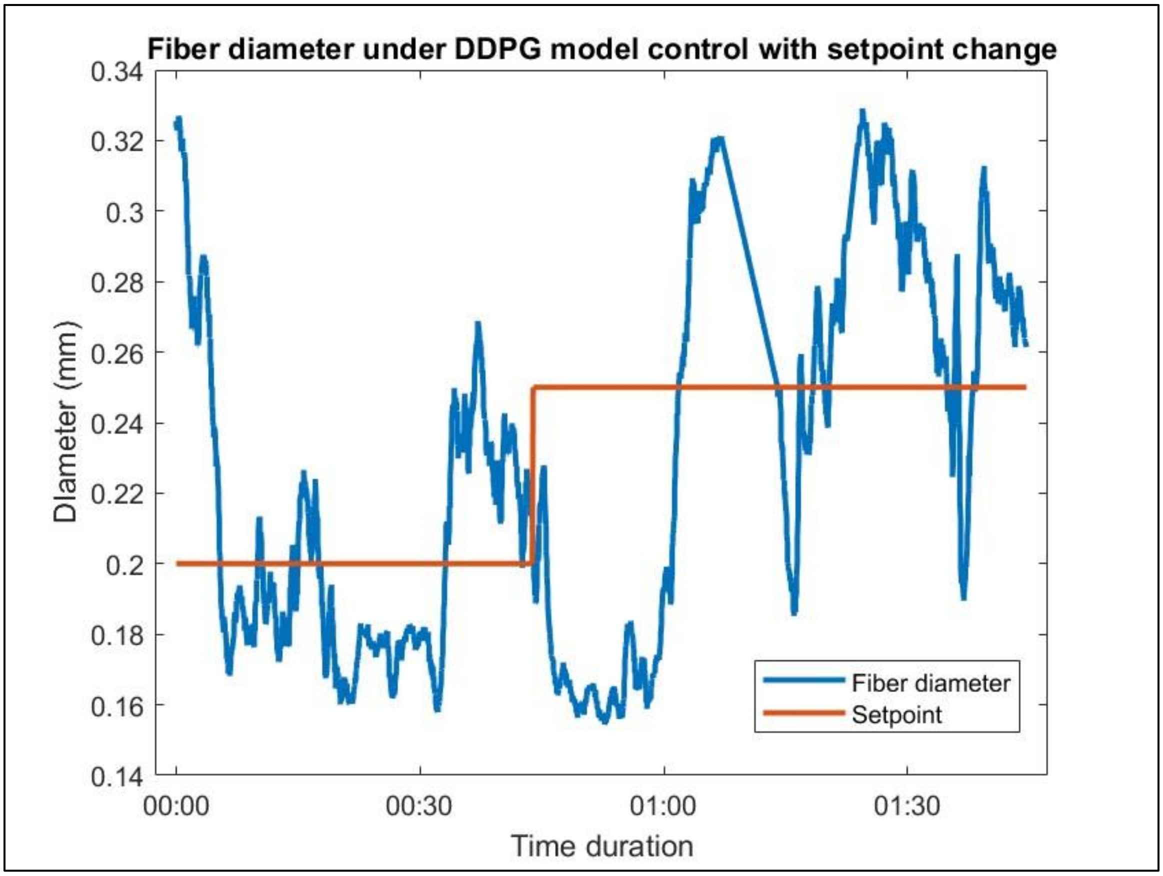


Figure 6.19. Freq – 0.3 Hz Diameter Data DDPG Model Accuracy (Sinusoidal Spool Speed Input)

Table 6.3: Average, standard deviation, range and relative error of the fiber diameter when controlled by the DDPG model (Patrick (5))

Setpoint (mm)	Average (mm)	Standard Deviation (mm)	Range (mm)	Relative error (%)
0.25	0.272	0.035	0.144	8.8
0.20	0.198	0.028	0.111	1

Additional online tests on the DDPG models were run and the results are compared to PID Controller (Zeigler Tuning) and PID Controller (Fully Heuristic). The results are summarized in table 6.4.

Table 6.4: Comparison between the performance of the PID controller (heuristic models) and the DDPG model in a second test

	Set point (mm)	Average (mm)	Standard Deviation (mm)	Range (in mm)	Relative Error (%)
PID controller - Zeigler Method (Zhang) (6)	0.20	0.200	0.020	0.130	0
	0.25	0.247	0.026	0.162	1.2

PID controller (Fully Heuristic)	0.20	0.200	0.010	0.040	0
	0.25	0.245	0.020	0.120	2
DDPG model Test 01(Patrick (5))	0.20	0.198	0.028	0.111	1
	0.25	0.272	0.035	0.144	8.8
DDPG model Test 02 (Patrick (5))	0.2	0.206	0.038	0.200	3
	0.25	0.242	0.038	0.250	3.2

It can be seen from the results shown in table 6.4 that PID controller (Fully Heuristic) and PID controller - Zeigler Method (Zhang) (6) perform the best overall relative to DDPG controller. That is the PID controllers have lower relative error which means the PID controller can reach target set points more accurately and also has lower standard deviation than DDPG. On the other hand, the range achieved by DDPG is comparable to those obtained by PID controller.

Chapter 7 Conclusion and Future Work

7.1 Conclusion

The work done in this thesis is the accumulation of multiple years of work done in parallel and in collaboration with the work done by Patrick (5) and Zhang (4). This work shows that it is possible to implement a learned control system on a simple manufacturing system such as a desktop fiber extrusion device. While learned DDPG learned control method implemented here is not as good as traditional PID control methods, improvement can be achieved by training on high quality and more data (on the order of 10^6 data points) on a HPC (high performance computer). The work discussed in this thesis picked up from where Othman had concluded (9). Through this thesis, mechanical design of the PLC FrED has been improved. This improvement can be quantified through multiple parameters.

Firstly, the implementation of the timing belt system traversing motion has eliminated all perturbances due to backlash. Thermal dependencies have been explored and optimum extrusion temperature of 87C has been determined. The shift to air cooling system from water cooling has shifted the achievable fiber diameter range. The range of fiber diameter the water bath system can achieve is (0.4-0.6) mm, while the range of fiber diameter the air-cooling system can achieve is 0.16 mm to 0.38 mm. The range itself improved by 10% and since the diameter has shifted to a smaller range, the system became more stable as the tensioning mechanism was easier to implement. Furthermore, the closed structural loop extrusion system that was re-designed and discussed in Chapter 4 has been shown to reduce slippage by 16% - 120% depending on the speed of the stepper motor. In addition, two different PID tuning methods have been discussed, Zeigler Nichols and Fully Heuristic methods. The spool speed and heater block temperature has achieved great results from Zeigler Nichols tuning as shown by Zhang (4). The fiber PID controller was further improved by the fully heuristic method that has been discussed in Chapter 5. Furthermore, this thesis improved upon the work of Patrick, by analyzing the training data set and conducting a DOE to optimize the hyperparameter (5). The DDPG model that has been trained is

20% accurate when trained and tested on sinusoidal data while Patrick has been shown to only achieve 18% accuracy (5).

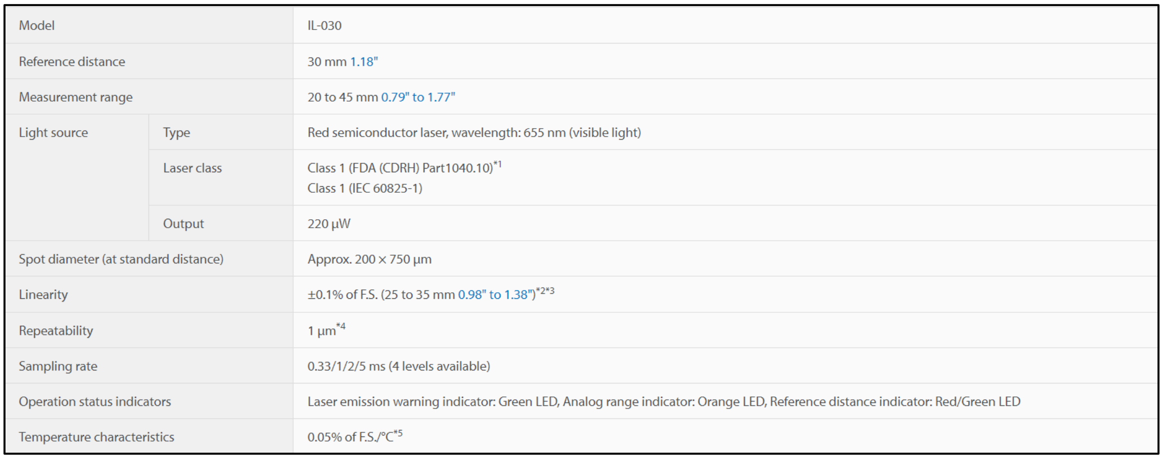
The implication of this study implementation is that while this the current DDPG model is worse than, it can be improved by following some of the recommendations suggested in section 7.2. Simultaneously, the successful implementation of the DDPG model on the desktop optical fiber manufacturing system shows promise that such methods could be implemented in real life manufacturing systems beyond optical fiber manufacturing.

7.2 Recommendations for Future Work

For future work, there can be improvements made in all the three major domains (mechanical design, electrical hardware and learned model) of this project. For the mechanical design, a more robust tensioner can be integrated. The major limitation in achieving the most optimal temperature as discussed in Chapter 4 is because for temperatures higher than 87C the fiber starts to lose tensions especially due to gravity. Therefore, a more robust tensioner would allow for the fiber to be extruded at a higher temperature which in turn would reduce the diameter variation of the extruded fiber. A feedback loop to sense and prevent loss in tension with increasing temperature is also desired. Furthermore, a more efficient cooling system is necessary as with higher temperatures, the cooling needs to be faster to regain the tension of the fiber back. Currently, the cooling system is two motorized fans fixtured in a way to provide forced convection along the length of the fiber. Since the fans are symmetrical, this setup minimizes vibration effects. However, a more uniform cooling method is desired such as with smaller but greater number of fans, water cooling that is more stable/doesn't force the fiber to change direction drastically (as that would lead to loss in tension). Furthermore, a spool diameter sensor needs to be properly integrated into the PLC system. There had been some attempts to implement a spool diameter measurement system (distance sensor mechanism) as shown in Figure 3.24. The instantaneous diameter measurement is a sensitive parameter that the model must know since the linear velocity of the spool (and thus the fiber diameter) is highly dependent on the instantaneous diameter of the spool. Currently the model only knows about the fiber diameter which is only one of the dependencies of fiber diameter. In addition, to increase the resolution limitations of the DC motor, as discussed in section 5.4.3, a PWM specific module (such as the 4402 Point I/O Pulse Width Modulation Output Module) can be integrated. This will help the spool motor reach certain speeds in between the major jumps (see Figure 5.22) or achieve more granularity (at cycle time of 0.5 ms, with duty cycle resolution of 0.1%). This enables more resolution in the fiber diameter that can be obtained which allows the DDPG model to be trained on/tested on a larger dataset without increasing the range/affecting tension. Lastly, the DDPG model can be improved by running on not only a larger dataset (on the order of 10^6) but also higher quality dataset. More high dependency variables can be introduced into the DDPG model, such as fiber tension, spool diameter, fan speed, etc. This will allow the model to learn the system dynamics more completely. Furthermore, although the hyperparameters were somewhat optimized in this thesis by the DOE study, it is recommended to fully implement the twin delayed DDPG method to reduce the high sensitivity the model has with respect to the hyperparameters.

Appendix A – Electrical Hardware

Keyence IL-030



Model		IL-030
Reference distance		30 mm 1.18"
Measurement range		20 to 45 mm 0.79" to 1.77"
Light source	Type	Red semiconductor laser, wavelength: 655 nm (visible light)
	Laser class	Class 1 (FDA (CDRH) Part1040.10)[*1] Class 1 (IEC 60825-1)
	Output	220 μW
Spot diameter (at standard distance)		Approx. 200 × 750 μm
Linearity		±0.1% of F.S. (25 to 35 mm 0.98" to 1.38")[*2*3]
Repeatability		1 μm[*4]
Sampling rate		0.33/1/2/5 ms (4 levels available)
Operation status indicators		Laser emission warning indicator: Green LED, Analog range indicator: Orange LED, Reference distance indicator: Red/Green LED
Temperature characteristics		0.05% of F.S./°C[*5]

$$T_X = \frac{1}{\left(\frac{\log_{10} R_T/R_{T0}}{\mathrm{B}}\right) + \left(\frac{1}{\mathrm{T_0}}\right)}$$

Where:

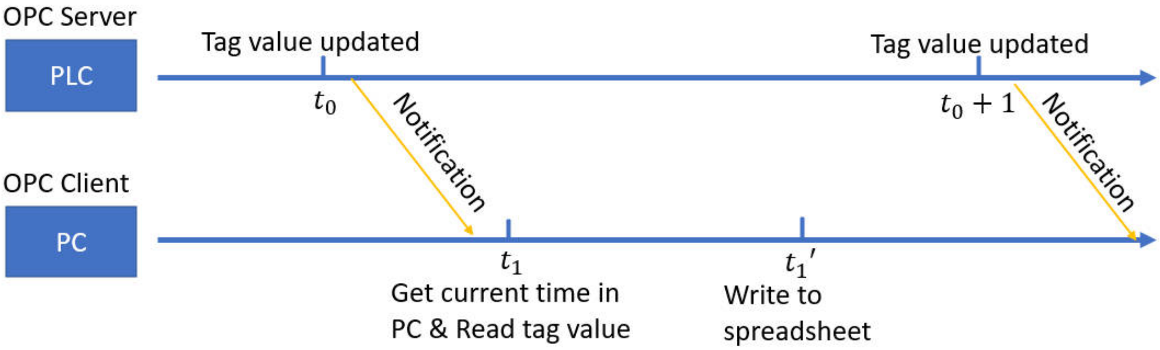
T_X is the temperature of the thermistor in Kelvin

R_T is the resistance of the thermistor

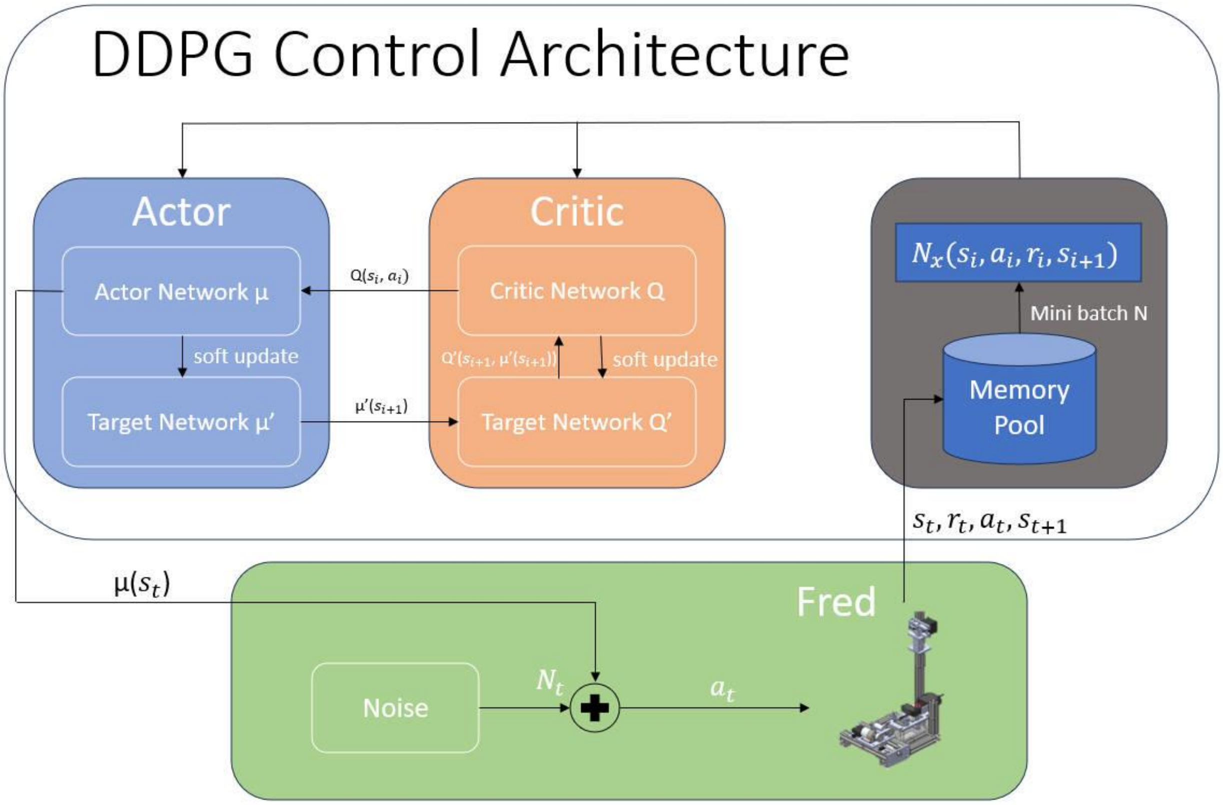
R_{T0} is the resistance of the thermistor at room temperature

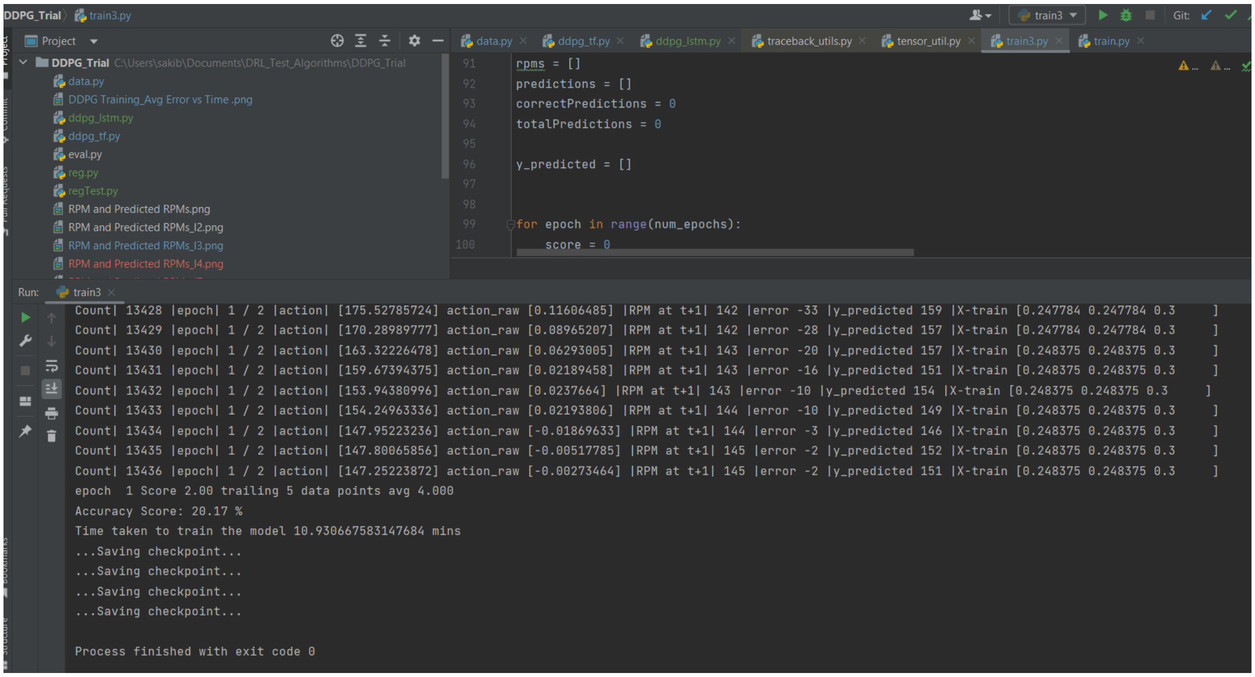
B is the beta thermistor constant

T_0 is room temperature in Kelvin



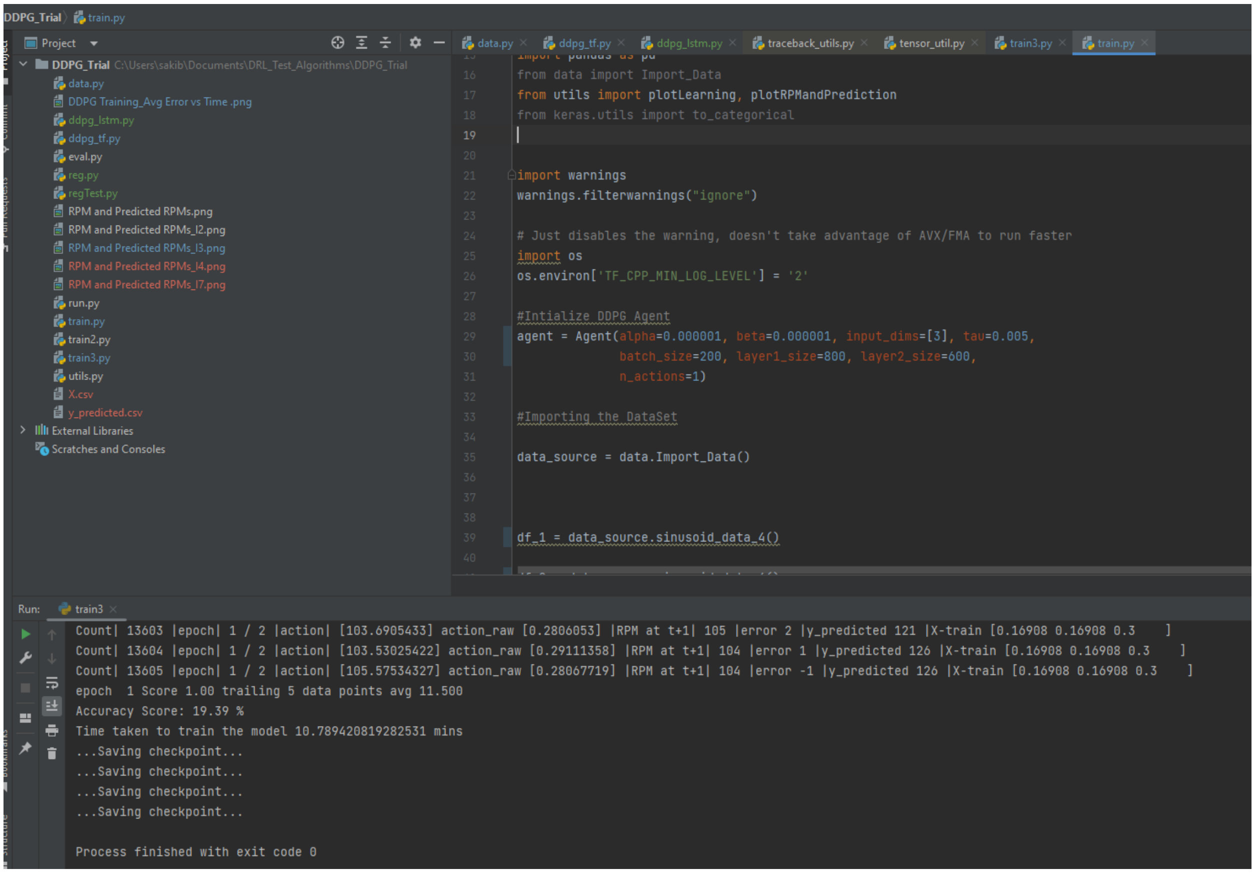
Appendix B – DDPG Code




DDPG_Trial
train3.py
train3
Git:
Project
data.py
ddpg_tf.py
ddpg_lstm.py
traceback_utils.py
tensor_util.py
train3.py
train.py
DDPG_Trial C:\Users\sakib\Documents\DRL_Test_Algorithms\DDPG_Trial
data.py
DDPG Training_Avg Error vs Time .png
ddpg_lstm.py
ddpg_tf.py
eval.py
reg.py
regTest.py
RPM and Predicted RPMs.png
RPM and Predicted RPMs_l2.png
RPM and Predicted RPMs_l3.png
RPM and Predicted RPMs_l4.png
Run:
train3

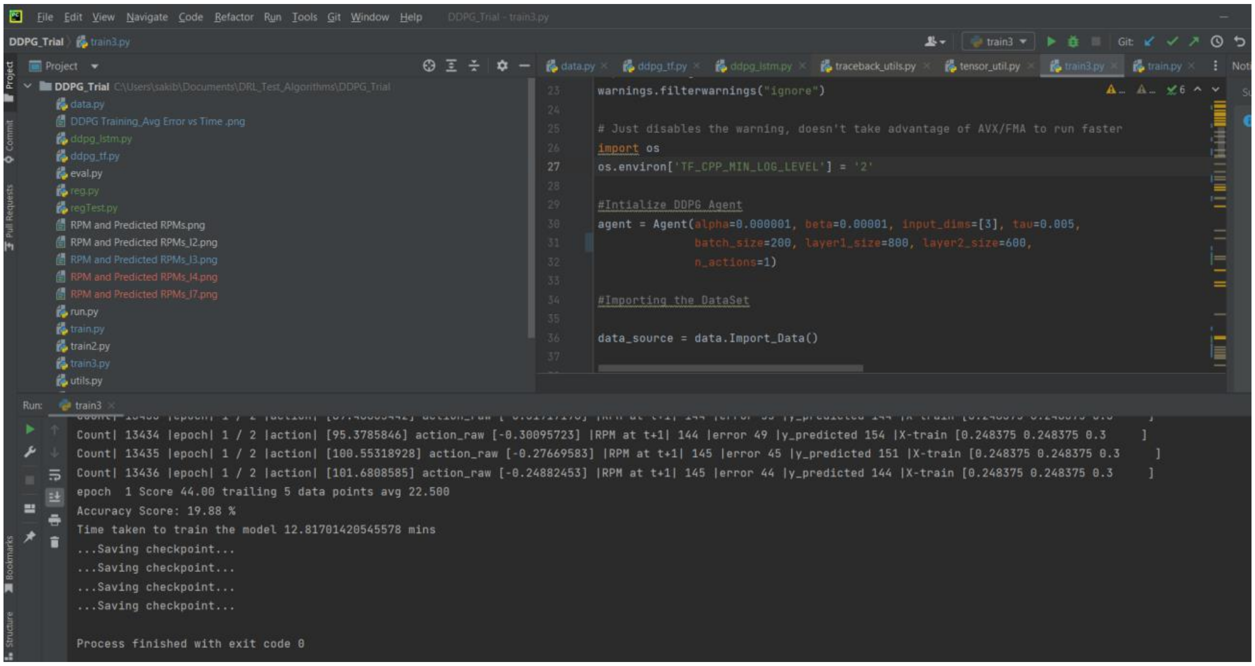
```
rpms = []
predictions = []
correctPredictions = 0
totalPredictions = 0

y_predicted = []


for epoch in range(num_epochs):
    score = 0
```

```
Count| 13428 |epoch| 1 / 2 |action| [175.52785724] action_raw [0.11606485] |RPM at t+1| 142 |error -33 |y_predicted 159 |X-train [0.247784 0.247784 0.3     ]
Count| 13429 |epoch| 1 / 2 |action| [170.28989777] action_raw [0.08965207] |RPM at t+1| 142 |error -28 |y_predicted 157 |X-train [0.247784 0.247784 0.3     ]
Count| 13430 |epoch| 1 / 2 |action| [163.32226478] action_raw [0.06293005] |RPM at t+1| 143 |error -20 |y_predicted 157 |X-train [0.248375 0.248375 0.3     ]
Count| 13431 |epoch| 1 / 2 |action| [159.67394375] action_raw [0.02189458] |RPM at t+1| 143 |error -16 |y_predicted 151 |X-train [0.248375 0.248375 0.3     ]
Count| 13432 |epoch| 1 / 2 |action| [153.94380996] action_raw [0.0237664] |RPM at t+1| 143 |error -10 |y_predicted 154 |X-train [0.248375 0.248375 0.3     ]
Count| 13433 |epoch| 1 / 2 |action| [154.24963336] action_raw [0.02193806] |RPM at t+1| 144 |error -10 |y_predicted 149 |X-train [0.248375 0.248375 0.3     ]
Count| 13434 |epoch| 1 / 2 |action| [147.95223236] action_raw [-0.01869633] |RPM at t+1| 144 |error -3 |y_predicted 146 |X-train [0.248375 0.248375 0.3     ]
Count| 13435 |epoch| 1 / 2 |action| [147.80065856] action_raw [-0.00517785] |RPM at t+1| 145 |error -2 |y_predicted 152 |X-train [0.248375 0.248375 0.3     ]
Count| 13436 |epoch| 1 / 2 |action| [147.25223872] action_raw [-0.00273464] |RPM at t+1| 145 |error -2 |y_predicted 151 |X-train [0.248375 0.248375 0.3     ]
epoch  1 Score 2.00 trailing 5 data points avg 4.000
Accuracy Score: 20.17 %
Time taken to train the model 10.930667583147684 mins
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...

Process finished with exit code 0
```



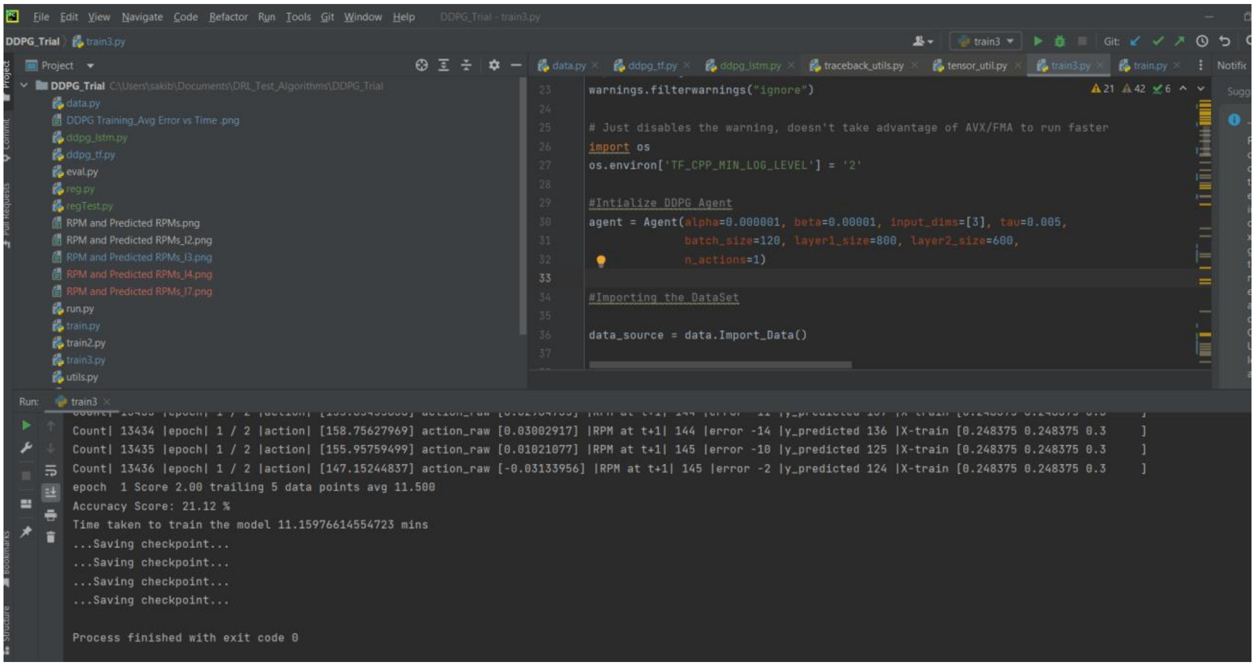
```python
import pandas as pd
from data import Import_Data
from utils import plotLearning, plotRPMandPrediction
from keras.utils import to_categorical


import warnings
warnings.filterwarnings("ignore")

# Just disables the warning, doesn't take advantage of AVX/FMA to run faster
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

#Intialize DDPG Agent
agent = Agent(alpha=0.000001, beta=0.000001, input_dims=[3], tau=0.005,
              batch_size=200, layer1_size=800, layer2_size=600,
              n_actions=1)

#Importing the DataSet

data_source = data.Import_Data()



df_1 = data_source.sinusoid_data_4()
```

```
Count| 13603 |epoch| 1 / 2 |action| [103.6905433] action_raw [0.2806053] |RPM at t+1| 105 |error 2 |y_predicted 121 |X-train [0.16908 0.16908 0.3     ]
Count| 13604 |epoch| 1 / 2 |action| [103.53025422] action_raw [0.29111358] |RPM at t+1| 104 |error 1 |y_predicted 126 |X-train [0.16908 0.16908 0.3     ]
Count| 13605 |epoch| 1 / 2 |action| [105.57534327] action_raw [0.28067719] |RPM at t+1| 104 |error -1 |y_predicted 126 |X-train [0.16908 0.16908 0.3     ]
epoch  1 Score 1.00 trailing 5 data points avg 11.500
Accuracy Score: 19.39 %
Time taken to train the model 10.789420819282531 mins
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...

Process finished with exit code 0
```


File Edit View Navigate Code Refactor Run Tools Git Window Help DDPG_Trial - train3.py
DDPG_Trial 〉 train3.py
train3
Git:
Project
DDPG_Trial C:\Users\sakib\Documents\DRL_Test_Algorithms\DDPG_Trial
data.py
DDPG Training_Avg Error vs Time .png
ddpg_lstm.py
ddpg_tf.py
eval.py
reg.py
regTest.py
RPM and Predicted RPMs.png
RPM and Predicted RPMs_l2.png
RPM and Predicted RPMs_l3.png
RPM and Predicted RPMs_l4.png
RPM and Predicted RPMs_l7.png
run.py
train.py
train2.py
train3.py
utils.py
data.py
ddpg_tf.py
ddpg_lstm.py
traceback_utils.py
tensor_util.py
train3.py
train.py
Run:
train3

```python
warnings.filterwarnings("ignore")

# Just disables the warning, doesn't take advantage of AVX/FMA to run faster
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

#Intialize DDPG Agent
agent = Agent(alpha=0.000001, beta=0.00001, input_dims=[3], tau=0.005,
              batch_size=200, layer1_size=800, layer2_size=600,
              n_actions=1)

#Importing the DataSet

data_source = data.Import_Data()
```

```
Count| 13434 |epoch| 1 / 2 |action| [95.3785846] action_raw [-0.30095723] |RPM at t+1| 144 |error 49 |y_predicted 154 |X-train [0.248375 0.248375 0.3     ]
Count| 13435 |epoch| 1 / 2 |action| [100.55318928] action_raw [-0.27669583] |RPM at t+1| 145 |error 45 |y_predicted 151 |X-train [0.248375 0.248375 0.3     ]
Count| 13436 |epoch| 1 / 2 |action| [101.6808585] action_raw [-0.24882453] |RPM at t+1| 145 |error 44 |y_predicted 144 |X-train [0.248375 0.248375 0.3     ]
epoch  1 Score 44.00 trailing 5 data points avg 22.500
Accuracy Score: 19.88 %
Time taken to train the model 12.81701420545578 mins
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...

Process finished with exit code 0
```

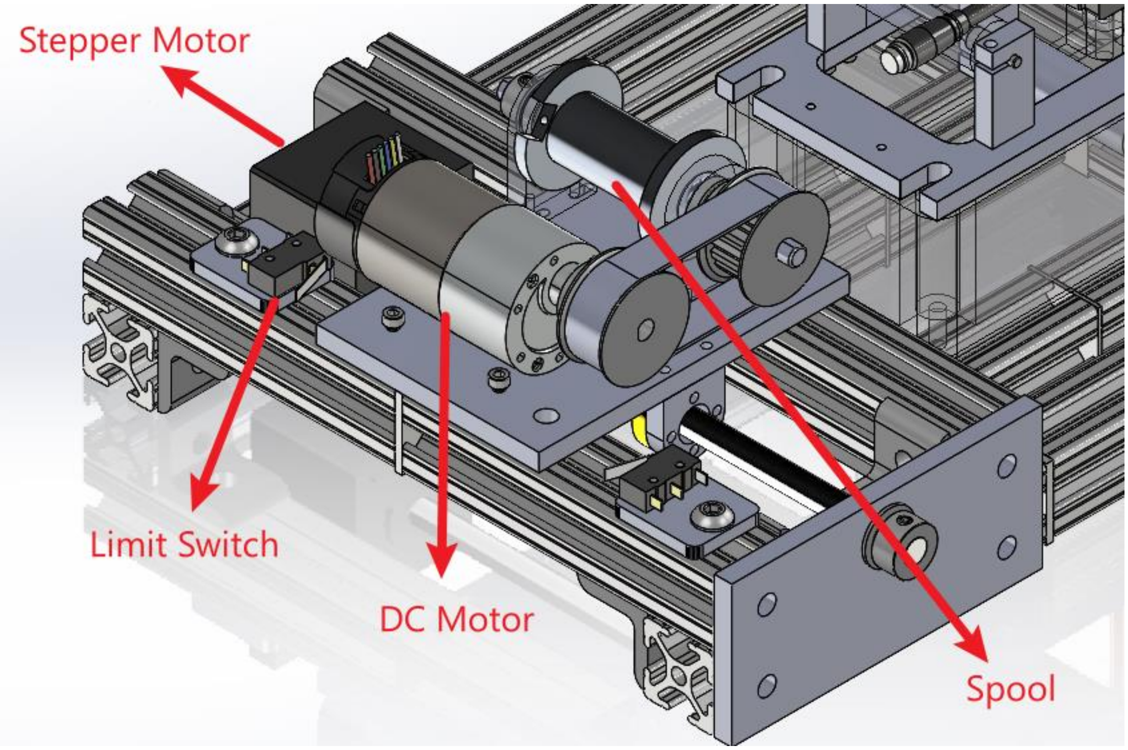

File Edit View Navigate Code Refactor Run Tools Git Window Help
DDPG_Trial - train3.py
DDPG_Trial
train3.py
Project
DDPG_Trial C:\Users\sakib\Documents\DRL_Test_Algorithms\DDPG_Trial
data.py
DDPG Training_Avg Error vs Time .png
ddpg_lstm.py
ddpg_tf.py
eval.py
reg.py
regTest.py
RPM and Predicted RPMs.png
RPM and Predicted RPMs_l2.png
RPM and Predicted RPMs_l3.png
RPM and Predicted RPMs_l4.png
RPM and Predicted RPMs_l7.png
run.py
train.py
train2.py
train3.py
utils.py
train3
Git:
data.py
ddpg_tf.py
ddpg_lstm.py
traceback_utils.py
tensor_util.py
train3.py
train.py
Notific
21
42
6
Run:
train3

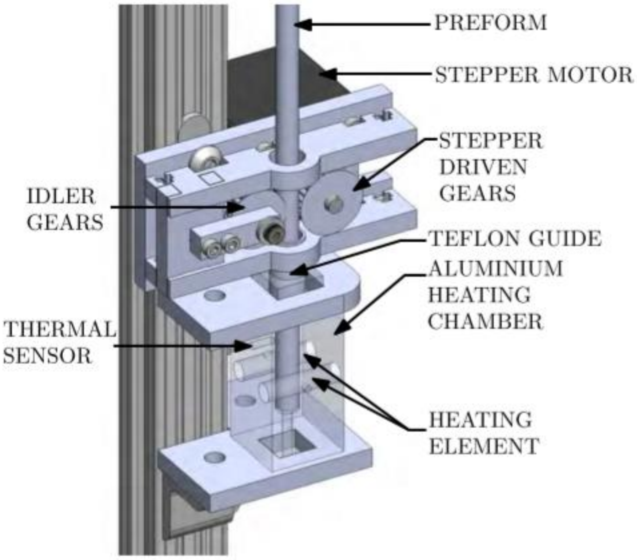
```
warnings.filterwarnings("ignore")

# Just disables the warning, doesn't take advantage of AVX/FMA to run faster
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

#Intialize DDPG Agent
agent = Agent(alpha=0.000001, beta=0.00001, input_dims=[3], tau=0.005,
              batch_size=120, layer1_size=800, layer2_size=600,
              n_actions=1)

#Importing the DataSet

data_source = data.Import_Data()
```

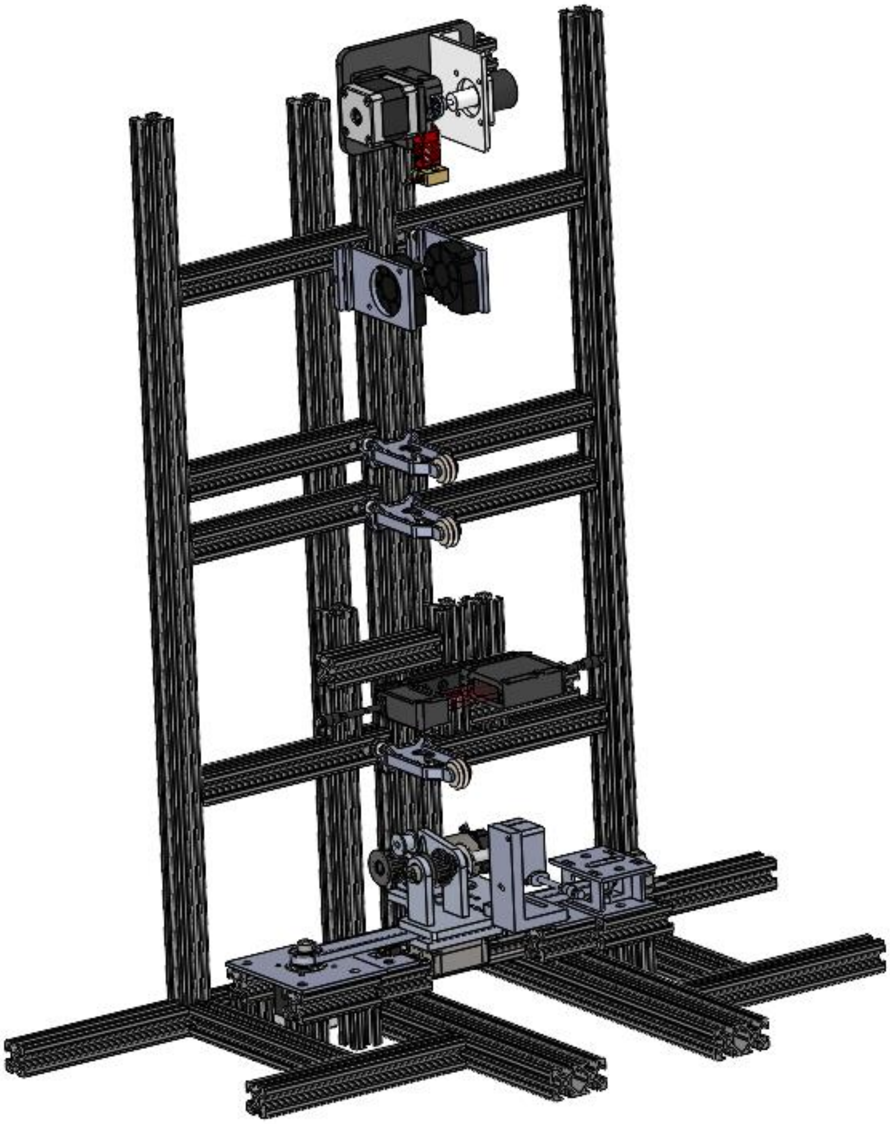
```
Count| 13434 |epoch| 1 / 2 |action| [158.75627969] action_raw [0.03002917] |RPM at t+1| 144 |error -14 |y_predicted 136 |X-train [0.248375 0.248375 0.3      ]
Count| 13435 |epoch| 1 / 2 |action| [155.95759499] action_raw [0.01021077] |RPM at t+1| 145 |error -10 |y_predicted 125 |X-train [0.248375 0.248375 0.3      ]
Count| 13436 |epoch| 1 / 2 |action| [147.15244837] action_raw [-0.03133956] |RPM at t+1| 145 |error -2 |y_predicted 124 |X-train [0.248375 0.248375 0.3      ]
epoch  1 Score 2.00 trailing 5 data points avg 11.500
Accuracy Score: 21.12 %
Time taken to train the model 11.15976614554723 mins
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...
...Saving checkpoint...

Process finished with exit code 0
```

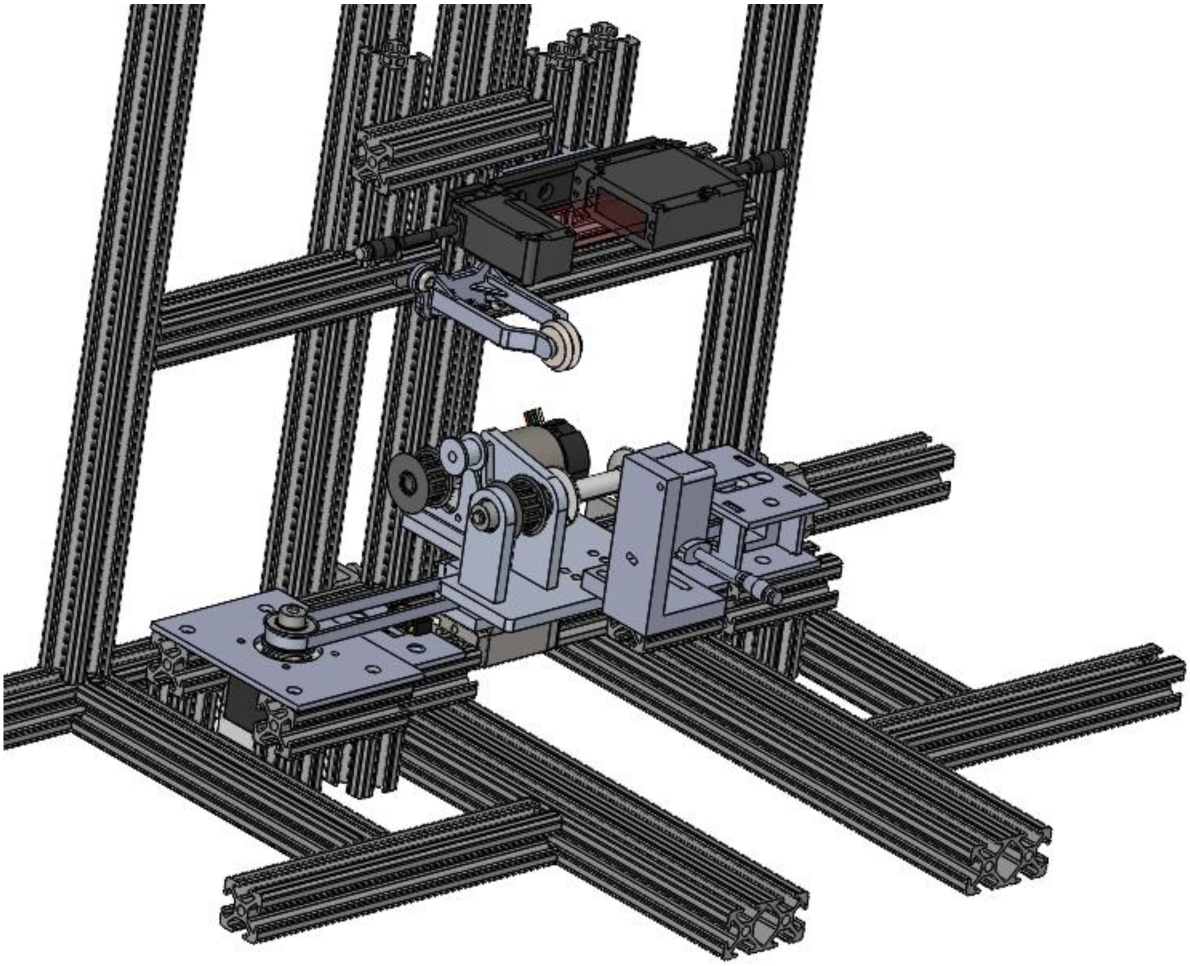
The full githhub code for the DDPG model can be found here:
https://github.com/gssakib/DRL_Test_Algorithms.git

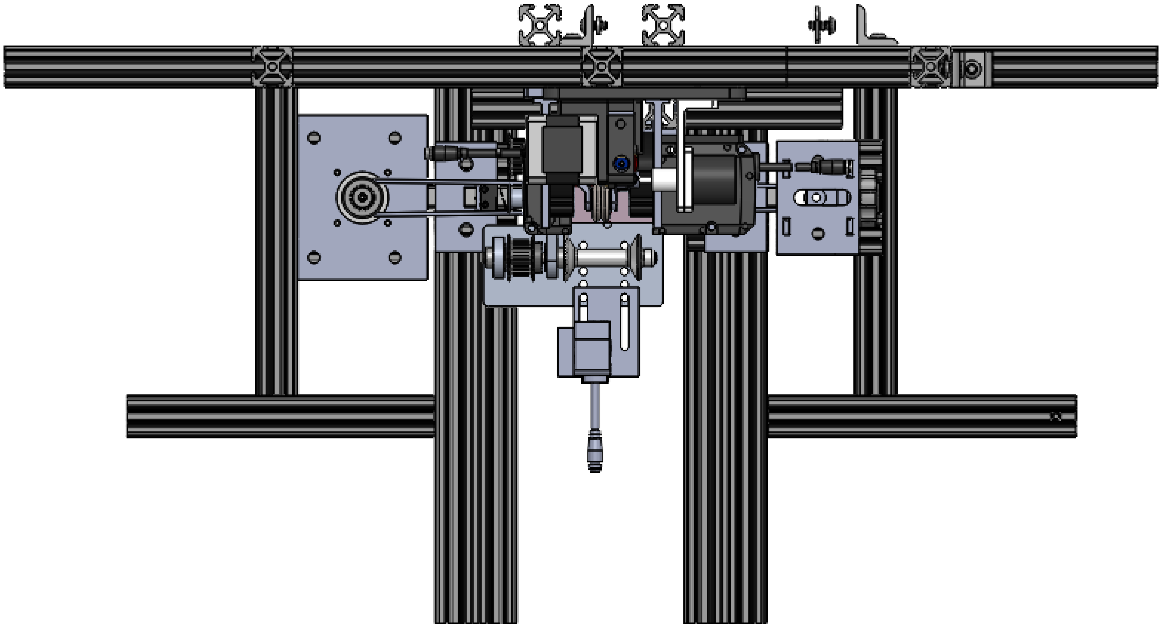
Appendix C – Mechanical Design


Stepper Motor
Limit Switch
DC Motor
Spool


PREFORM
STEPPER MOTOR
STEPPER DRIVEN GEARS
IDLER GEARS
TEFLON GUIDE
ALUMINIUM HEATING CHAMBER
THERMAL SENSOR
HEATING ELEMENT



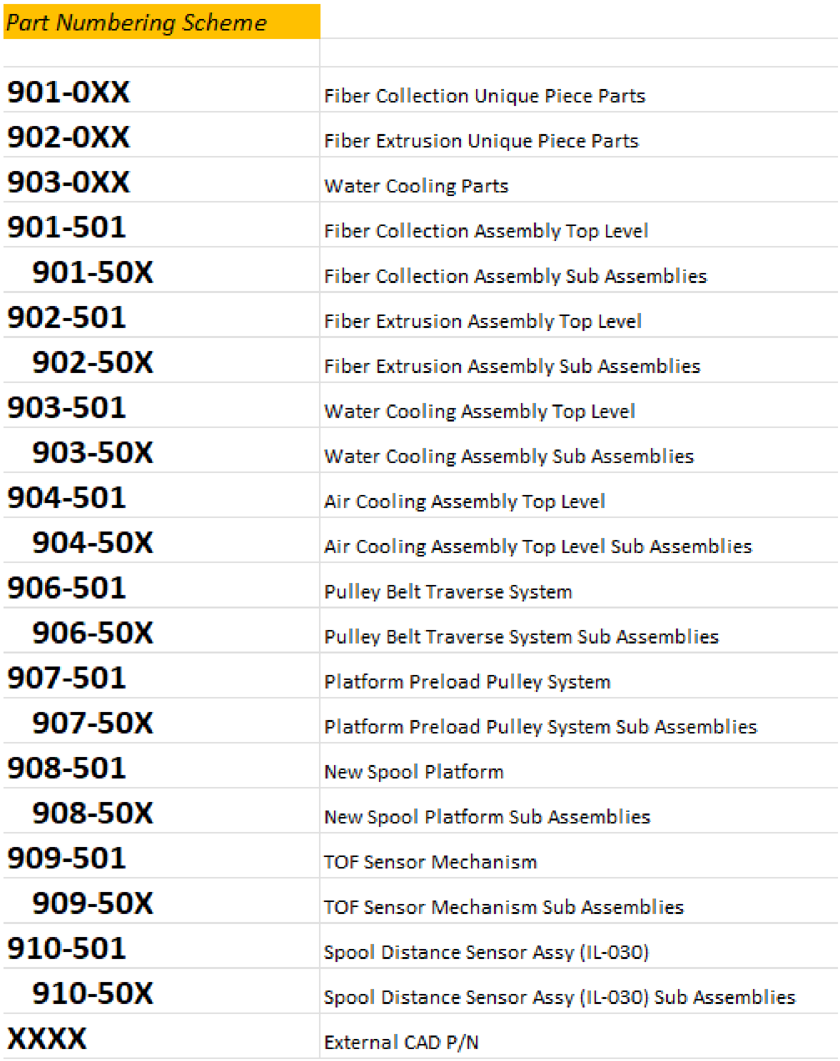




Appendix D – Symbol Definitions

Symbols	**Description**
a_t	*Action taken by agent at time, t*
μ_θ	*Policy with hyperparameter, θ*
π_θ	*Policy with hyperparameter, θ*
s_t	*State at time, t*
τ	*Trajectories*
r_t	*Individual reward at time, t*
R	*Cumulative reward*
$J(\pi_\theta)$	*Jacobian of Policy with hyperparameter, θ*
Q^*	*Optimal Action*
$V^*(s')$	*Previous Action*
$\theta^{Q'}$	*Actor Networks*
$\theta^{\mu'}$	*Critic Networks*
$\boldsymbol{D}$	*Replay Buffer*
$\boldsymbol{L}(\boldsymbol{\phi}, \boldsymbol{D})$	*Targets in the Bellman error loss functions*

Appendix E – CAD Structure and Bill of Materials



Part Numbering Scheme	
901-0XX	Fiber Collection Unique Piece Parts
902-0XX	Fiber Extrusion Unique Piece Parts
903-0XX	Water Cooling Parts
901-501	Fiber Collection Assembly Top Level
901-50X	Fiber Collection Assembly Sub Assemblies
902-501	Fiber Extrusion Assembly Top Level
902-50X	Fiber Extrusion Assembly Sub Assemblies
903-501	Water Cooling Assembly Top Level
903-50X	Water Cooling Assembly Sub Assemblies
904-501	Air Cooling Assembly Top Level
904-50X	Air Cooling Assembly Top Level Sub Assemblies
906-501	Pulley Belt Traverse System
906-50X	Pulley Belt Traverse System Sub Assemblies
907-501	Platform Preload Pulley System
907-50X	Platform Preload Pulley System Sub Assemblies
908-501	New Spool Platform
908-50X	New Spool Platform Sub Assemblies
909-501	TOF Sensor Mechanism
909-50X	TOF Sensor Mechanism Sub Assemblies
910-501	Spool Distance Sensor Assy (IL-030)
910-50X	Spool Distance Sensor Assy (IL-030) Sub Assemblies
XXXX	External CAD P/N

CAD P/N	Part Name	Owner	Checked Out By
901-501	**Fiber Collection Assembly Top Level**		
901-001	BaseScrewSupport	Gazi	
901-002	BaseMotorSupport	Gazi	
901-003	MotionBasePlate	Gazi	
901-004	SpoolRoller	Gazi	
901-005	SpoolEnd	Gazi	
901-006	SpoolMount	Gazi	
901-007	IdleExtruder	Gazi	
901-008	IdlerHolder	Gazi	
901-009	IdlerMotionStage	Gazi	
901-010	Limit Switch Holder	Gazi	
901-011	DC Motor Bracket	Gazi	
902-501	**Fiber Extrusion Assembly Top Level**		
902-001	8020_Vertical Support	Gazi	
902-002	8020_Water Bath Support	Gazi	
902-003	Preform Motor Lever	Gazi	
902-004	Preform Motor Support	Gazi	
902-005	Driver Gear	Gazi	
902-006	8020_Horizontal Support	Gazi	
902-007	IdleExtruder	Gazi	
902-008	8020_Preform Support	Gazi	
902-009	IdlerMotionStage	Gazi	
902-010	8020_Laser Sensor	Gazi	
902-011	Bronze Sleeve Bearing	Yutong	
902-012	Rotary Encoder	Yutong	
902-013	Shaft Coupling	Yutong	
902-014	Encoder Fixture	Yutong	
902-015	Encoder Support	Yutong	
902-016	8020_Laser Sensor 02	Gazi	
902-017	Heater Block	Gazi	
902-018	8020_Laser Sensor Support 02	Gazi	
902-019	Heater block 02	Gazi	
902-020	Tensioner w 6mm Ball Short Wide	Gazi	
902-021	Big Metal Pulley	Gazi	
902-022	8020_Limit_Sensor_Mount	Gazi	

903-501	**Water Cooling Assembly Top Level**	
903-001	PulleyBase	Gazi
903-002	PulleySubSide	Gazi
903-003	PulleySide	Gazi
903-004	Roller	Gazi
903-005	two piece shaft collar	Yutong
904-501	**Air Cooling Assembly Top Level**	
904-001		
904-002		
904-502	Fan Mount Assy	Gazi
904-003	FanHolder Fan Mount	Gazi
904-004	FanHolder Backing Mount	Gazi
906-501	**Pulley Belt Traverse System**	Gazi
906-502	Clamp System	Gazi
906-001	Clamp Side Plate 01	Gazi
906-002	Clamp Side Plate 02	Gazi
906-003	Motor Mount 01	Gazi
906-004	Motor Support 8020	Gazi
906-005	Motor Support 8020	Gazi
906-006	Motor Support 8021	Gazi
906-503	Tensioner System	Gazi
906-007	Idler Interface Clamp	Gazi
906-008	Idler Base Plate	Gazi
906-009	Idler Side Plate	Gazi
906-010	Tensioner Mount 8020	Gazi
906-011	Tensioner Mount 8020	Gazi
906-012	Idler Front Plate	Gazi

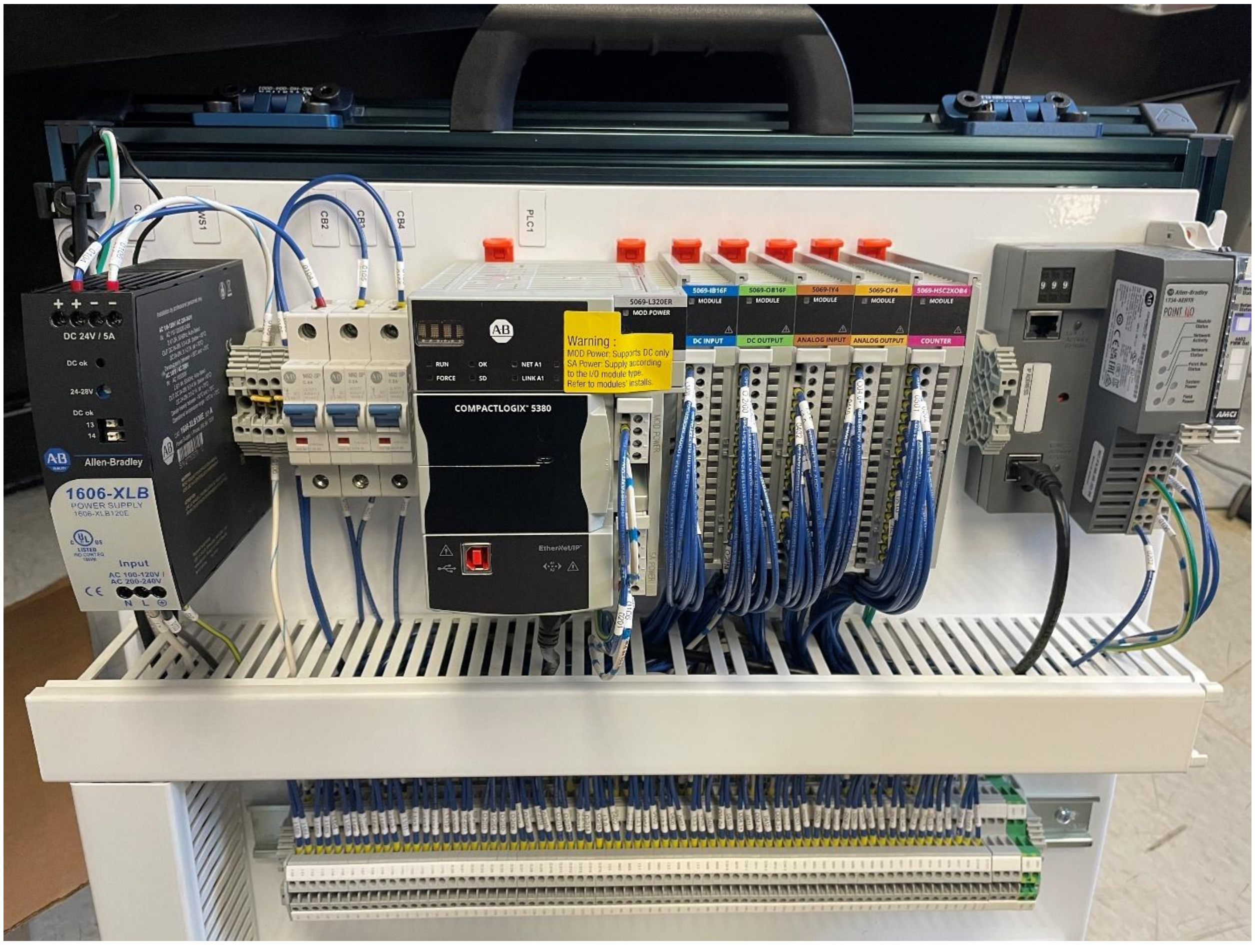
907-501	Platform Preload Pulley System	Wenhao	
907-502	Rod Slider System		
907-001	Shaft Mount		
907-002	5mm shaft ____mm		
907-503	Pulley Directional System		
907-003	Pulley Holder 1		
907-004 (902-021)	Big Metal Pulley		
907-005	Pulley Holder 2		
907-006	Fishing line		
907-007	Basket		
907-008	7gm Weights		

908-501	New Spool Platform	Wenhao	No. of Pcs	
908-001	Base Plate	waterjetted	1	
908-002	Spool Plate 1 (with tensioner)	waterjetted then milled for precise hole	1	
908-003	Spool Plate 2	waterjetted then milled for precise hole	1	
908-004	Removable Spool	3D printed	1	
908-005	Spool Rod (0.25 in)	https://www.mcmaster.com/8632T144/	1	
908-006 (901-005)	Spool Ball Bearing	https://www.mcmaster.com/57155K324	2	u
908-007	Timing Belt Pulley	https://www.mcmaster.com/57105K14/	2	
908-008	Timing Belt (Outer 8", 0.375" width, 0.2" pitch)	https://www.mcmaster.com/1679K16/	1	
908-009	Tensioner Pulley	https://a.co/d/amaCKqh	1	
908-010 (901-011)	DC Motor Mount	https://www.pololu.com/product/1084	1	
908-011	DC Motor	https://www.pololu.com/category/116/	1	
908-012	Shaft Collar	https://www.mcmaster.com/9414T6/	2	
			to buy	

909-501	TOF Sensor Mechanism	Gazi
		Gazi
909-001	TOF Base	Gazi
909-002	TOF Moving	Gazi
909-003	TOF Moving Side	Gazi
909-004	TOF Base Structure	Gazi
909-005	TOF Bulkheads	
909-006	TOF Upper Stiffner	Gazi
909-007	TOF Roller Bulkhead	Gazi

910-501	Spool Distance Sensor Assy (IL-030)	Gazi
		Gazi
910-001	Spool Sensor Mount	Gazi
IL-030-2	Keyance Distance Sensor	Gazi

Appendix F – Organized/Integrated PLC System (Allen Bradley 5380 Compact Logix Modules)



References

[1] Addanki, Satish, et al. “Review of optical fibers-introduction and applications in Fiber Lasers.” *Results in Physics*, vol. 10, Sept. 2018, pp. 743–750, https://doi.org/10.1016/j.rinp.2018.07.028.

[2] Fiber Optics Market Size, Share and Growth Report, 2030.” *Fiber Optics Market Size, Share And Growth Report, 2030*, www.grandviewresearch.com/industry-analysis/fiber-optics-market. Accessed 26 Mar. 2024.

[3] Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. “*Continuous Control with Deep Reinforcement Learning,” 2015. https://doi.org/10.48550/ARXIV.1509.02971.*

[4] Zhang, Yutong. “*Development of Process Control Framework Incorporating Deep Reinforcement Learning for Desktop Fiber Extrusion Device via PLC Implementation”, 2024.*

[5] Patrick, Keeghan J. “*A Machine Learning Approach to Improve Diameter Control in Desktop Fiber Extrusion Processes”*, 2024.

[6] Fujimoto, Scott, et al. “Addressing Function Approximation Error in Actor-Critic Methods.” *International Conference on Machine Learning*, 2018.

[7] Chakrabarti, Abhranil, and Arushi Jain. “Perfecting Optical Fibers: Deep Reinforcement Learning for Predictive Control.” Massachusetts Institute of Technology, 2023.

[8] Mahnke, Wolfgang, Stefan-Helmut Leitner, and Matthias Damm. *OPC Unified Architecture*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. https://doi.org/10.1007/978-3-540-68899-0.

[9] Othman, Mohamed. “Application of Machine Learning in Process Control in Optical Fiber Manufacturing,” 2023.

[10] Kim, Sangwoon. *“Model-Free Tracking Control of an Optical Fiber Drawing Process Using Deep Reinforcement Learning.” Massachusetts Institute of Technology, 2021. https://dspace.mit.edu/handle/1721.1/132901?show=full.*

[11] Kim, Sangwoon, David Donghyun Kim, and Brian Anthony*. “Dynamic Control of a Fiber Manufacturing Process Using Deep Reinforcement Learning.” arXiv, January 6, 2021. http://arxiv.org/abs/1911.10286.*

[12] Dhar, Shreya. “*Design and Analysis of a Desktop Fiber Manufacturing Device”*, 2022.

[13] Slocum, Alexander H. *Precision Machine Design*. Prentice-Hall, 1992.

[14] Silver, David, et al. *Deterministic Policy Gradient Algorithms*, https://doi.org/Deepmind.com.

[15] Fujimoto, Scott, Herke Van Hoof, and David Meger. “Addressing Function Approximation Error in Actor-Critic Methods.” *International Conference on Machine Learning*, 2018.