

Photocopy and Use Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission for the extensive copying of my thesis for scholarly purposes may be granted by the Dean of Graduate Studies, Dean of academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature_____

Date_____

Particle Swarm Optimization of Fuzzy Self-Tuning PI Controller
for Laser Powder-Directed Energy Deposition Process: A Simulation

by

Asa G. Monson

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in the Department of Mechanical Engineering

Idaho State University

Fall 2020

To the Graduate Faculty:

The members of the committee appointed to examine the thesis of ASA G. MONSON find it satisfactory and recommend that it be accepted.

Dr. Marco P. Schoen, Major Advisor

Dr. Anish Sebastian, Co-Advisor

Dr. Arya Ebrahimpour, GFR

Acknowledgments

I would like to thank Dr. Schoen and Tom Walters for their endless support and encouragement during the execution of this research and writing. I would also like to thank you both for the opportunities you presented me. I am eternally grateful to you for sharing your experience and for your mentorship.

I wish to thank Michael McMurtrey for his support and guidance during this project. Without your commitment to keeping this project going this thesis may never have been written. I also want to thank Randy Lloyd and Michael Heighes for their technical support. I have learned much from you and am looking forward to continuing to do so. I would also like to thank Zachary Thompson for all the knowledge he has given me and for allowing me to bounce ideas around.

I also want to thank my father Brett Monson for always being proud of me and for his unwavering encouragement these past seven years.

Dedication

I would like to dedicate this work to my wife, Mary; and to my two wonderful children, Eric, and Annabelle, without whom I could not have achieved all that I have. You supported me completely through these challenging years and I am looking forward to our next adventure.

Table of Contents

| | |
|--|-----|
| List of Figures | vii |
| List of Tables | ix |
| Abstract | x |
| Nomenclature | xi |
| Chapter One: Introduction | 1 |
| 1.1 Metal Additive Manufacturing Techniques | 1 |
| 1.1.1 Powder Bed Fusion and Directed Energy Deposition | 1 |
| 1.1.2 LENS® | 2 |
| Chapter Two: Objective and Theory | 5 |
| 2.1 Objective and Justification | 5 |
| 2.1.1 Justification for Research | 5 |
| 2.1.2 Research Objective | 5 |
| 2.2 Applied Theory | 6 |
| 2.2.1 System Identification | 6 |
| 2.2.2 Process Model | 7 |
| 2.2.3 Fuzzy Logic Control | 9 |
| 2.2.4 PID Controller | 12 |
| 2.2.5 Fuzzy Self-Tuning PID Controller | 13 |
| 2.2.6 Particle Swarm Optimization | 14 |
| Chapter Three: Experimental Method | 17 |
| 3.1 Experimental Data Collection | 17 |
| 3.1.1 Hardware | 17 |
| 3.1.2 Software | 19 |
| 3.1.3 Image Processing | 20 |
| 3.2 System Identification Experiment | 22 |
| 3.2.1 Experiment Parameters | 22 |
| 3.2.2 Experimental Data | 22 |
| 3.3 System Identification | 24 |
| 3.3.1 Dynamic Model | 24 |
| 3.3.2 Stability | 27 |

| | |
|---|----|
| 3.4 Controller Design | 30 |
| 3.4.1 Fuzzy Inference Engine | 30 |
| 3.4.2 PI Parameters | 35 |
| Chapter Four: Simulation and Conclusion..... | 37 |
| 4.1 Simulation | 37 |
| 4.1.1 Simulation Setup..... | 37 |
| 4.1.2 Results | 38 |
| 4.1.3 Discussion..... | 40 |
| 4.2 Conclusion..... | 42 |
| References | 43 |
| Appendix A – Basler acA720-290gm Camera Calibration | 46 |
| Appendix B – Threshold Value Determination | 48 |
| Appendix C – LabVIEW Documentation..... | 54 |
| Appendix D – MATLAB® Scripts..... | 64 |

List of Figures

| | |
|---|----|
| Figure 1.1 LENS® Diagram [4] | 2 |
| Figure 2.1 Fuzzy Logic Controller..... | 10 |
| Figure 2.2 Triangular Membership Function | 11 |
| Figure 2.3 PID Controller Structure..... | 12 |
| Figure 2.4 Fuzzy Self-Tuning PID Controller Block Diagram..... | 14 |
| Figure 3.1 Schematic Overview of Data Acquisition System | 18 |
| Figure 3.2 NI cRIO [22]..... | 18 |
| Figure 3.3 OPTOMECH LENS® 500 [23] | 18 |
| Figure 3.4 LabVIEW Front Panel | 20 |
| Figure 3.5 Image Processing Algorithm | 21 |
| Figure 3.6 Layer 1 – 4..... | 23 |
| Figure 3.7 Layer 5 - 8..... | 23 |
| Figure 3.8 Layer 9 – 10..... | 24 |
| Figure 3.9 System Identification GUI..... | 25 |
| Figure 3.10 Static Gain and Time Constant by Layer..... | 26 |
| Figure 3.11 Pole-Zero Map (Layer 1)..... | 28 |
| Figure 3.12 Root Locus (Layer 1)..... | 28 |
| Figure 3.13 Step Response showing Sensitivity to Gain | 29 |
| Figure 3.14 Bode Diagram (Layer 1)..... | 29 |
| Figure 3.15 Step Response with Increasing Gain Values | 30 |
| Figure 3.16 e Membership Functions | 32 |
| Figure 3.17 e' Membership Functions | 32 |
| Figure 3.18 K_p Membership Functions | 33 |
| Figure 3.19 K_i Membership Functions | 33 |
| Figure 3.20 Initial FIS Validation | 34 |
| Figure 3.21 Tuned FIS Validation | 34 |
| Figure 3.22 Step Response..... | 35 |
| Figure 3.23 SIMULINK® Block Diagram of FST-PI Controller..... | 36 |
| Figure 4.1 Simulation SIMULINK® Block Diagram | 37 |

| | |
|--|----|
| Figure 4.2 Layer Step Response | 40 |
| Figure A.1 Millimeter Ruler (Keyence)..... | 46 |
| Figure A.2 Millimeter Ruler (Basler ace) | 46 |
| Figure A.3 Calibration | 47 |
| Figure B.1 Threshold Value Determination Runs | 48 |
| Figure B.2 200 W Track | 49 |
| Figure B.3 250 W Track | 49 |
| Figure B.4 300 W Track | 50 |
| Figure B.5 350 W Track | 50 |
| Figure B.6 400 W Track | 51 |
| Figure C.1 Host PC Main VI | 54 |
| Figure C.2 Host PC Main VI Documentation..... | 55 |
| Figure C.3 Build Waveform VI | 56 |
| Figure C.4 Build Image Buffer VI..... | 57 |
| Figure C.5 Create Data File VI..... | 57 |
| Figure C.6 Write Data File VI | 58 |
| Figure C.7 Save Images to Disk VI | 58 |
| Figure C.8 cRIO Main VI | 59 |
| Figure C.9 cRIO Main VI (2) | 60 |
| Figure C.10 cRIO Main VI Documentation | 61 |
| Figure C.11 Enable Laser VI | 62 |
| Figure C.12 Laser On/Off VI..... | 62 |
| Figure C.13 Reset Laser VI | 63 |

List of Tables

| | |
|---|----|
| Table 2.1 Effect of Increasing PID gains on the Step Response..... | 13 |
| Table 3.2 Transfer Function Parameters for different Layers with Padè Approximation..... | 26 |
| Table 3.3 Fuzzy Rule Base..... | 31 |
| Table 4.1 Rise Time and % Overshoot | 38 |
| Table B.1 Threshold Value Determination Measurements..... | 53 |

Particle Swarm Optimization of Fuzzy Self-Tuning PI Controller
for Laser Powder-Directed Energy Deposition Process: A Simulation

Thesis Abstract--Idaho State University (2020)

The laser powder-directed energy deposition process is non-linear, and the dynamic response of the system changes with each layer. Current systems require considerable manual tuning and process knowledge to achieve quality production. To take advantage of the technology, it is necessary to develop in-situ process monitoring and closed-loop control.

In this research a single-input, single-output fuzzy self-tuning PI controller was developed to ensure a constant melt pool diameter during the build. The laser power was selected as the control output and the diameter as the control input variable. A thin wall part was built with ten layers while varying the laser power using a pseudo random binary sequence.

The controller was designed, and its performance simulated using MATLAB/SIMULINK® and compared to a simple PI controller. The results suggest that the fuzzy self-tuning proportional and integral controller will result in a more consistent melt pool diameter during the build process.

Key Words:

additive manufacturing, directed energy deposition, LENS®, fuzzy logic, PI control, self-tuning, system identification, particle swarm, optimization.

Nomenclature

| | |
|-----------|---|
| AM | Additive Manufacturing |
| PBF | Powder bed Fusion |
| DED | Directed Energy Deposition |
| LENS® | Laser Engineered Net Shaping |
| PID | Proportional, Integral, and Derivative |
| PI | Proportional and Integral |
| FST-PI | Fuzzy Self-Tuning Proportional and Integral |
| PSO | Particle Swarm Optimization |
| I/O | Input, Output |
| GUI | Graphical User Interface |
| LTI | Linear Time Invariant |
| SISO | Single input Single Output |
| e | Process Error |
| \dot{e} | Process Error Rate of Change |
| mm | millimeter |
| in | inch |
| ipm | Inch per minute |
| rpm | Revolutions per minute |
| ppm | Parts per million |
| psi | Pounds per square inch |
| fps | Frames per second |
| VDC | Voltage, Direct Current |
| W | Watts |
| dB | Decibels |
| ms | millisecond |

Chapter One: Introduction

Chapter One will discuss metal additive manufacturing (AM) processes and current research in the field. The chapter will also discuss current research in the development of control of the process used in this work.

1.1 Metal Additive Manufacturing Techniques

Section 1 covers the two sub-categories of metal AM and their applications. Section 1 will also discuss in more detail the process of laser powder-directed energy deposition.

1.1.1 Powder Bed Fusion and Directed Energy Deposition

Metal AM enables the fabrication of complex parts by adding material in a layer-wise fashion. It offers reduced lead production time, decreased buy-to-fly ratio, which refers to the weight of raw material compared to the weight of the final part, and repair and remanufacturing of high value components [1]. Various AM techniques that utilize metal have been developed in recent years. Powder bed fusion (PBF) and directed energy deposition (DED) are the two well-known sub-categories of metal AM processes [2]. According to ISO/ASTM 52900:2015 DED is defined as “an additive manufacturing process in which focused thermal energy is used to fuse materials by melting as they are being deposited” [3]. The DED process consists of feeding metallic powder, wire, or both into a melt pool that is formed utilizing a high energy heat source such as a laser, arc, or electron beam. PBF works differently, a bed of powder is selectively sintered or melted using a laser or electron beam heat source. Each method has their own pros and cons. PBF can be used to build overly complex parts with internal features. The surface finish of PBF is also superior to DED. Though there are advantages to using DED over PBF. DED has an unlimited buildup size,

higher deposition rates, and the ability to make functionally graded materials which are materials in which the composition and structure gradually change resulting in a corresponding change in the properties of the material. Additionally, the ability to repair worn or damaged high value components makes DED appealing to select industries.

1.1.2 LENS®

Laser powder-directed energy deposition also known by the proprietary name LENS® uses computer-controlled lasers that weld air-blown streams of metallic powder into custom parts and manufacturing molds [4]. Figure 1.1 shows the laser engineered net shaping diagram. Metallic powder is delivered to the melt-pool through four nozzles equally spaced in a circle angled toward the center. The metallic powder is then melted by a high energy laser to form a part. Argon is used as a shield gas to clear the workspace of any debris and to prevent oxidation. As can be seen in the diagram the workplace is free to move in the x and y axis, while the print head moves in the z axis.

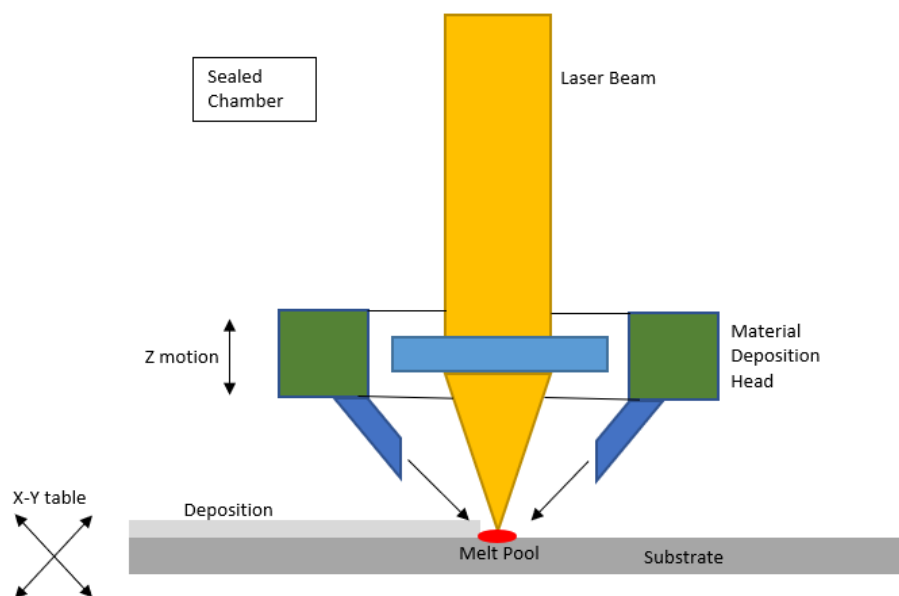


Figure 1.1 LENS® Diagram [4]

LENS® has been used in various studies for a variety of applications including the work presented here. For instance, Avila et al. [5] showed that the LENS® technique can mimic bone properties and increase early-stage host tissue and implant integration for biomedical applications with titanium. Reichardt et al. [6] used laser metal deposition to additively manufacture functionally graded Ti-6Al-4V to 304L stainless steel components with a vanadium interlayer. Wilson et al. [7] showed the effectiveness of the laser powder feeding process to repair and remanufacture a damaged turbine blade.

In laser powder-directed energy deposition the main variables such as powder flow rate, laser power, layer thickness, travel speed, and step-over value affect the quality of deposition and stability of the process [8]. To achieve consistency and stability these parameters need to be optimized. However, due to disturbances such as change in the thermal conduction, thermal conductivity, powder flowrate, or laser power, the implementation of an in-situ sensing and control system is required [8].

Akbari and Kovacevic [9] developed an adaptive PI controller with layer-dependent control gains to ensure a constant melt pool width through the entire build. They showed that by adjusting the laser power to maintain melt pool width resulted in a more uniform and finer microstructure. Tang et al. [10] developed a method of variable powder flow rate control to adjust the powder flow rate to maintain a uniform powder deposition per unit length even when disturbances occur. Since the powder flow rate at the nozzle cannot be directly measured it was estimated using a system model. The experimental results show that their method is successful in maintaining a uniform track morphology and recommend combining this with melt pool control. Bi et al. [11] used infrared (IR) temperature signals from a single wall deposition in order to develop a feedback system in the laser powder feeding process. They maintained the melt pool temperature as a control input

by adjusting the laser power. The results show a nearly constant melt pool size and homogeneous microstructure. Farshidianfar [12] developed a feedback PID controller using an infrared camera in the laser powder cladding process. By adjusting the travel speed, they were able to control the cooling rate during the process. The closed loop controller could produce a relatively consistent microstructure.

Existing work related to closed loop control of the laser-directed energy deposition process have focused on controllers with fixed parameters. However, due to the existence of disturbances the process is highly non-linear, for example, the relation between the laser power and melt pool size is non-linear throughout the entire build. In this work, a fuzzy self-tuning PI controller is proposed to provide more accurate and reliable control. Using experimental data, a process model is developed, and a controller designed. The controller is then built and simulated using MATLAB® and Simulink®.

Chapter Two: Objective and Theory

Chapter Two discusses the objectives of the research as well as the justification. Chapter Two will also discuss the theory applied in this research including the system identification and the development of a fuzzy self-tuning PID controller utilizing particle swarm optimization.

2.1 Objective and Justification

Section 2.1 discusses the objective and justifications for the research.

2.1.1 Justification for Research

As of 2019, the global additive manufacturing industry is estimated to be worth over \$9 billion dollars [13]. While this is still a fraction of the wider \$12 trillion manufacturing industry, the technology is evolving rapidly, creating many opportunities for the industry. Before additive manufacturing can become a larger fraction of this industry there are some significant shortcomings that need to be overcome. Such as, no critical structural components can be produced feasibly due to a lack of consistency in part quality, each machine and each material are different, and every part produced is different with variations in material properties and defects. To address these issues techniques for in-situ process monitoring and real-time control need to be researched and developed.

2.1.2 Research Objective

The objective of this research is to develop a fuzzy self-tuning PID controller to maintain a constant melt pool size during the laser powder-directed energy deposition process. It has been shown in previous work, discussed in Chapter One, that such a controller results in more uniform material properties and part consistency. The aim of the work was to investigate a self-tuning controller to

better address disturbances in the process, such as change in the thermal conduction, thermal conductivity, powder flow rate, or laser power.

2.2 Applied Theory

Section 2.2 discusses the theory applied during this research.

2.2.1 System Identification

System identification is a methodology for building mathematical models of dynamic systems using measurements of the system's input and output signals. There are four steps in this process; measure input and output signals from the system in time or frequency domain, select a model structure, apply an estimation method to estimate value for the adjustable parameters in the candidate model structure, and then evaluate the estimated model to see if the model is adequate for the application.

System identification uses input and output signals measured from a system to estimate the values of adjustable parameters in a given model structure. A good model depends on the quality of the measured data. Good experimental design ensures that the right variables are measured accurately to capture the system dynamics. The experiment should use inputs that adequately excite the system dynamics. A single step response is not sufficient. Data should be measured long enough to capture the time constants. Data acquisition systems should have a good signal-to-noise ratio. Lastly, measure data at appropriate sampling intervals or frequency resolution.

A model structure is a mathematical relationship between input and output variables that contain unknown parameters. Transfer functions, state-space equations, and polynomial models are three examples of model types. Equation 2.1 represents a simple model structure where a and b are adjustable parameters.

$$y(k) + ay(k - 1) = bu(k) \quad (2.1)$$

Select a model that can reproduce measured data and is as simple as possible. Various structures should be tried and compared to determine the most accurate one. This approach to modeling is called black-box modeling. For this research, a process model was selected and is discussed in Section 2.2.2. After developing a model, the quality of the model should be evaluated. The model response should be compared to data that was not used in the initial system identification process. A model may be a good fit for one set of data and less so for another. The validation of the model with new data verifies the accuracy of the dynamic model.

2.2.2 Process Model

A process model was chosen because they are simple, support transport delay estimation, and the model coefficients have an easy interpretation as poles and zeros [14]. The structure of a process model is a continuous-time transfer function that describes the dynamics of a linear system in terms of one or more of the following: static gain K_p , one or more time constants T_{pk} , for complex poles the time constant is T_ω and is equal to the inverse of the natural frequency and the damping coefficient ξ , process zero T_z , time delay T_d , or possible enforced integration. Different model structures can be created by varying the number of poles, adding an integrator, or by adding or removing a time delay or a zero. Specify a first, second, or third order system, and the poles can be real or complex. The model structure used in the research is shown in Equation (2.2):

$$G(s) = \frac{K_p}{1 + sT_p} e^{-sT_d} \quad (2.2)$$

such that,

$$Y(s) = G(s)U(s) + E(s) \quad (2.3)$$

where $Y(s)$, $U(s)$, and $E(s)$ represent the Laplace transform of the output, input, and output error, respectively. The output error, $e(t)$, is white Gaussian noise with variance λ . A disturbance model $H(s)$ can also be added such that.

$$Y(s) = G(s)U(s) + H(s)E(s) \quad (2.4)$$

The development of the process model for the SISO system discussed in the research will be discussed in Chapter Three.

In systems with time delay it is necessary to create a sensitivity of the controller to the amount of delay represented in the system. A common workaround consists of replacing delays with their Padè Approximation given by Equation (2.5). The Padè Approximation uses a series expansion of the transcendental function e^{-sT} and matches as many coefficients as possible with a series expansion of a rational function of specified order [15].

$$e^{-sl} \cong \frac{N_p(sT_d)}{D_p(sT_d)} \quad (2.5)$$

Where,

$$N_p = \sum_{k=0}^p \frac{(2p-k)!}{k! (p-k)!} (-sT_d)^k \quad (2.6)$$

$$D_p = \sum_{k=0}^p \frac{(2p-k)!}{k! (p-k)!} (-sT_d)^k \quad (2.7)$$

p is the order of the approximation. A third order approximation will give:

$$\frac{N_3(sL)}{D_3(sL)} = \frac{-T_d^3 s^2 + 12T_d^2 s^2 - 60T_d s + 120}{T_d^3 s^2 + 12T_d^2 s^2 + 60T_d s + 120} \quad (2.8)$$

When applying a 3rd order Padè Approximation to the process model in Equation (2.2) the system will become Equation (2.9).

$$G(s) = \frac{K_p}{1 + sT_p} e^{-sT_d} \left[\frac{-T_d^3 s^2 + 12T_d^2 s^2 - 60T_d s + 120}{T_d^3 s^2 + 12T_d^2 s^2 + 60T_d s + 120} \right] \quad (2.9)$$

2.2.3 Fuzzy Logic Control

Fuzzy logic control is a heuristic approach that easily embeds the knowledge and key elements of human thinking in the design of nonlinear controllers [16]. By applying fuzzy control, qualitative and heuristic considerations which cannot be handled by conventional control theory can be used in a systematic form [17]. Fuzzy logic controllers do not need an accurate model, can handle imprecise inputs, nonlinearity, and disturbances in the system.

The structure of a fuzzy logic controller is shown in Figure 2.1. Utilizing the Mamdani method the fuzzifier converts the “crisp” input values into fuzzy values. The fuzzy knowledge base stores the knowledge about all the input-output fuzzy relationships and has the membership functions which define the input variables to the fuzzy rule base and output variables to the plant. The fuzzy rule base stores the knowledge about the operation of the process. The inference engine acts as a kernel, simulating human decisions by performing approximate reasoning. Last, the defuzzifier converts the fuzzy value into a “crisp” value.

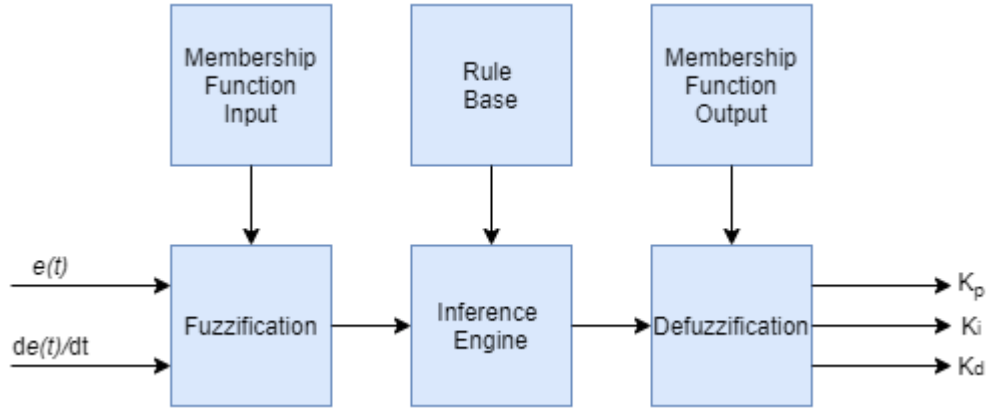


Figure 2.1 Fuzzy Logic Controller

The first step in the fuzzy controller design procedure is to define the input and output variables of the fuzzy controller. Typically, the variables chosen are the states of the controlled system, their errors, error variation, and/or error accumulation [18]. Then, the fuzzy rule base and membership functions are configured. The fuzzy controller rules are usually formulated in linguistic terms [18]. Thus, the use of linguistic variables and fuzzy sets implies the fuzzification procedure, that is, the mapping of the input variables into suitable linguistic values. Rule evaluation or decision-making infers, using an inference engine, the fuzzy control action from the knowledge of the fuzzy rules and the linguistic variable definition [18]. The output of a fuzzy controller is a fuzzy set which needs to be defuzzified. Defuzzification is the conversion of the inferred fuzzy result to a nonfuzzy or “crisp” control action, that better represents the fuzzy one. This last step obtains the crisp value for the controller output $u(k)$. These steps can be implemented online or offline. For the development of an adaptive controller they are best implemented online.

A membership function for a fuzzy set A on the universe of discourse X is defined as $\mu_A: X \rightarrow [0,1]$, where each element of X is mapped to a value between 0 and 1. This value is called the membership value or the degree of membership and quantifies the grade of membership of the element in X to the fuzzy set A . Membership functions are used to graphically represent a fuzzy set. The x-axis represents the universe of discourse and the y-axis represents the degree of membership in the $[0, 1]$ interval. A triangular membership function is defined by a lower limit a , and upper limit b , and a value m , where $a < m < b$.

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{m-a}, & a < x \leq m \\ \frac{b-x}{b-m}, & m < x < b \\ 0, & x \geq b \end{cases} \quad (2.10)$$

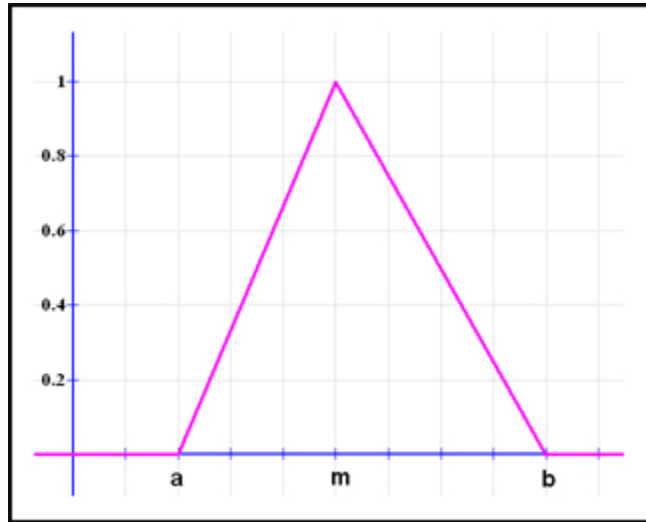


Figure 2.2 Triangular Membership Function

2.2.4 PID Controller

One form of controller widely used in industrial process control is the three-term, PID controller [15]. This controller has a transfer function shown in Equation (2.11).

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (2.11)$$

The three-term controller is called a PID controller because it contains a proportional gain (K_p), an integral gain (K_i), and a derivative gain (K_d), hence the name. The structure of a PID controller is shown in Figure (2.3), where $r(t)$ is the reference, $e(t)$ is the error, $u(t)$ is the controller output, and $y(t)$ is the system output.

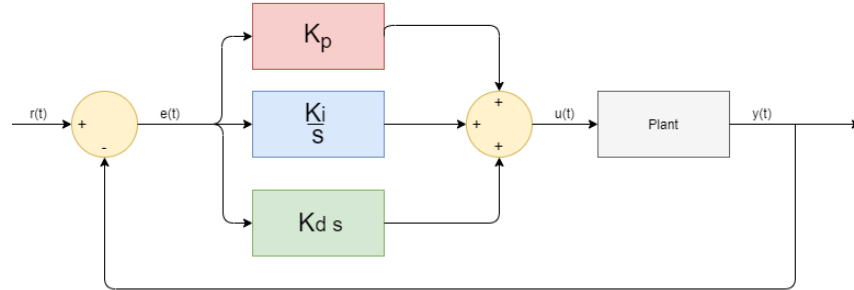


Figure 2.3 PID Controller Structure

The equation for the output in the time domain is Equation (2.12) below,

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2.12)$$

where $e(t)$ is the error shown in Equation (2.13).

$$e(t) = r(t) - y(t) \quad (2.13)$$

If $K_d = 0$, then we have a proportional plus integral (PI) controller.

$$G_c(s) = K_p + \frac{K_i}{s} \quad (2.14)$$

When $K_i = 0$, we have

$$G_c(s) = K_p + K_d s \quad (2.15)$$

which is called a proportional plus derivative (PD) controller.

PID controllers are commonly tuned manually, where the PID control gains are obtained by trial-and-error with minimal analytic analysis used step responses obtained from simulation or by simply testing on the system and determining the gains based on observation and experience [18]. A more analytic method is the Ziegler-Nichols tuning method and has several variations.

One manual approach to manual tuning is to set $K_i = 0$ and $K_d = 0$. Then slowly increase the gain K_p until the out-put of the closed-loop system oscillates just on the edge of instability. Once the value of K_p is found that brings the closed-loop system to the edge of stability, then reduce the value of gain K_p to achieve what is known as the quarter amplitude decay [18]. That is, the amplitude of the closed-loop response is reduced to one-fourth of the maximum value in one oscillatory period. A general recommendation is to reduce the proportional gain by one-half [18]. Then increase K_i and K_d manually to achieve a desired step response. Table 2.1 shows in general terms the effect of increasing K_i and K_d .

Table 2.1 Effect of Increasing PID gains on the Step Response

| PID Gain | Percent Overshoot | Settling Time | Stead-State Error |
|------------------|-------------------|----------------|-------------------------|
| Increasing K_p | Increases | Minimal Impact | Decreases |
| Increasing K_i | Increases | Increases | Zero steady-state error |
| Increasing K_d | Decreases | Decreases | No impact |

2.2.5 Fuzzy Self-Tuning PID Controller

A fuzzy self-tuning PID controller is a controller that is based on fuzzy logic with a PID structure [19]. A block diagram showing the basic structure of a fuzzy self-tuning PID controller is shown in Figure (2.4).

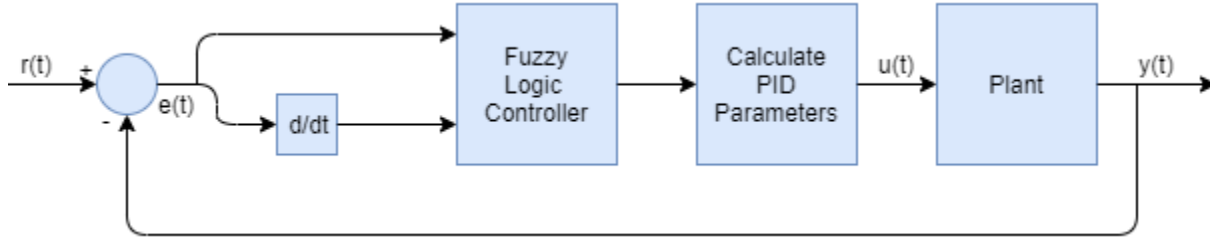


Figure 2.4 Fuzzy Self-Tuning PID Controller Block Diagram

The fuzzy self-tuning PID controller shows enormous advantages in processes with intensive nonlinearity [20]. PID gain parameters are adjusted online by evaluating the error $e(t)$ and the rate of change of the error $\dot{e}(t)$. The output of the fuzzy controller is $u(k)$.

$$u(k) = K_p e(k) + K_i \sum_{i=1}^k e(i) + K_d [e(k) - e(k-1)] \quad (2.16)$$

The PID parameters can be updated online using Equations (2.17), (2.18), and (2.19), respectively [21].

$$K_p = K_{p0} + \Delta K_p \quad (2.17)$$

$$K_i = K_{i0} + \Delta K_i \quad (2.18)$$

$$K_d = K_{d0} + \Delta K_d \quad (2.19)$$

Where K_p , K_i , and K_d are the proportional, integral, and derivative gains, respectively. K_{p0} , K_{i0} , and K_{d0} are the initial parameters calculated by conventional methods, and ΔK_p , ΔK_i , and ΔK_{d0} are the corrections.

2.2.6 Particle Swarm Optimization

Particle swarm optimization (PSO) is an evolutionary heuristic algorithm introduced by Kennedy and Eberhart in 1995 [22, 23]. It has been found to be robust in solving continuous nonlinear optimization problems [22]. PSO may be used to find optimal or near optimal solutions in a big search space.

PSO is an evolutionary computational technique based on the movement and intelligence of swarms. PSO is a simulation of social system where the “swarm” is a collection or population of moving individuals that tend to cluster together, while each individual seems to be moving in a random direction. It uses a number of particles that constitute a swarm moving around in the search space looking for the best solution [24, 25].

The idea of PSO is that the system is initialized with a population of random solutions called particles. Each particle which represents a candidate solution for a multidimensional optimization problem is also assigned with a randomized velocity and then the particles are then “flown” through the problem space. Each particle adjusts its “flying” according to its own flying experience as well as the flying experiences of other particles. Each particle keeps track of its coordinates in the problem space which are associated with the best solution or fitness it has achieved so far, called P_{best} . The PSO also tracks the best solution obtained so far by any particle in the swarm, called g_{best} [24, 25]. The PSO concept is, at each time step, changing the velocity of each particle toward its P_{best} and the g_{best} position. Every particle tries to modify its current position and velocity according to the distance between its current position and P_{best} , and the distance between its current position and g_{best} .

Clerc [26] proposed a constriction factor approach that modifies the particles position and velocity, given by Equations (2.20) and (2.21), respectively.

$$p_{i,g}^{(k+1)} = p_{i,g}^{(k)} + v_{i,g}^{(k+1)} \quad (2.20)$$

$$v_{i,g}^{(k+1)} = K * [v_{i,g}^{(k)} + c_1 * r_1 * (P_{best_i} - p_{i,g}^{(k)}) + c_2 * r_2 * (g_{best} - p_{i,g}^{(k)})] \quad (2.21)$$

Where, $p_{i,g}^{(k+1)}$ and $v_{i,g}^{(k+1)}$ represent the position and velocity of particle i at dimension g at the iteration $k + 1$, respectively. r_1 and r_2 are random numbers between 0 and 1. c_1 and c_2 are cognitive and social acceleration constants, respectively. K is call the constriction factor and given by $K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}$, with $\varphi = c_1 + c_2, \varphi > 4$. The velocity is confined with the range of $[-V_{max}, V_{max}]$. With $i = 1, 2, \dots, n; g = 1, 2, \dots, d$, where n is the number of particles in the swarm and d is the dimension of the optimization problem.

Chapter Three: Experimental Method

Chapter Three will discuss the experimental method. Including the data acquisition system designed and developed, the image processing algorithm, and the system identification experiment. A ten-layer thin wall was deposited while filming the melt pool. The melt pool size was then measured to determine the relationship between melt pool size and laser power. The chapter will also cover the system identification process and controller design.

3.1 Experimental Data Collection

Section 3.1 will cover the data acquisition system hardware and software.

3.1.1 Hardware

The block diagram of the data acquisition system developed for system identification is shown in Figure 3.1. The system consists of a National Instruments™ cRIO-9057 controller (Fig. 3.2) with an analog and digital I/O module, a YLR fiber laser with wavelength of 1070 nm, and the print chassis of the OPTOMECH LENS® 500 system (Fig. 3.3). The print chassis includes a Siemens 828D controller which handles the G-code. The image acquisition system uses a Basler ace, acA720-290gm monochrome camera with a resolution of 720 x 540 pixels, capturing at a frame rate of 300 frames per second. The captured images are post processed utilizing the National Instruments™ LabVIEW software.

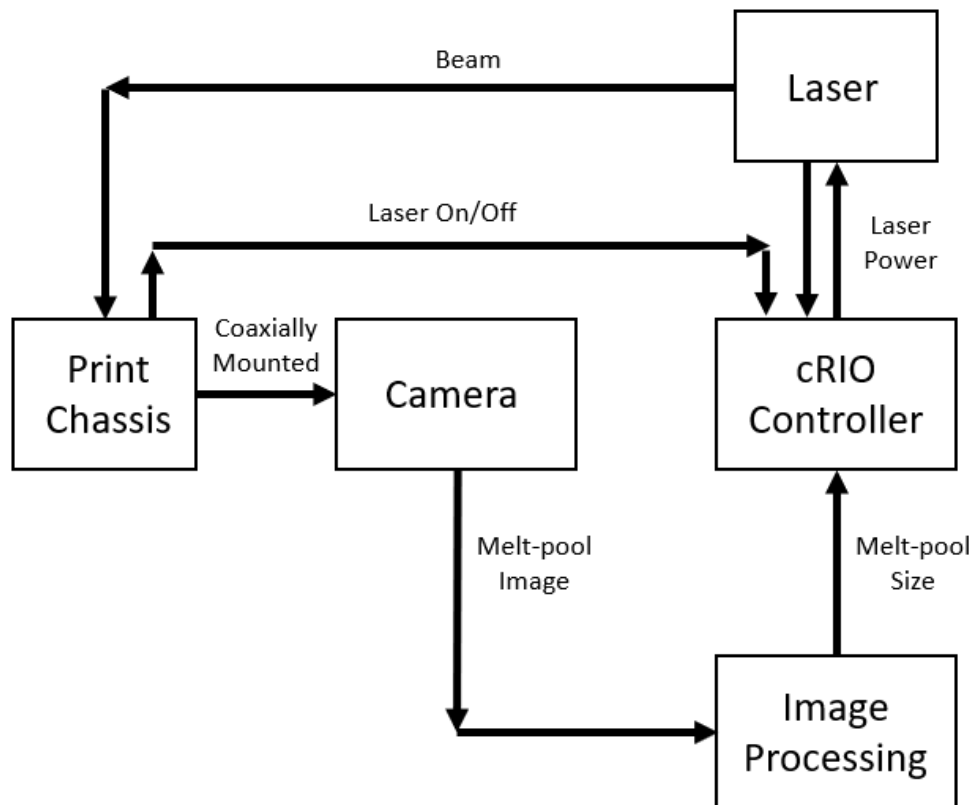


Figure 3.1 Schematic Overview of Data Acquisition System

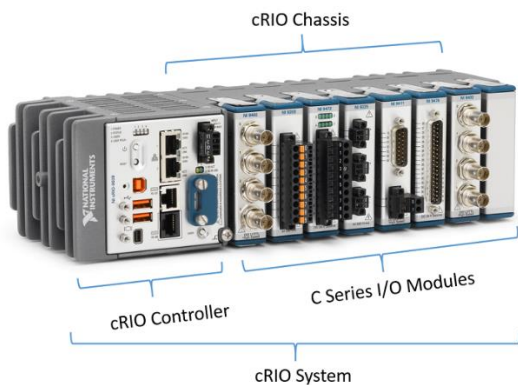


Figure 3.2 NI cRIO [22]



Figure 3.3 OPTOMECH LENS® 500 [23]

The image acquisition system is turned on by a digital signal sent from the Siemens controller to the cRIO controller. This signal is then passed on to the laser and turns the beam on. The system is turned off in a similar manner. The output laser power is recorded and timestamped by the cRIO with one reading corresponding to every image captured.

3.1.2 Software

All software was developed using National Instruments™ LabVIEW. This section will briefly explain the layout of the LabVIEW VI developed. There are two VI's that execute simultaneously, the image acquisition is run on the host PC and the laser power acquisition and digital control signals are handled on the cRIO. Figure (3.4) shows the LabVIEW front panel of the image and data acquisition VI run on the host PC.

The first task executed on the host PC is the generation of the input waveform from data in a .csv file. Then the camera is opened and configured for continuous acquisition. A buffer for 2000 images is built using the host PC's DRAM. This is to ensure that every image is captured at a rate of 308 fps. The data and image acquisition are started utilizing a digital signal from the Siemens controller on the LENS® when the laser emission is enabled. This signal is captured by the cRIO and relayed to the host PC. The laser on signal is then sent by the cRIO approximately 200 ms after, ensuring that no data is missed. The analog laser power return signal from the laser to the Siemens is captured by the cRIO. Due to a delay in communication between the cRIO and the host PC there is approximately a 200 ms slew between the images captured and the corresponding laser power. This can be corrected by shifting the recorded data, correlating when the laser was turned on to when the melt pool is first detected.

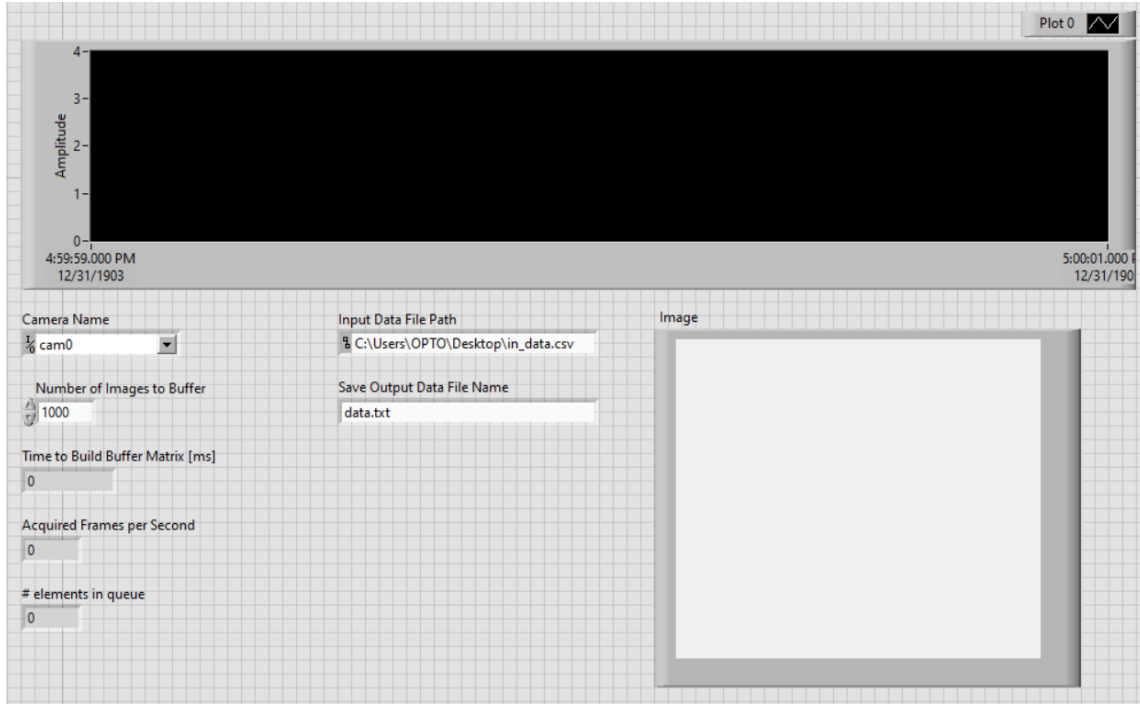


Figure 3.4 LabVIEW Front Panel

When the input waveform is complete the laser off signal is sent by the cRIO to the laser. This also stops the image and data acquisition. All digital signals are then reset to their default values by the cRIO. Last, the images saved in the data buffer are then written to file as .tif files. Complete documentation of developed LabVIEW DAQ software is available in Appendix C.

3.1.3 Image Processing

The image processing algorithm was developed using the Vision Assistant in the LabVIEW Vision Development Module and is shown in Figure 3.2. The original 8-bit gray level (0-255 intensity) image is captured by the camera (Fig. 3.2(a)). The image is then converted to a binary image (0-1) by applying a threshold value in (Fig. 3.2(b)). The threshold value is set to 50 in the experiments. The ideal threshold value was determined by depositing several tracks with different

widths. Then, the measurement of track widths was taken and compared with the corresponding images to get the appropriate threshold value. More information on the threshold value experiment can be found in Appendix B. The image is then masked or cropped to eliminate clutter (Fig. 3.2(c)). Due to the mounting of the camera the melt pool is always centered in the frame.

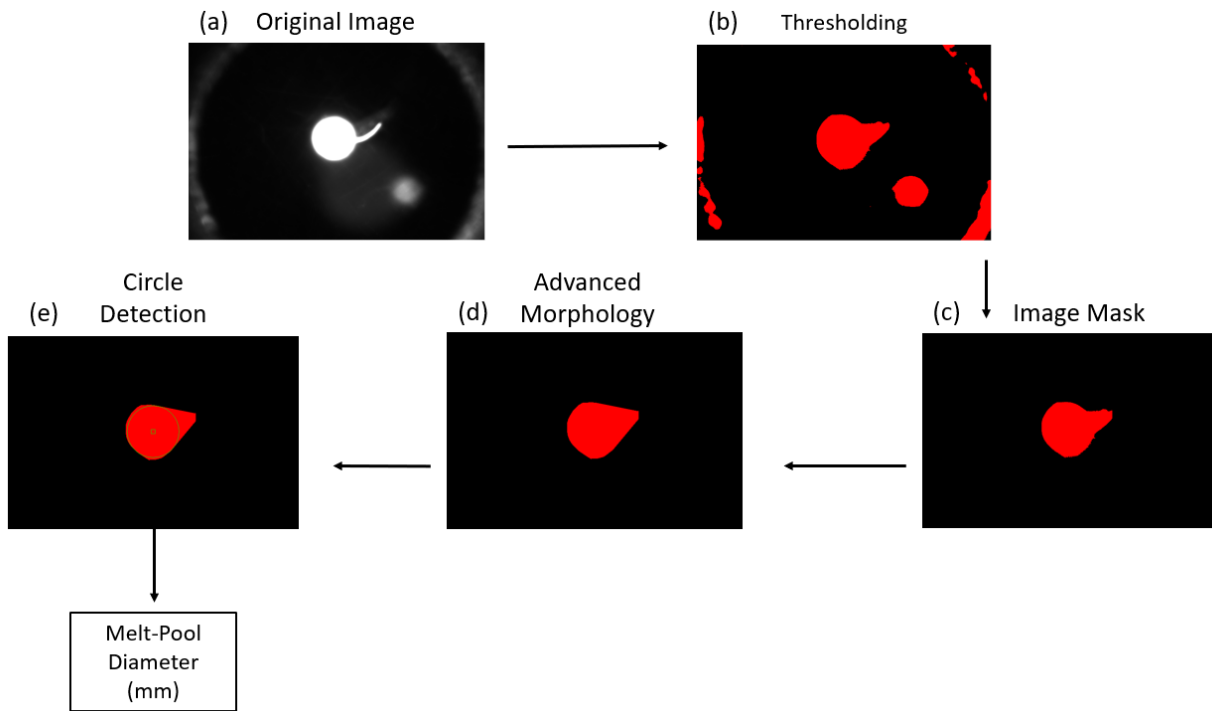


Figure 3.5 Image Processing Algorithm

An advanced morphology (convex hull) is then applied which eliminates or reduces the detection of multiple circles from (Fig. 3.2(d)). Ultimately, to measure the width of the melt pool, all circles contained inside the outline were detected (Fig. 3.2(e)). The largest circle was recognized as the melt pool. The width was obtained by measuring the number of pixels that made up the radius of the circle. During the camera calibration it was determined that 92 pixels is equivalent to 1 mm in length. See Appendix A for information on camera calibration. Using Equation 3.1 the diameter of the melt pool is calculated in millimeters.

$$\frac{pixels}{92} * 2 = diameter (mm) \quad (3.1)$$

3.2 System Identification Experiment

Section 3.2 will discuss the system identification (SID) experiment conducted. The raw data is also presented.

3.2.1 Experiment Parameters

The experiment consists of a pseudo-random binary input (laser power) with values of 300 and 400 W at a frequency of 4 Hz. The depositions consist of ten 25.4 mm (1 in) long layer at a travel speed of $4.23 \frac{mm}{sec}$ (10 ipm) for 6.5 seconds. The hopper that controls powder feed-rate is set to 5 rpm corresponding to a mass flow rate of 8.67 g/min. The print chamber environment is argon at a slightly positive pressure (≈ 10 mBar) to prevent air ingress through the chamber opening seals. The oxygen content is typically < 0.1 ppm. The chamber environment also supplies a gas pressurizing system that supplies argon at ≈ 25 psi to the powder feeders and print head center purge nozzle. The powder is spherical, gas atomized Type 316L stainless steel with a particle size of 45 – 150 μm , deposited onto 6.4 mm ($\frac{1}{4}$ in) 316L plates as substrates.

3.2.2 Experimental Data

Figures (3.6), (3.7), and (3.8) show the collected experimental data. For each layer approximately 1800 data points were captured. The laser power return signal is a 0-4 VDC analog signal with 0.97 VDC corresponding to a laser power of 300 W and 1.41 VDC corresponding to 400 W. The average width of the melt pool for the first layer as a function of laser power can be found in Table (B.1) in Appendix B.

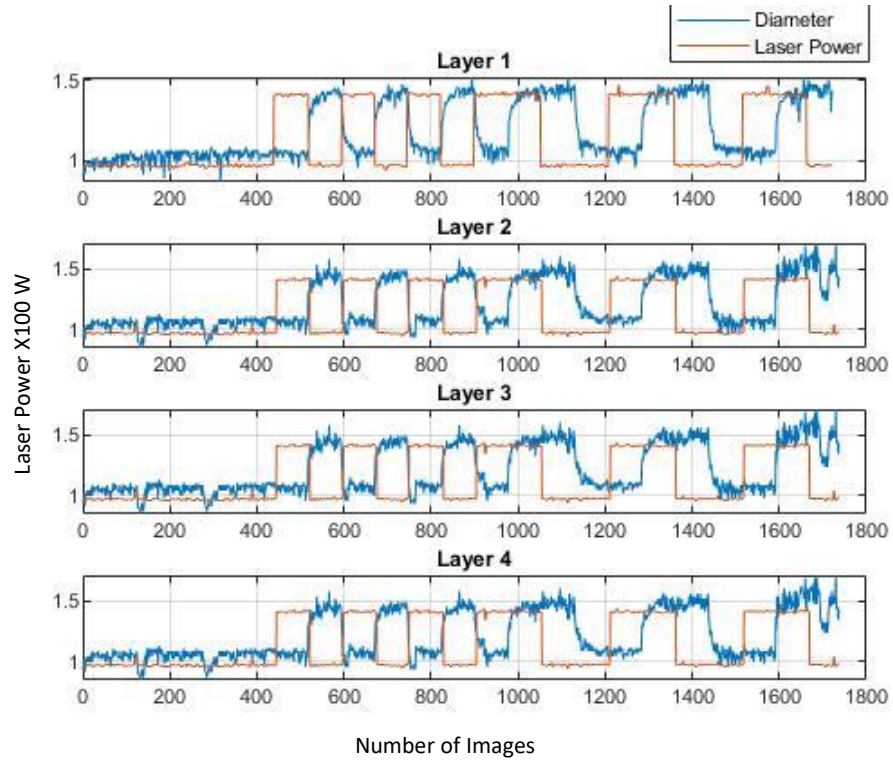


Figure 3.6 Layer 1 – 4

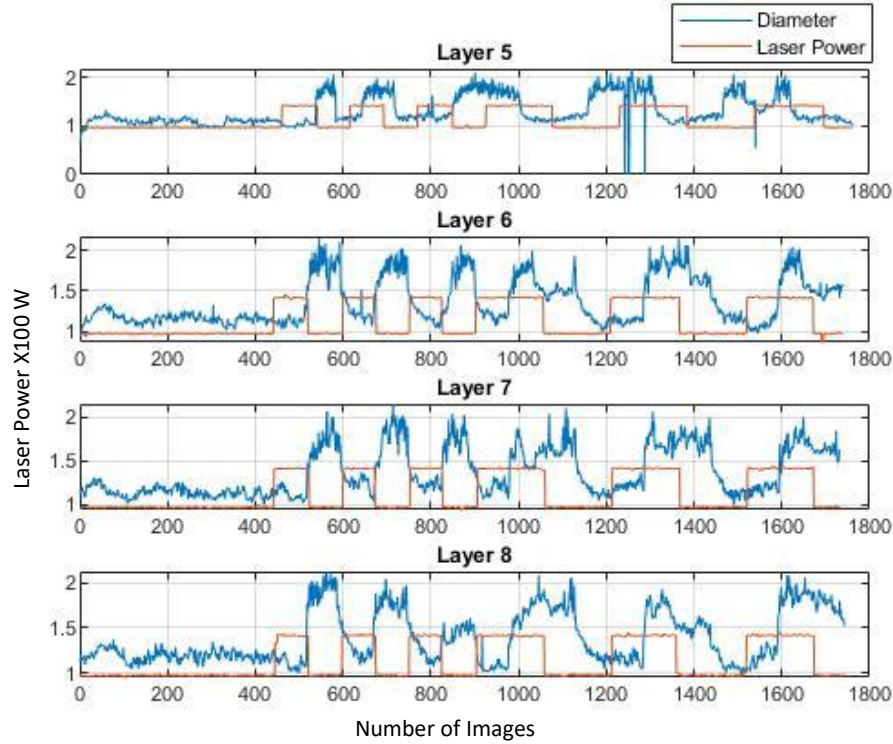


Figure 3.7 Layer 5 - 8

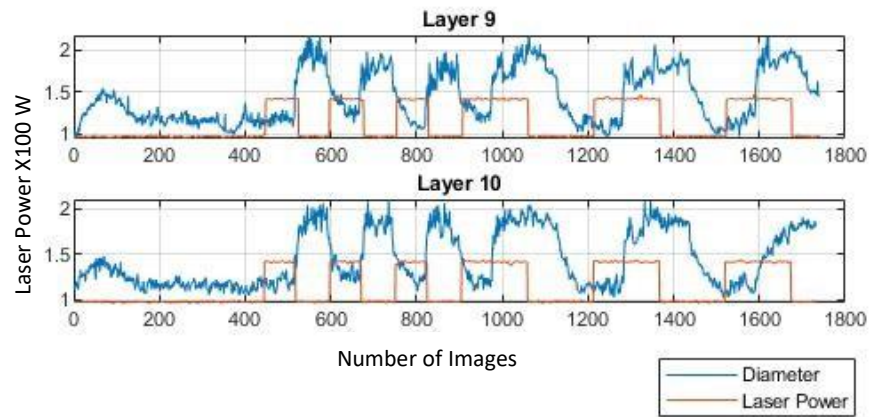


Figure 3.8 Layer 9 – 10

From the data the time delay in the system is clear and consistent in each layer. It is also evident that the melt pool diameter increases after layer 4. Also, as can be seen in all layers there is no overshoot in the responses. Looking at layer 5 in Figure (3.7) the data is inconsistent with the other 9 layers. During the deposition of layer 5 a powder nozzle was obstructed by melted powder throw off from the previous layer. The experiment was halted, and the obstruction was cleared. It was not considered necessary to restart the experiment so long as no other layers were deposited with obstructions. While a model of layer 5 was extracted the data will not be used in the design of the controller.

3.3 System Identification

Section 3.3 covers the process of system identification.

3.3.1 Dynamic Model

The MATLAB® system identification toolbox was utilized to build first order transfer functions with time delay of each layer. The GUI for the system identification toolbox is show below in Figure (3.9). The structure of each transfer function is shown in Equation (2.2). The transfer

function parameters for each layer are shown in Table (3.1). The static gain generally decreased with each layer, while the time constant decreased with each layer as can be seen in Figure (3.10). The time delay remained constant for every layer. Again, note that the data for layer 5 was neglected. In its place an average of layer 4 and layer 6 was taken.

Table 3.1 Transfer Function Parameters for different Layers

| Layer | Kp | Tp | Td |
|-------|--------|---------|--------|
| 1 | 1.0554 | 0.25385 | 0.099 |
| 2 | 1.0772 | 0.21084 | 0.099 |
| 3 | 1.1064 | 0.22969 | 0.099 |
| 4 | 1.1337 | 0.2095 | 0.099 |
| 5 | 1.2076 | 1.3051 | 0.0122 |
| 6 | 1.2107 | 0.1739 | 0.099 |
| 7 | 1.2166 | 0.19029 | 0.099 |
| 8 | 1.2272 | 0.16571 | 0.099 |
| 9 | 1.2761 | 0.15849 | 0.099 |
| 10 | 1.2869 | 0.19788 | 0.099 |

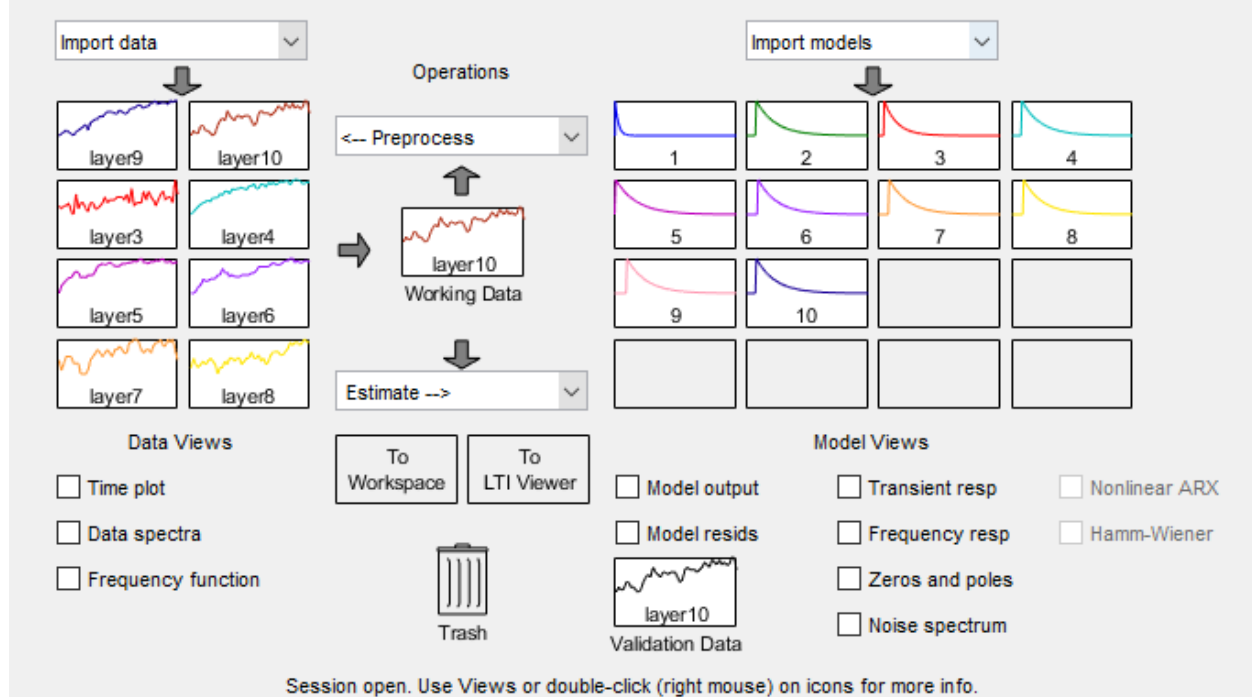


Figure 3.9 System Identification GUI

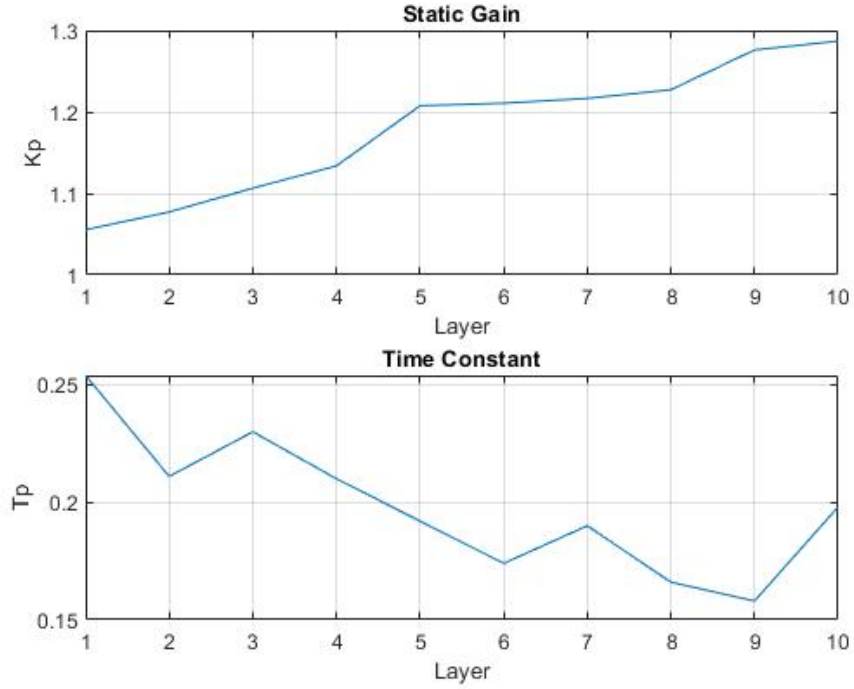


Figure 3.10 Static Gain and Time Constant by Layer

After the inclusion of the Padè approximation each transfer function is represented by Equation (3.2). The individual layer parameters are shown in Table (3.2). Again, note that the parameter values of layer 5 diverge from the trend.

$$G(s) = \frac{B(s)}{A(s)} = \frac{-b_3s^3 + b_2s^2 - b_1s + b_0}{a_4s^4 + a_3s^3 + a_2s^2 + a_1s + a_0} \quad (3.2)$$

Table 3.2 Transfer Function Parameters for different Layers with Padè Approximation

| Layer | b_0 | b_1 | b_2 | b_3 | a_0 | a_1 | a_2 | a_3 | a_4 |
|-------|----------|----------|-------|-------|----------|----------|---------|-------|--------|
| 1 | 13.05e05 | 6459 | 127.9 | 1.055 | 1.237e05 | 3.752e04 | 1675 | 31.77 | 0.2539 |
| 2 | 1.333e05 | 6594 | 130.6 | 1.077 | 1.237e05 | 3.22e04 | 1412 | 26.56 | 0.2108 |
| 3 | 1.368e05 | 6773 | 1.341 | 1.106 | 1.237e05 | 3.453e04 | 1527 | 28.84 | 0.2297 |
| 4 | 1.402e05 | 6940 | 137.4 | 1.134 | 1.237e05 | 3.203e04 | 1404 | 26.39 | 0.2095 |
| 5 | 7.98e07 | 4.868e05 | 1188 | 1.208 | 6.608e07 | 8.665e07 | 5.27e05 | 1285 | 1.305 |

| | | | | | | | | | |
|-----------|----------|------|-------|-------|----------|----------|------|-------|--------|
| 6 | 1.497e05 | 7412 | 146.8 | 1.211 | 1.237e05 | 2.763e04 | 1186 | 22.08 | 0.1739 |
| 7 | 1.505e05 | 7448 | 147.5 | 1.217 | 1.237e05 | 2.966e04 | 1286 | 24.07 | 0.1903 |
| 8 | 1.518e05 | 7513 | 148.8 | 1.227 | 1.237e05 | 2.662e04 | 1136 | 21.09 | 0.1657 |
| 9 | 1.578e05 | 7812 | 154.7 | 1.276 | 1.237e05 | 2.572e04 | 1091 | 20.21 | 0.1585 |
| 10 | 1.592e05 | 7878 | 156 | 1.287 | 1.237e05 | 3.059e04 | 1333 | 24.99 | 0.1979 |

3.3.2 Stability

Control systems should be stable to avoid loss of control and damage to equipment. Stability can be assessed by looking at the poles of the closed-loop transfer function. All layers were determined to be stable, only the stability of the first layer will be discussed.

The poles of the first layer transfer function were found to be $(-49.8750 + 47.8186i)$, $(-49.8750 - 47.8186i)$, $(-10.6227 + 9.8421i)$, and $(-10.6227 - 9.8421i)$ plotted in Figure (3.11). The feedback loop is stable since all poles have negative real parts. The sensitivity of the stability was determined from the root locus plot shown in Figure (3.12). The root locus shows that the feedback loop is stable for a gain of $(k < 4.4)$. The closed-loop step response with $k = 4.4$ and $k = 4.5$ confirms this in Figure (3.13).

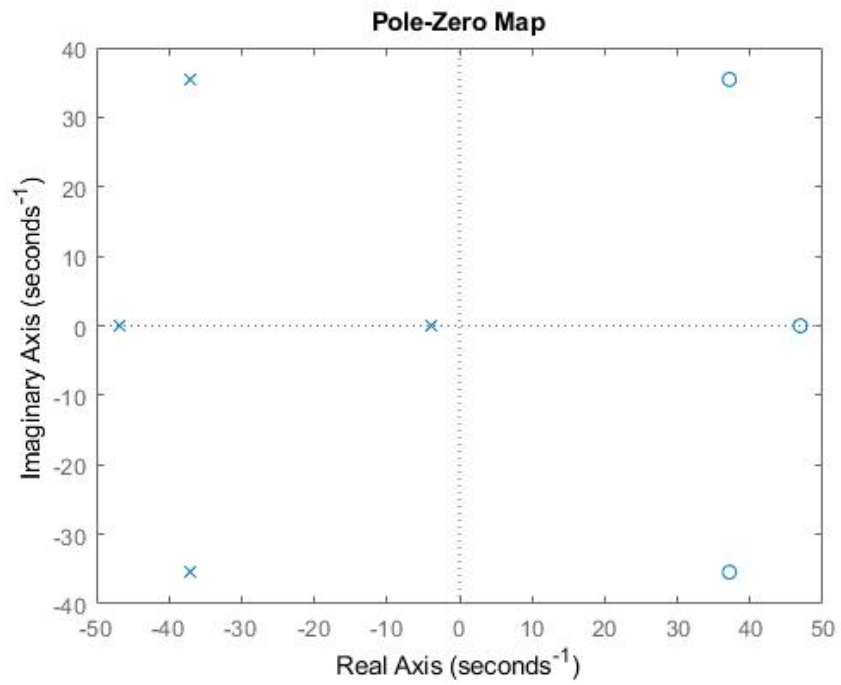


Figure 3.11 Pole-Zero Map (Layer 1)

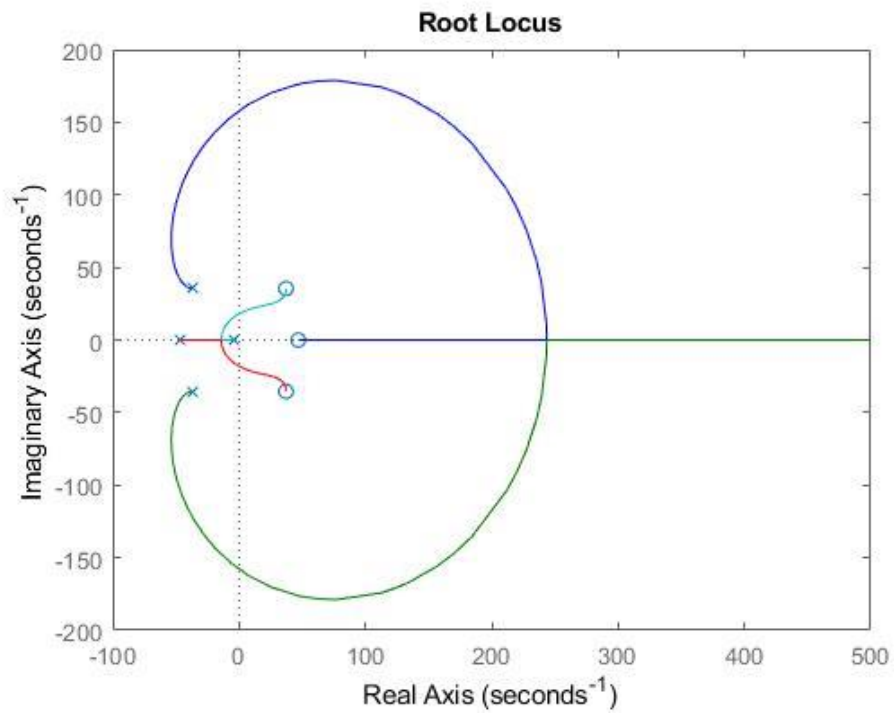


Figure 3.12 Root Locus (Layer 1)

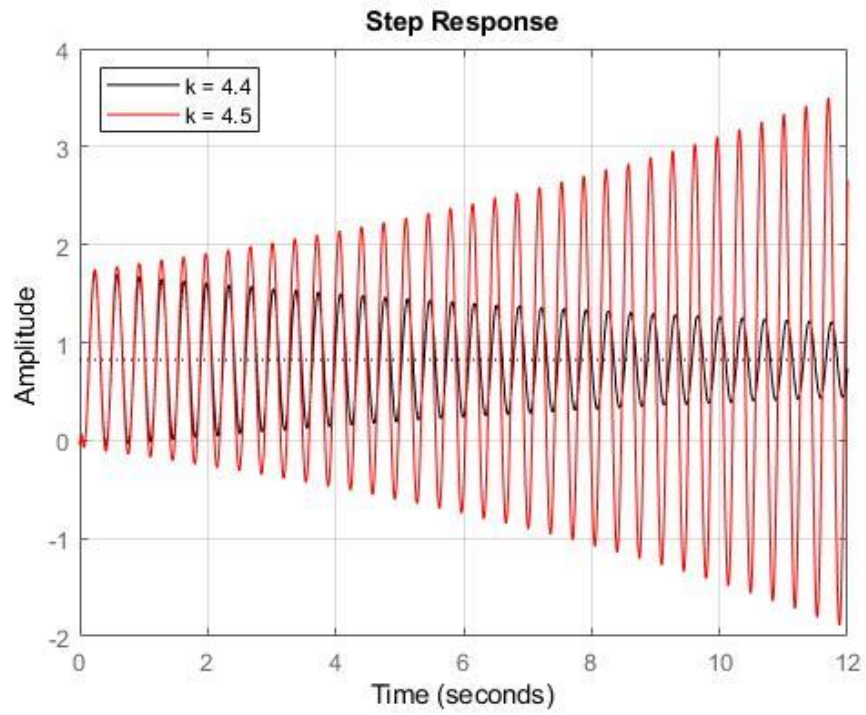


Figure 3.13 Step Response showing Sensitivity to Gain

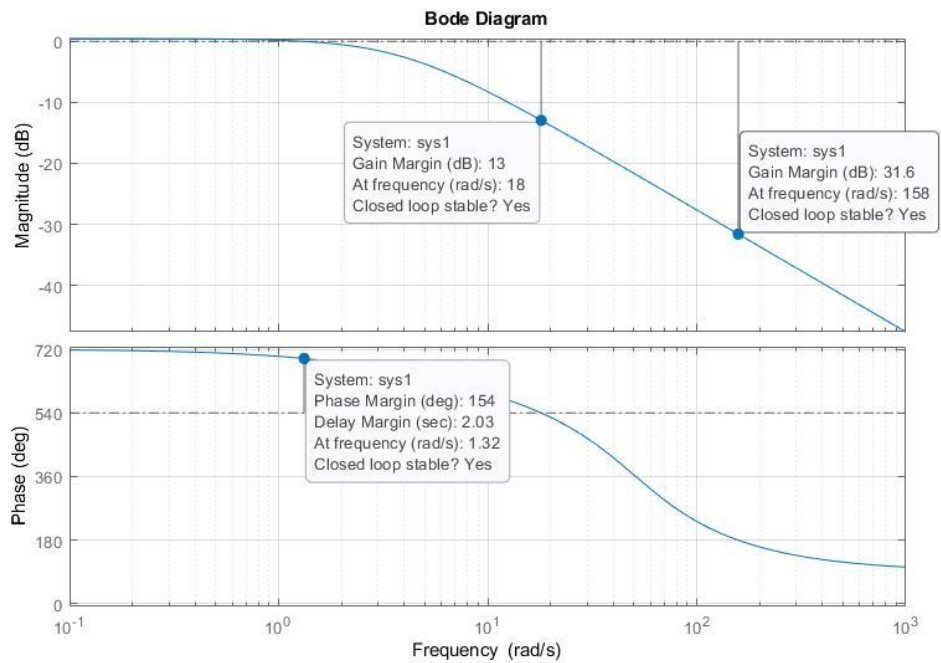


Figure 3.14 Bode Diagram (Layer 1)

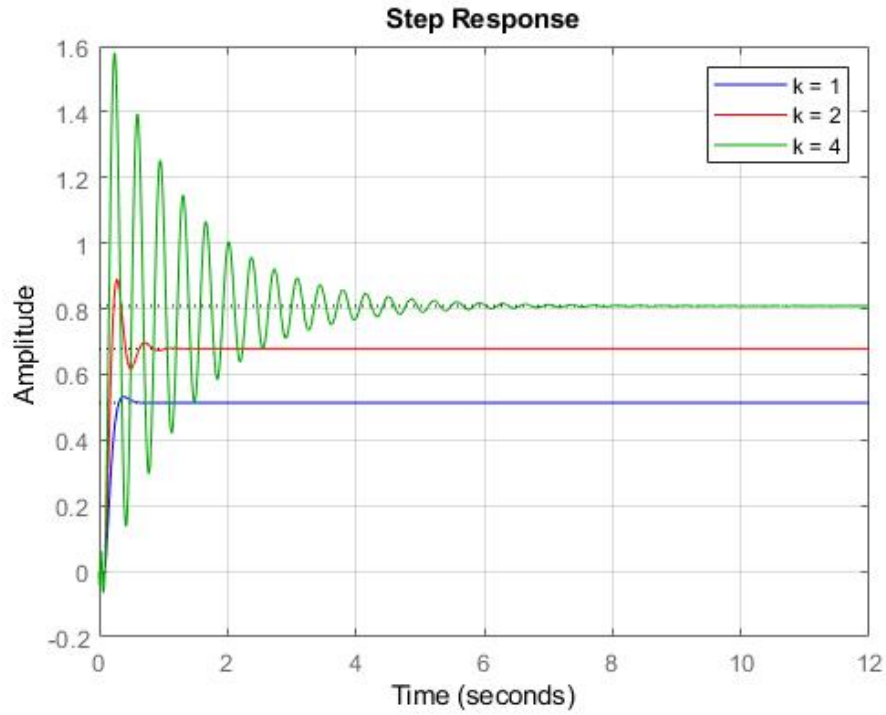


Figure 3.15 Step Response with Increasing Gain Values

The bode diagram showing the gain and phase margin of the frequency response of the system is shown in Figure (3.14). These margins indicate the relative stability of the closed-loop system formed by applying unit negative feedback to the system. There are two gain margins at 13 dB and 31.6 dB. Both margins are positive indicating that stability is lost by increasing the gain. This is confirmed in Figure (3.15) by plotting the closed-loop step response where the gain value is increased.

3.4 Controller Design

Section 3.4 discussed the design of the FST-PI controller.

3.4.1 Fuzzy Inference Engine

The block diagram of the closed loop control system is shown in Figure (2.3). The system error and the rate of change of the error are sent to the fuzzy inference engine and new PI parameters are calculated. The fuzzy inference engine was generated from data using the MATLAB® Optimization toolbox. A Mamdani type engine was used with the And Method “min”, Or Method “max”, Implication Method “min”, Aggregation Method “max”, and the Defuzzification Method “centroid”. The rule base was learned using particle swarm optimization from experimental data. Eight rules were learned with each input having three triangular membership functions and each output having nine triangular membership functions. To reduce the number of rules, three membership functions were used for each input variable corresponding to negative error, zero error, and positive error {N, Z, P}. This results in $3^2 = 9$ input membership function combinations. Therefore, the fuzzy inference engine uses a maximum of 9 rules corresponding to the input membership function combinations. To improve data generalization, 9 membership functions were used for the output variables corresponding to negative big, negative medium, negative small, negative zero, zero, positive zero, positive small, positive medium, and positive big {NB, NM, NS, NZ, Z, PZ, PS, PM, PB}. Doing so allows the inference engine to use a different output membership function for each rule. The rule base is shown in Table (3.3). The fuzzy inference system (FIS) was validated using a validation data set. The initial pre-tuned and tuned FIS are shown in Figure (3.20) and Figure (3.21), respectively. The root mean square error (rmse) was also calculated with values of 0.35 for untuned and 0.18 for the tuned FIS because the values are relatively similar, the parameter values are not over tuned.

Table 3.3 Fuzzy Rule Base

| | |
|---|--|
| 1 | <i>If (e is Z) then (K_p is NS) and (K_i is NS)</i> |
| 2 | <i>If (e is N) then (K_p is PM) and (K_i is PS)</i> |
| 3 | <i>If (e is Z) then (K_p is PB) and (K_i is PB)</i> |
| 4 | <i>If (e is P) and (e is Z) then (K_p is PS) and (K_i is NZ)</i> |

| | |
|---|--|
| 5 | <i>If (\dot{e} is P) then (K_p is Z)</i> |
| 6 | <i>If (e is N) and (\dot{e} is N) then (K_p is PZ) and (K_i is NZ)</i> |
| 7 | <i>If (e is P) and (\dot{e} is N) then (K_p is NM) and (K_i is PZ)</i> |
| 8 | <i>If (e is P) and (\dot{e} is P) then (K_p is Z) and (K_i is Z)</i> |

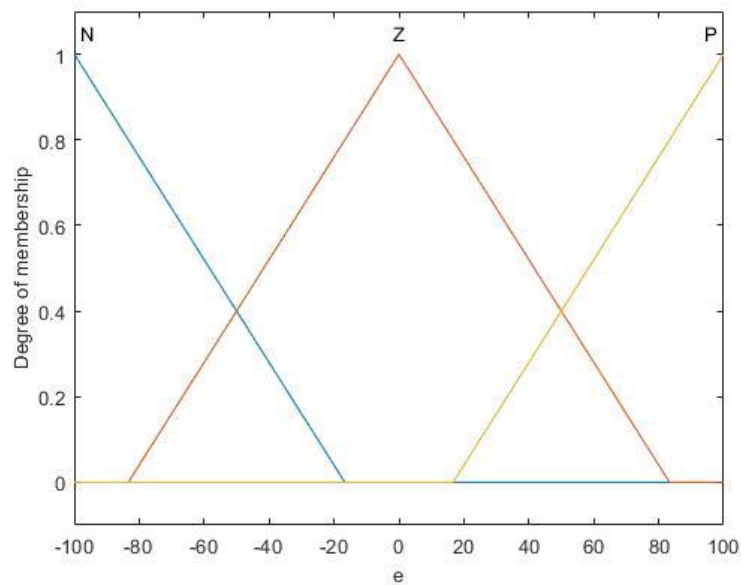


Figure 3.16 e Membership Functions

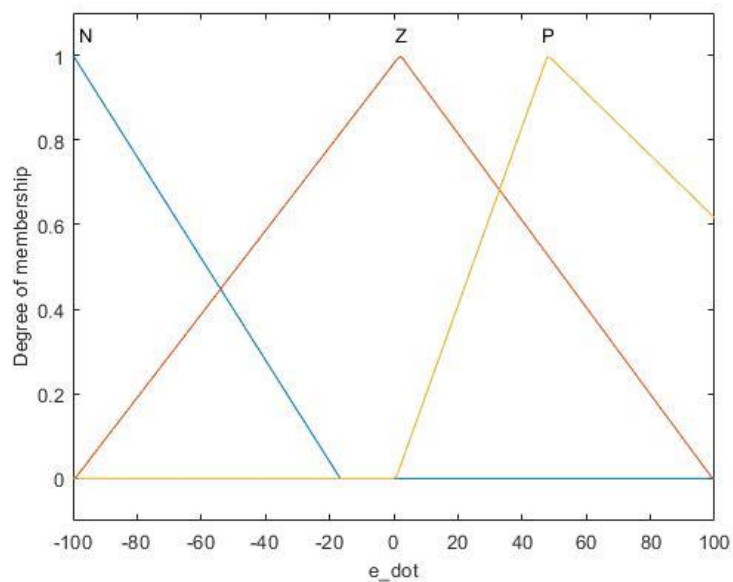


Figure 3.17 \dot{e} Membership Functions

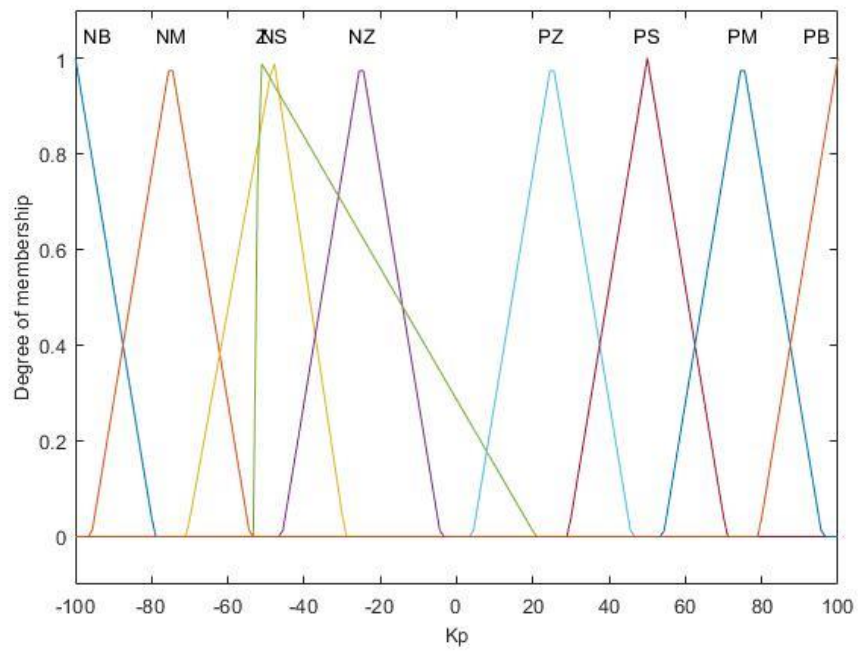


Figure 3.18 K_p Membership Functions

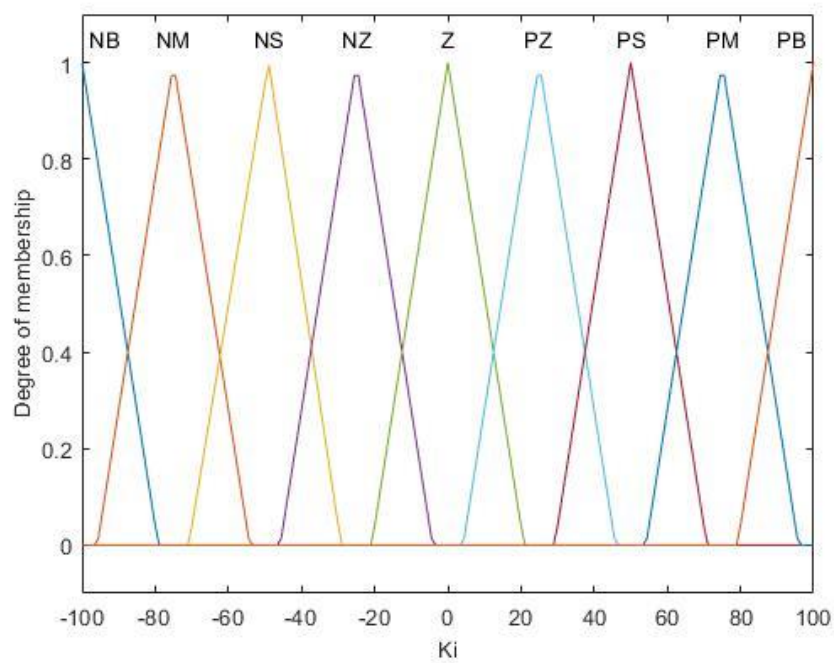


Figure 3.19 K_i Membership Functions

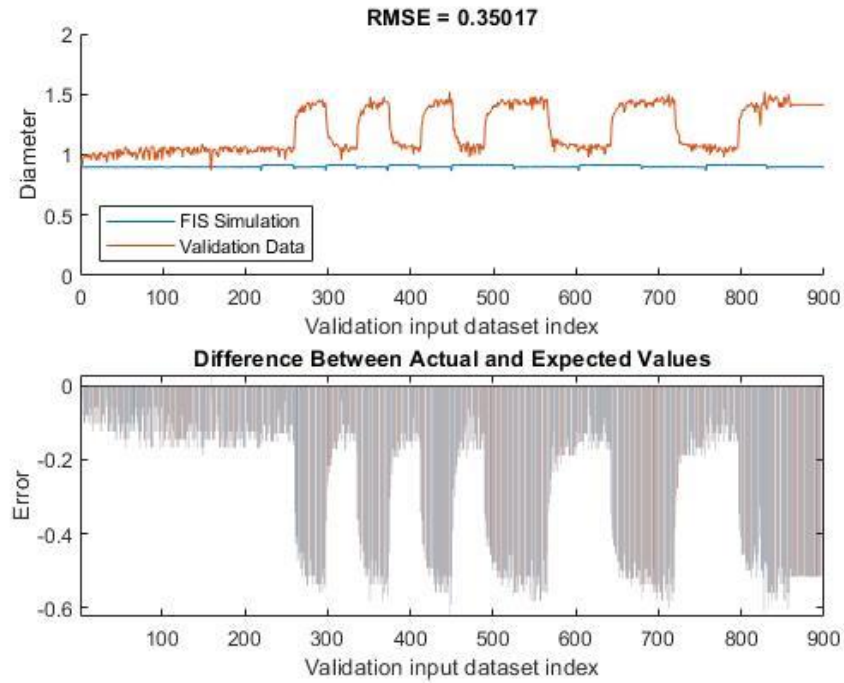


Figure 3.20 Initial FIS Validation

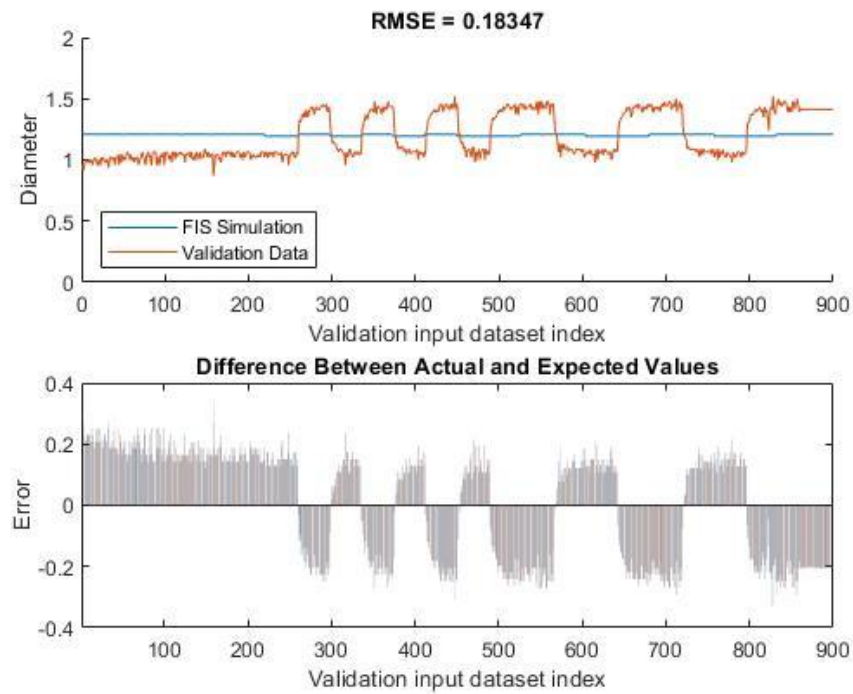


Figure 3.21 Tuned FIS Validation

3.4.2 PI Parameters

The initial PID parameters were tuned manually by looking at the step response of the closed loop system of layer one, shown in Figure (3.22). While, the derivative term could lead to faster response time, it caused oscillation and eventual instability in the system. Therefore, a PI algorithm was chosen for control of this process. The PI parameters are adjusted in real time using Equations (2.17) and (2.18), respectively. The initial parameters are set to $K_p = 1.23$, and $K_i = 6.19$. Additional, manual tuning of the initial parameters for the FST-PI controller resulted in initial values of $K_p = 1$, and $K_i = 2.75$. The PI controller used in the simulation maintained the initial parameter values. The SIMULINK® block diagram of the control is shown in Figure (3.23).

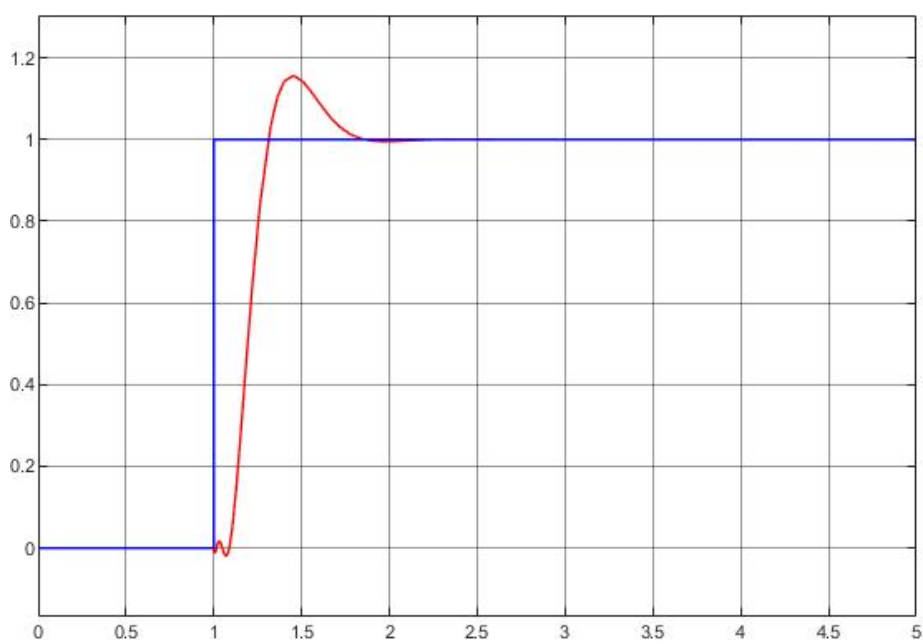


Figure 3.22 Step Response

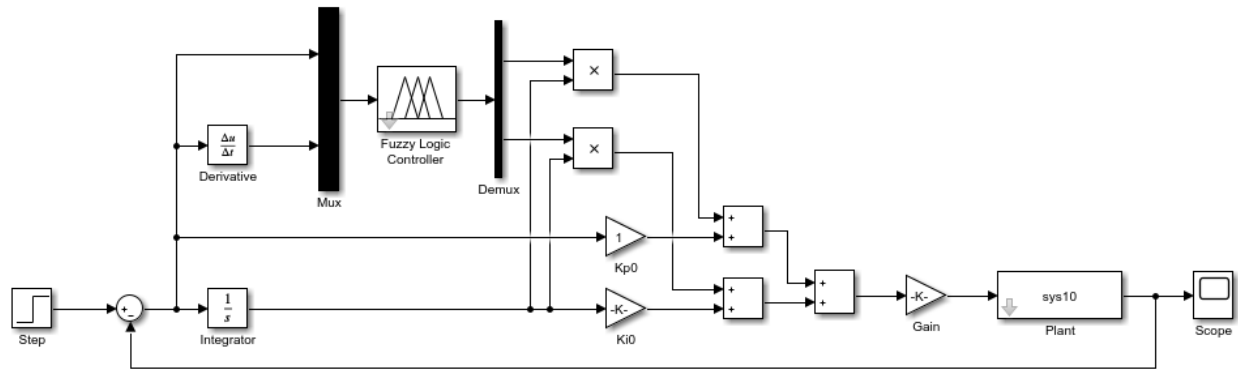


Figure 3.23 SIMULINK® Block Diagram of FST-PI Controller

Chapter Four: Simulation and Conclusion

Chapter four will discuss the simulation conducted and the results of the simulation. The results are discussed, and the conclusion presented.

4.1 Simulation

Section 4.1 will present the simulation conducted, the parameters used, and the results.

4.1.1 Simulation Setup

The step response of the FST-PI controller developed was compared to a simple PI controller.

The PI controller parameters were manually tuned and set to $K_p = 1.23$, and $K_i = 6.20$, respectively. The SIMULINK® block diagram used for the simulations is shown in Figure (4.1).

The model of each layer is stored in the MATLAB® workspace and called using the LTI block.

Note that the step response of layer five was not considered.

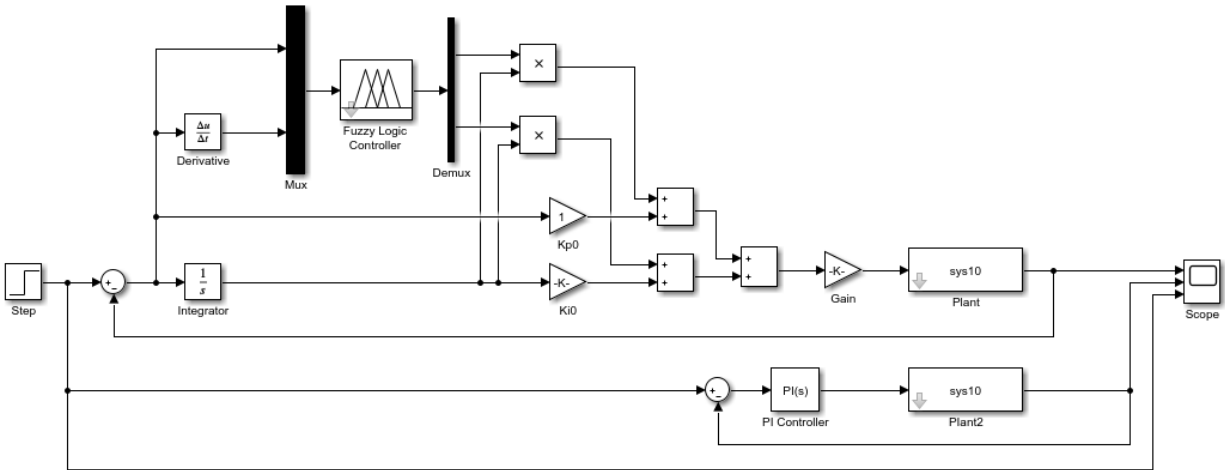


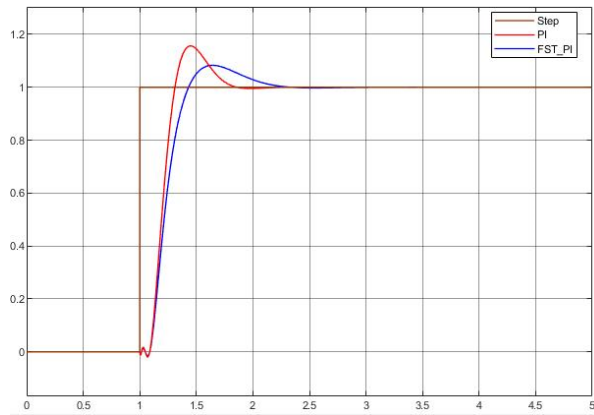
Figure 4.1 Simulation SIMULINK® Block Diagram

4.1.2 Results

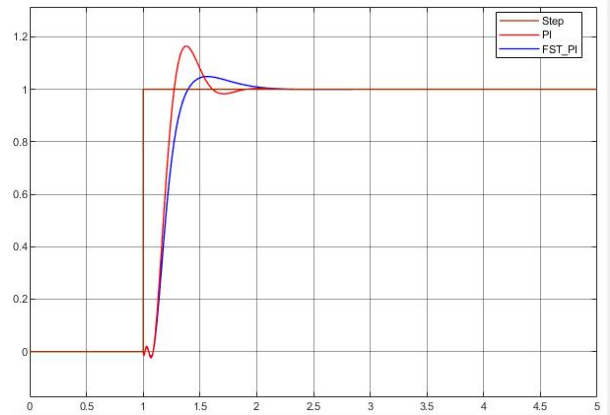
The rise time and percent overshoot of each layer is shown in Table (4.1). Figure (4.1) shows the step responses of each layer.

Table 4.1 Rise Time and % Overshoot

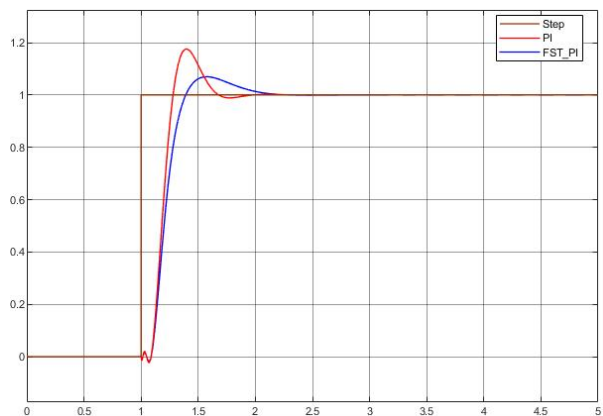
| Layer | FST-PI Controller | | PI Controller | |
|-------|-------------------|-------------|----------------|-------------|
| | Rise Time (ms) | % Overshoot | Rise Time (ms) | % Overshoot |
| 1 | 233.7 | 8.2 | 161.9 | 15.9 |
| 2 | 198.0 | 4.8 | 134.8 | 16.1 |
| 3 | 201.9 | 7.1 | 139.3 | 17.5 |
| 4 | 178.4 | 5.9 | 125.7 | 20.1 |
| 6 | 136.3 | 4.8 | 101.3 | 26.6 |
| 7 | 145.2 | 6.0 | 108.1 | 25.0 |
| 8 | 127.2 | 6.0 | 95.5 | 30.0 |
| 9 | 116.2 | 8.3 | 90.0 | 33.6 |
| 10 | 138.8 | 8.3 | 103.4 | 30.0 |



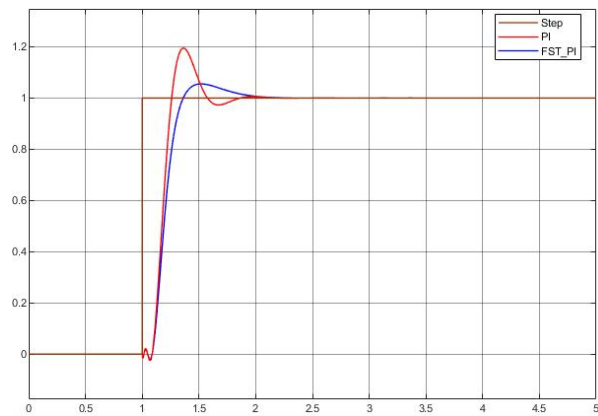
Layer 1



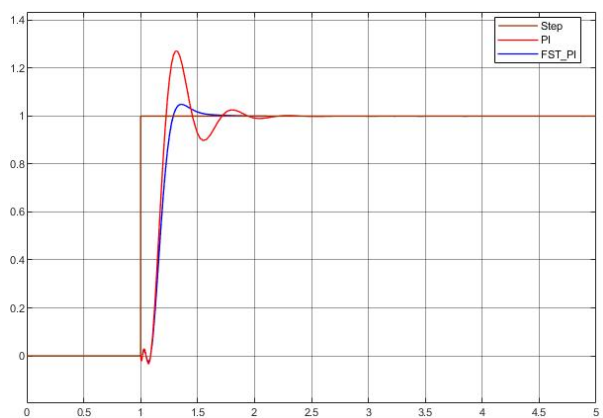
Layer 2



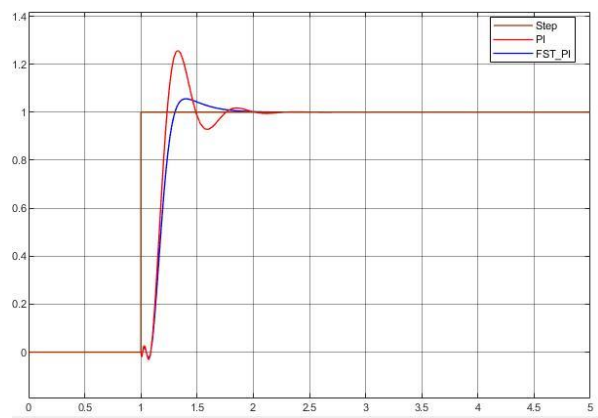
Layer 3



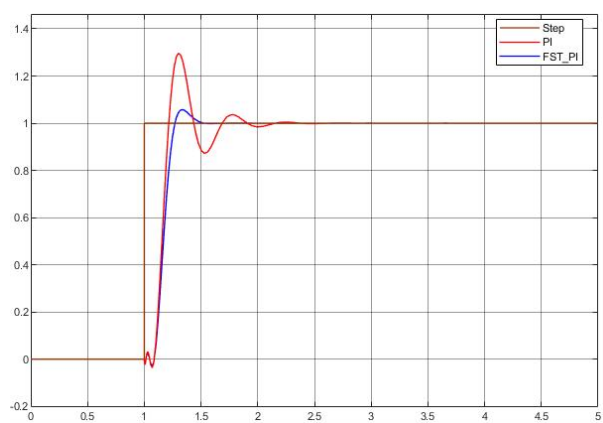
Layer 4



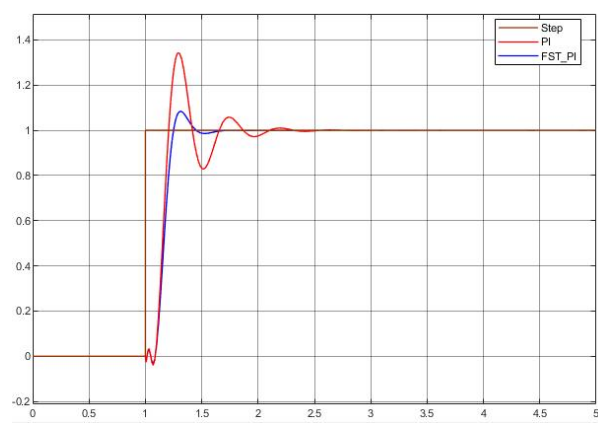
Layer 6



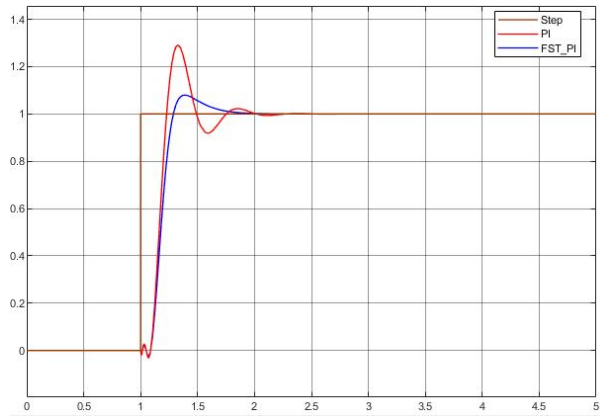
Layer 7



Layer 8



Layer 9



Layer 10

Figure 4.2 Layer Step Response

4.1.3 Discussion

The dynamic response of the system changes from layer to layer. This change is due to the change in thermal conductivity after each layer is deposited. The change of parameters for each layer was linear with the parameter values in the denominator increasing with each layer and the parameters values in the numerator decreasing with each layer, this can be seen in Table (3.2). The change in value between each layer was generally small until layer four, where the value change was more pronounced. After layer four the change in value between each layer was again generally small. This suggests that the greatest change in the heat conduction mode occurred between layer four and layer six. It can be suggested that at this point heat transfer mode changes from 3D to 2D conduction. Therefore, the thermal analysis of the change observed between these two layers will be considered in future studies. The geometry and volume of the component being printed may also affect this change. With smaller volumes cooling more rapidly than larger prints.

In Figure (4.1) the time delay of the system is evident in each step response. This time delay is approximately 100 ms, the small oscillation in the delay could be reduced by increasing the order of the Padè approximation. However, doing so did not increase the model accuracy or the overall controller performance. High-order Padè approximations produce transfer functions with clustered poles. These pole configurations can be sensitive to perturbation and it is recommended to have order $N < 10$ [29].

Looking at the step response of each layer it is clear that both controllers perform reasonably well in the first layer. Though the simple PI controller quickly begins to oscillate in layer two. As previously mentioned, the melt pool diameter increased significantly after the fourth layer. This is evident in the increased oscillation in the step response between layer four and layer six. Note that due to the poor quality of the data from layer five the step response was not considered. There is no significant change in the step response between layer six and layer 10. The oscillation in the simple PI controller increased with each layer. It is assumed that this oscillation would eventually reach instability after sufficient layers were deposited. Whether it is possible to deposit a sufficient number of layers is practical or simply theoretical cannot be determined. While the FST-PI controller performs consistently in each layer with the rise time and overshoot decreasing with each layer, shown in Table (4.1).

These results suggest that the FST-PI controller will result in a more consistent melt pool diameter during the build. From the experimental data it is shown that the melt pool diameter increases significantly after layer 4. This has also been observed in small diameter cylinders printed using this system. Where the diameter of the cylinder is smaller near the substrate and as the height of the cylinder increases, the diameter increases until an equilibrium point is reached.

The objective of this research was to develop a controller to maintain a constant melt pool diameter during the build process. It was the intent that a constant diameter will result in a more uniform track morphology and this will result in more uniform material properties in specimens printed using the laser powder directed energy deposition process. The simulation conducted here suggests that the FST-PI controller developed is capable of achieving this objective.

4.2 Conclusion

A process monitoring control system based on an image of the melt pool was developed and simulated using MATLAB® and SIMULINK®. This research showed that the laser powder-directed energy deposition process is a non-linear process. The dynamic model of the system changed during each layer of the deposition. This non-linearity is most likely due to the change in heat transfer mode during the process.

Due to this non-linearity a fixed parameter controller could not guarantee consistent performance. An FST-PI controller was developed and simulated that showed consistent performance in each layer. Based on the results of this simulation an FST-PI controller is being developed utilizing NI LabVIEW. The results of real-time measurement and microstructure analysis will be presented in a separate paper.

References

- [1] Akbari, Meysam, “Development of a Robotized Laser Directed Energy Deposition System and Process Challenges” (2019). *Mechanical Engineering Research Theses and Dissertations*. 21. https://scholar.smu.edu/engineering_mechanical_etds/21
- [2] Uriondo A, Esperon-Miguez M, Perinpanayagam S., “The Present and Future of Additive Manufacturing in the Aerospace Sector: a review of Important Aspects.”, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 229(11):2132-2147
- [3] Additive Manufacturing – General Principles – Terminology (2015) ISO/ASTM 52900. *International Organization of Standards*, Geneva.
- [4] Chang, K., “e-Design, Computer-Aided Engineering Design.”, Academic Press, 2015.
- [5] Avila, J.D., Bose, S., Bandyopadhyay, A., “Additive Manufacturing of Titanium and Titanium alloys for Biomedical Applications.” Woodhead Publishing Series in Biomedical, 2018.
- [6] Reichart, A., Dillion, R.P., Borgonia, J.P., Shapiro, A., McEnerney, B., Momose, T., Hosemann, P., “Development and Characterization of Ti-6Al-4V to 304L Stainless Steel Gradient Components Fabricated with Laser Deposition Additive Manufacturing.” *Materials and Design* 104 (2016) 404-413
- [7] Wilson, J.M., Piya, C., Shin, Y.C, Zhao, F. Ramani, K., “Remanufacturing of Turbine Blades by Laser Direct Deposition with its Energy and Environmental impact Analysis.” *Journal of Cleaner Products* 80:170-178
- [8] Akbari, M., Kovacevis, R., “Laser Directed Energy Deposition Process.” *Materials World Magazine*, 2019. <https://www.iom3.org/materials-world-magazine/feature/2019/sep/27/laser-directed-energy-deposition-process>
- [9] Akbari, M., Kovacevis, R., “Closed Loop Control of Melt Pool Width in Robotized Laser Powder-Directed Energy Deposition Process.” *The International Journal of Advanced Manufacturing Technology* (2019) 104:2887-2898
- [10] Tang, L., Ruan, J., Landers, R.G., Liou, F., “Variable Powder Flow Rate Control in Laser Metal Deposition Processes.” *Journal of Manufacturing Science and Engineering*, 2008, 130(4):041016
- [11] Bi, G., Gasser, A., Wissenbach, K., Drenker, A., Poprawe, R., “Characterization of the Process Control for the Directed Laser Metallic Powder Deposition.” *Surface and Coating Technology* 201(6):2676-2683

- [12] Mohammad Hossein Farshidianfar (2017), “Real-Time Closed-Loop Control of Microstructure and Geometry in Laser Materials Processing.” UWSpace.
<http://hdl.handle.net/10012/12735>
- [13] “Additive manufacturing” (2019, Feb. 27), “The Additive Manufacturing Industry Landscape 2019: 171 Companies Driving the Industry Forward [Updated]”, AMFG Blog.
<https://amfg.ai/2019/02/27/additive-manufacturing-industry-landscape-2019/?cn-reloaded=1>
- [14] “The structure” (2020), “What is a Process Model?”, MATLAB® Documentation.
<https://www.mathworks.com/help/ident/ug/what-is-a-process-model.html>
- [15] Dorf, R.C., Bishop, R.H., “Modern Control Systems, 13th Edition.” Pearson Education Inc, 2017
- [16] Zadeh, L.A., “Fuzzy Sets”, Information and Control. 8 (1965) 338-353
- [17] Candel, A., Langholz, G., “Fuzzy Control Systems.” CRC Press, 1994.
- [18] Silva, J.F., Pinto, S.F., “Power Electronics Handbook. 4th Edition.” Academic Press, 2001
- [19] Ross, T., “Fuzzy Logic with Engineering Applications. 3rd Edition.” Wiley, 2010.
- [20] Junwen Deng, Liu Yang, Xinrong Cheng, Wu Liu. “Self-tuning PID-type Fuzzy Adaptive Control for CRAC in Datacenters.” 7th International Conference on Computer and Computing Technologies in Agriculture (CCTA), Sep 2013, Beijing, China. pp.215-225, ff10.1007/978-3-642-54344-9_27ff. fffhal01220917f
- [21] Live, H.J., Wang, H.B., Zhu, W.M., Shen, Z.H., Chen, J.Y., “Simulation Research of Fuzzy Auto-Tuning PID Controller Based on MATLAB.” 2017 International Conference on Computer Technology, Electronics and Communications (ICCTEC)
- [22] Setting Up Your CompactRIO System (2018). <http://www.ni.com/tutorial/54281/en/>
- [23] LENS® 500 Machine Tool System. <https://optomec.com/3d-printed-metals/lens-printers/lens-500-machine-tool-systems/>
- [24] Kennedy, J., Eberhart, R., “Particle Swarm Optimization”, *Proceedings of ICNN’95 – International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi:10.1109/ICNN. 1995.488968.
- [25] Eberhart, R., Shi, Y., “Comparison Between Genetic Algorithms and Particle Swarm Optimization” *Proceedings of IEEE International Conference on Evolutionary Programming*, Anchorage, AK, 1998, pp. 611-616.
- [26] Eberhart, R., Shi, Y., “Particle Swarm Optimization: Development, Applications and Resources” *Proceedings of Congress on Evolutionary Computation*, vol. 1, 2001, pp. 81-86.
- [27] Kennedy, J., Eberhart, R., “A New Optimizer Using Particle Swarm Theory” *Proceedings of 6th International Symposium on Micromachine and Human Science*, Nagoya, Japan, 1995, pp. 39-43.

[28] Clerc, M., “The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization” *Proceedings of the 1999 congress on Evolutionary Computation*, Vol. 3, 1999, pp. 1951-1957.

[29] “pade” (2020), MATLAB® Documentation.
<https://www.mathworks.com/help/control/ref/pade.html>

Appendix A – Basler acA720-290gm Camera Calibration

The number of pixels corresponding to 1 mm was determined.



Figure A.1 Millimeter Ruler (Keyence)

The Keyence Digital Microscope was used to verify the distance on the mm ruler.

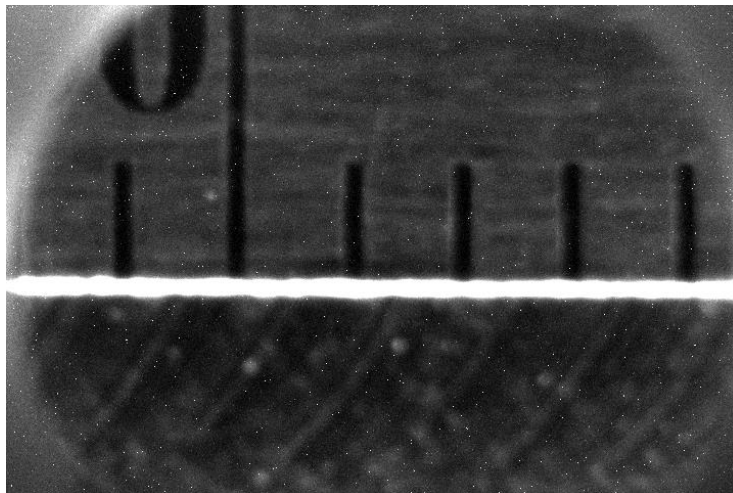


Figure A.2 Millimeter Ruler (Basler ace)

The image above was captured using the following camera attributes on NI MAX. (mm ruler)

Brightness: 0

Exposure: 2.0

Gain: 350

Resolution: 720 x 480

LENS® z-offset was -19.452 which is the recommended distance between the substrate (1/4” 316L steel plate) and powder nozzles. Using the Vision Assistant in the LabVIEW Vision Development Module the number of pixels corresponding to 1 mm was determined.

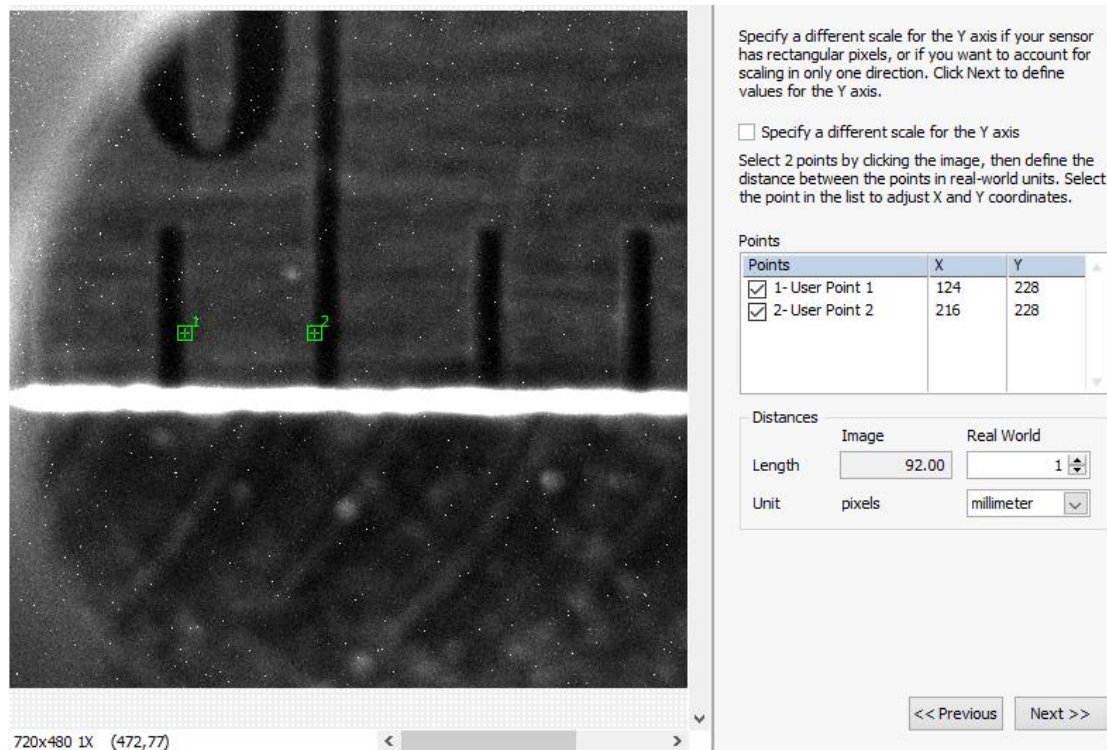


Figure A.3 Calibration

Two points were selection with the same y value. It was determined that 92 pixels corresponds to 1 mm in real-world units.

Appendix B – Threshold Value Determination

Multiple (1 in.) runs were conducted at varying power (200-400 W) in 50 W increments.

System parameters; powder feed rate 5 rpm, linear feed rate 19.2 ipm. Process was recorded at 150 fpm. Measurements of the track width were taken and compared to the corresponding images.

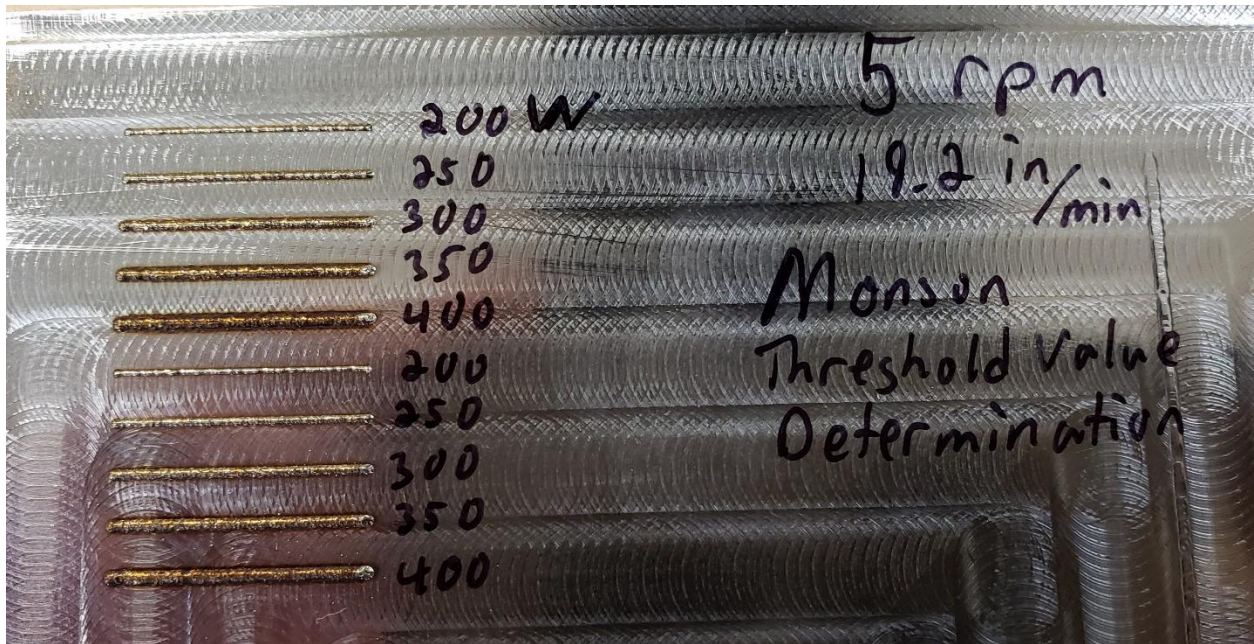


Figure B.1 Threshold Value Determination Runs



Figure B.2 200 W Track

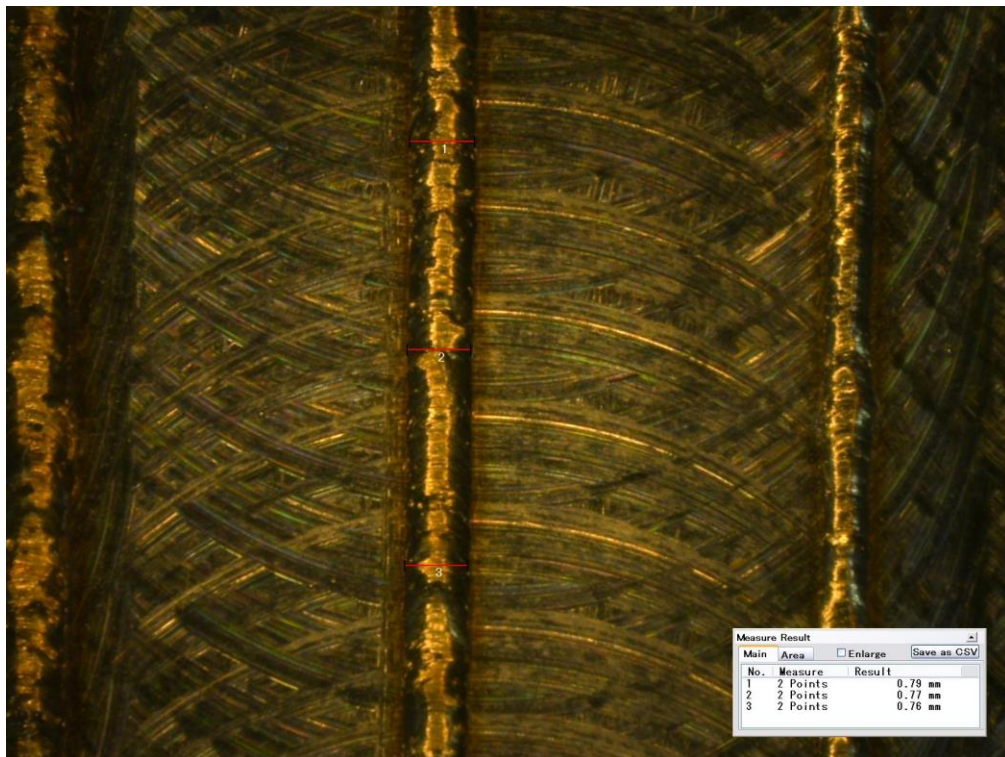


Figure B.3 250 W Track

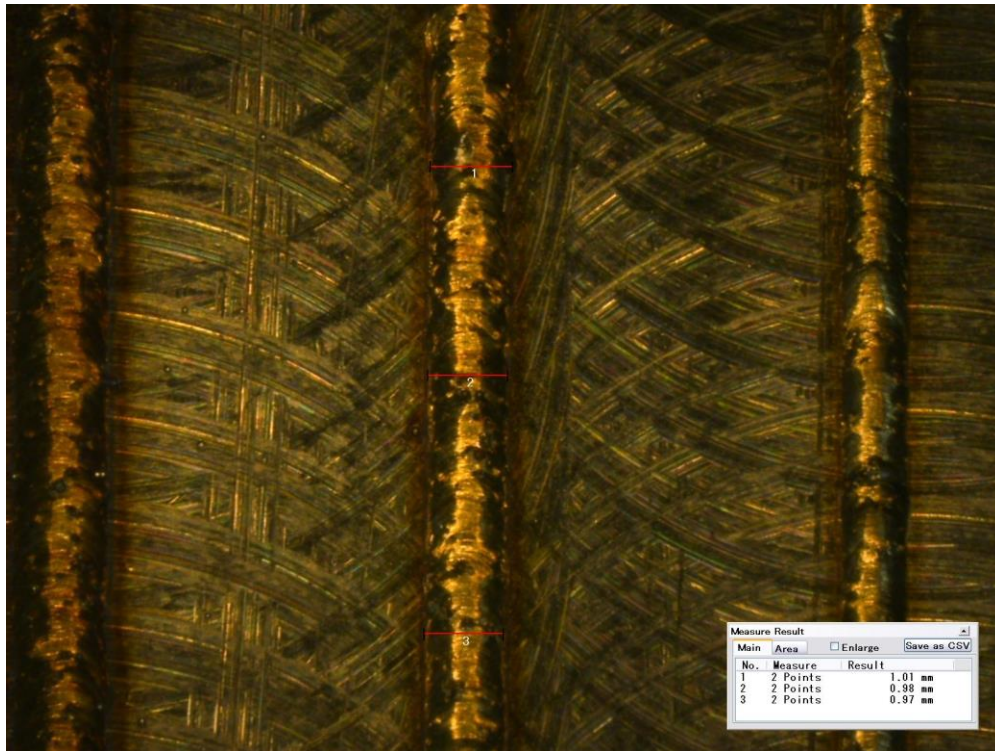


Figure B.4 300 W Track



Figure B.5 350 W Track

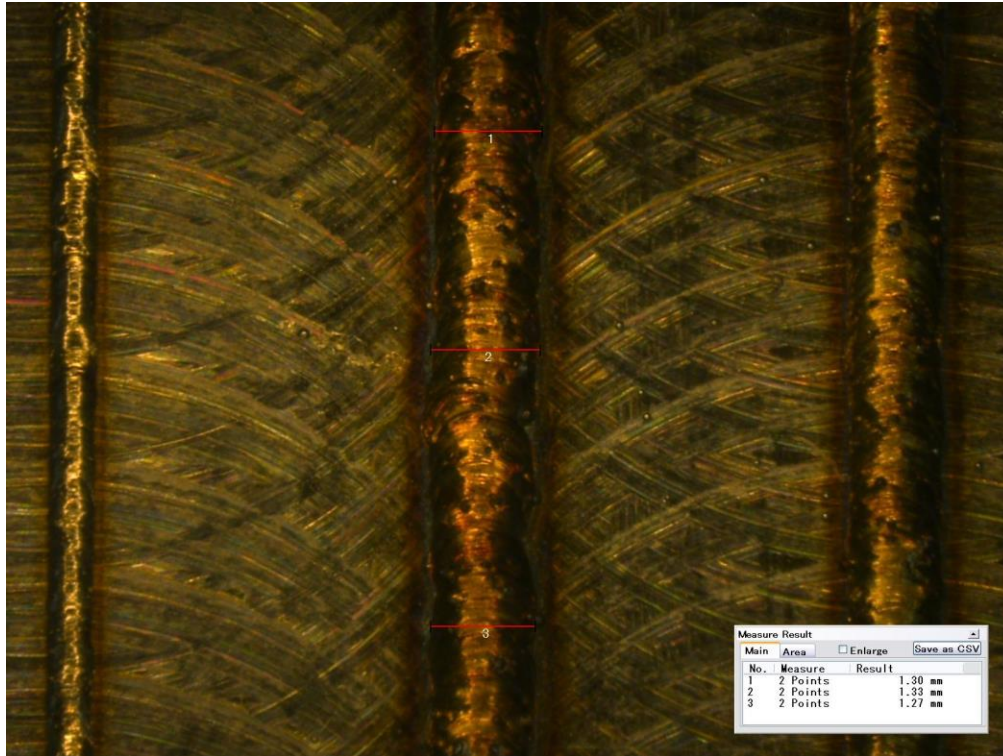


Figure B.6 400 W Track

Each track was measured in three locations and averaged. Using the image processing algorithm developed threshold values were selected by trial and error until an ideal value was found. The results are shown in Table B.1 with ideal threshold value being determined to be 50.

Table B.1 Threshold Value Determination Measurements

| Power (W) | Avg Measured Width (mm) | Avg Width LabVIEW (mm) | % Error |
|-----------|-------------------------|------------------------|---------|
| 200 | 0.53 | 0.55 | 3.80 |
| 250 | 0.77 | 0.76 | 1.32 |
| 300 | 0.99 | 0.94 | 4.21 |
| 350 | 1.13 | 1.07 | 5.61 |
| 400 | 1.30 | 1.29 | 0.78 |

Appendix C – LabVIEW Documentation

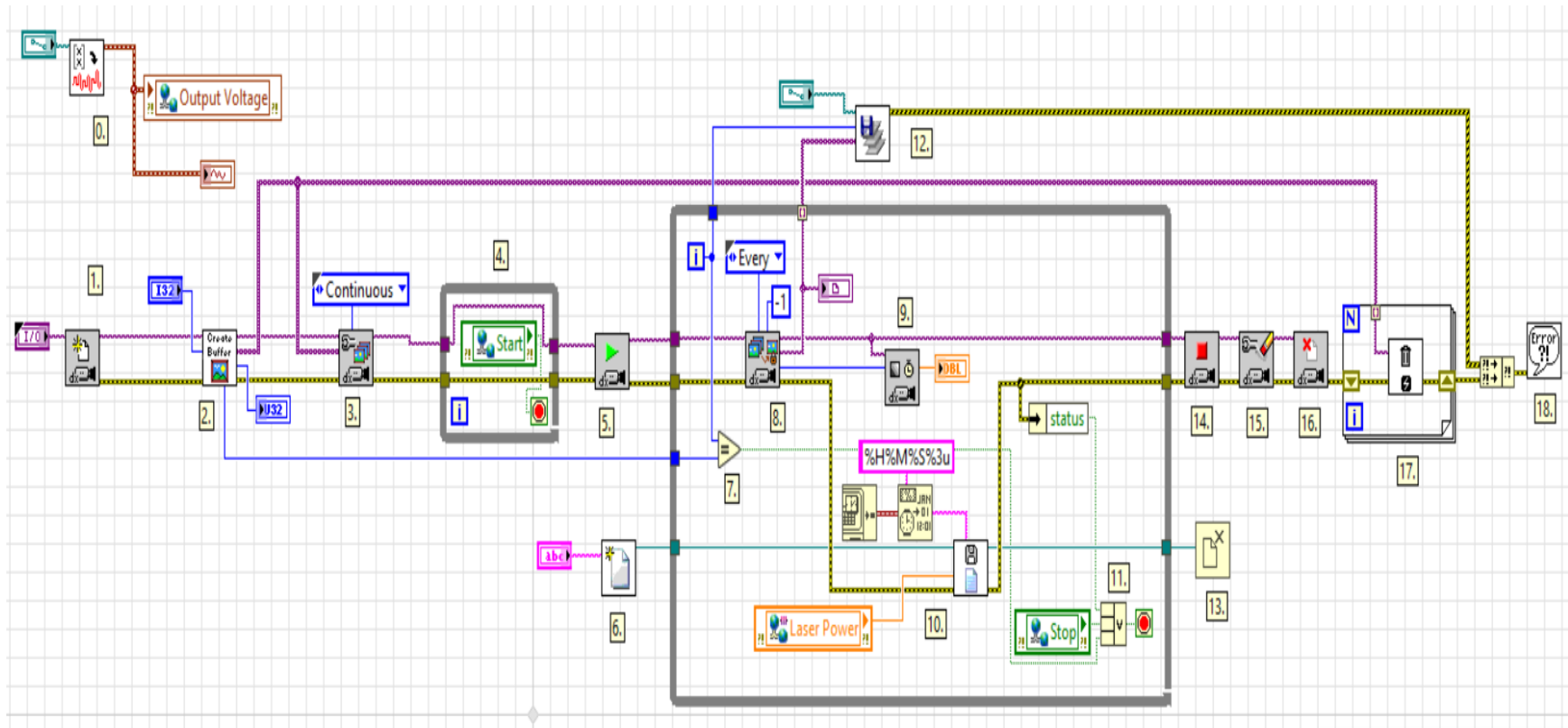


Figure C.1 Host PC Main VI

Steps:**Initialization**

0. Create output waveform. Arbitrary start.
1. IMAQdx Open Camera.vi: Open and configure the camera acquisition.
2. Create Buffer.vi (SubVI): Create a buffer to store captured images.
3. IMAQdx Configure Ring Acquisition.vi: Configures a ring acquisition.
A ring acq. is an acq. in which the img is acquired directly into the img in the Images array.
4. Start data acquisition. Network shared variable from cRIO_Main.vi
5. IMAQdx Start Acquisition.vi: Starts an acq that was previously config with the IMAQdx Config Acq VI.
6. Data File Setup.vi (SubVI): Create .txt file to record data.

Run

7. Will send F signal to stop While Loop when desired number of images are captured.
8. IMAQdx Extract Image.vi: Retrieves an acquired image directly from the circular buffer list, and does not make a copy of the image. Valid only within the context of a ring acquisition.
9. IMAQdx Calculate Frames per Second.vi: Calculates the acquired and and processing frame rate, the images behind, and images missed. Used to ensure data is gathered at a consistent frequency. Changes in capture rate can effect the dynamic model of a system during SID.
10. SaveDataFile.vi (SubVI): Saves gathered data to .txt file. Laser power network shared variable from cRIO_Main.vi. Records time stamp for each data point.
11. While Loop stopped from: error, clicking stop button on front panel, or when desired number of images captured.

Deinit

12. Save Images from Buffer.vi (SubVI): Saves images captured in buffer to disk as .tif files.
13. Close File: Closes an open file specified by the refnum and returns the path to the file associated with the refnum.
14. IMAQdx Stop Acquisition.vi: Stops an acq started with the IMAQdx Start Acquisition VI.
15. IMAQdx Unconfigure Acquisition.vi: Unconfigures an acq configured with the IMAQdx Configure Acquisition VI.
16. IMAQdx Close Camera.vi: Stops an acq in progress, releases resources associated with an acq, and closes the camera session
17. IMAQ Dispose: Dispose the image buffer.
18. Simple Error Handler.vi: Indicates whether an error occurred. If an error occurred this VI returns a description of the error and optionally displays a dialog box.

Figure C.2 Host PC Main VI Documentation

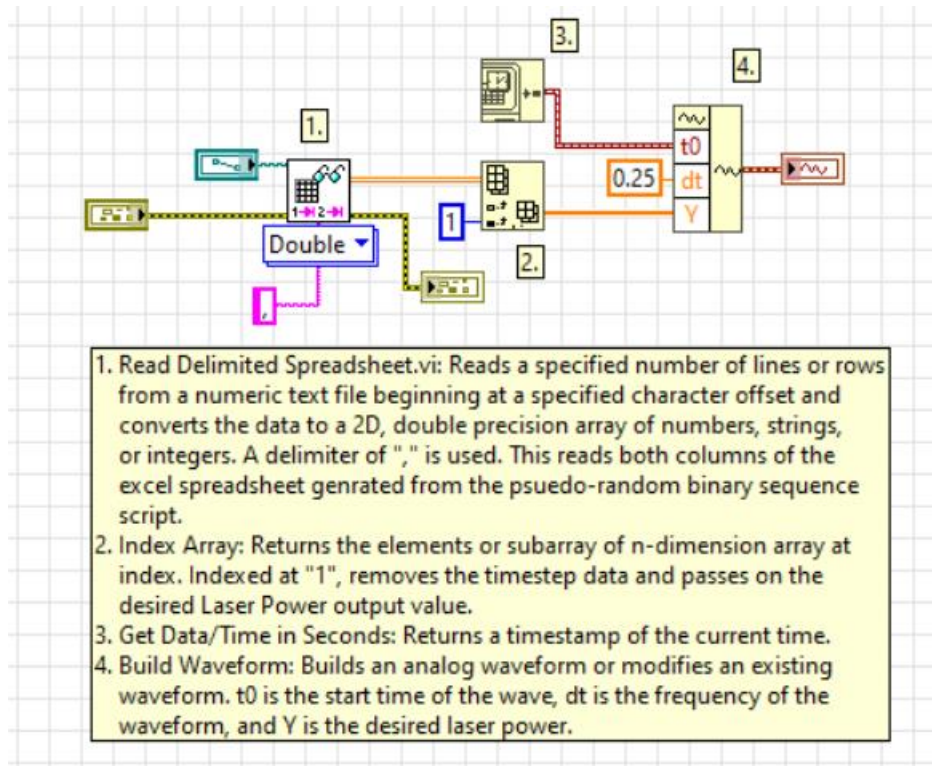


Figure C.3 Build Waveform VI

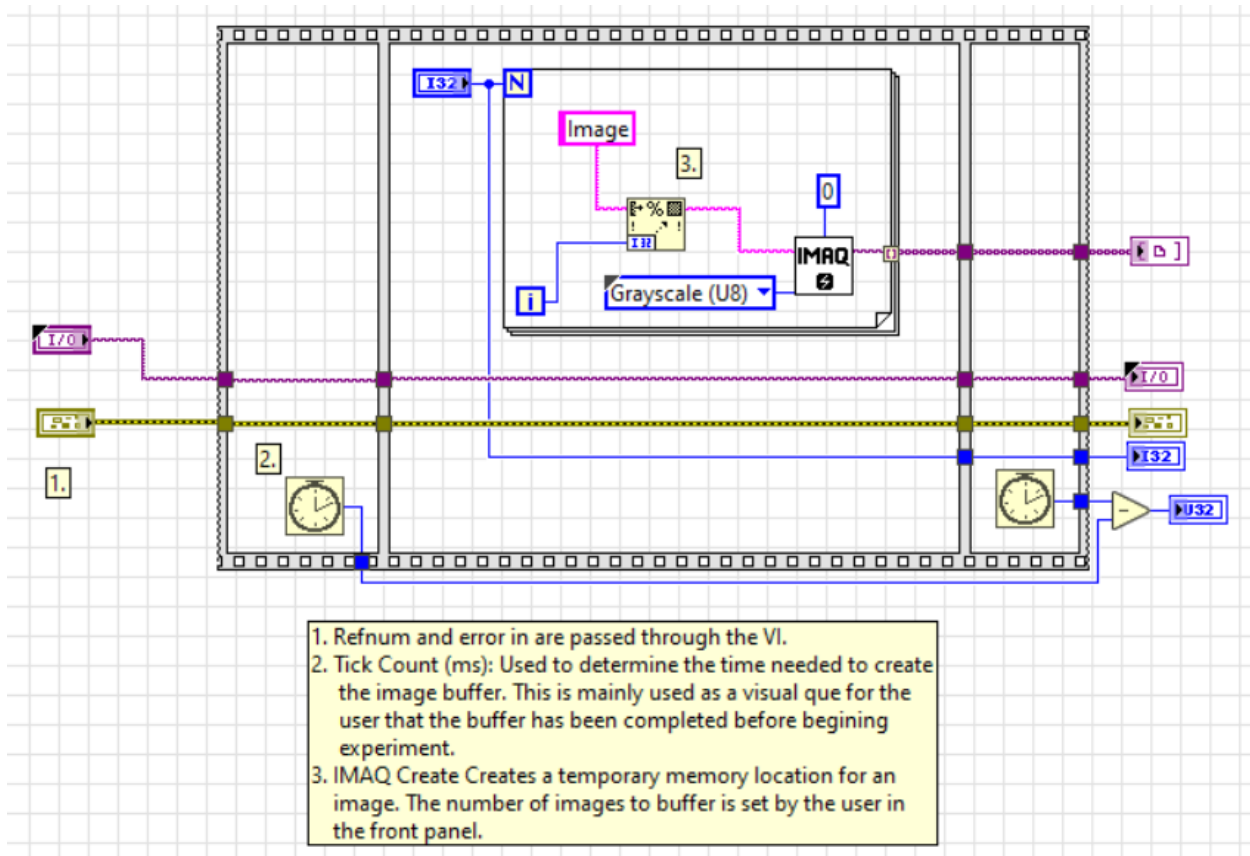


Figure C.4 Build Image Buffer VI

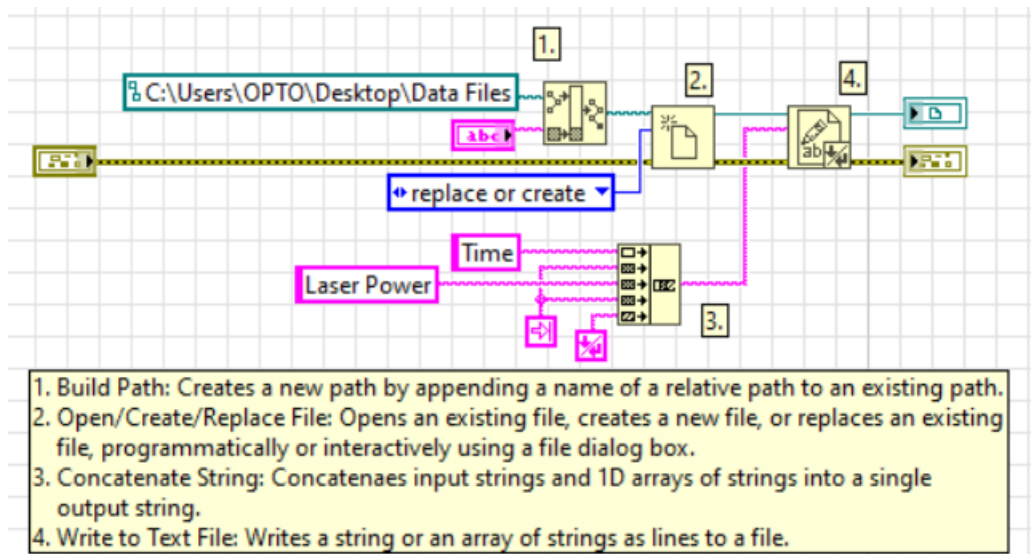


Figure C.5 Create Data File VI

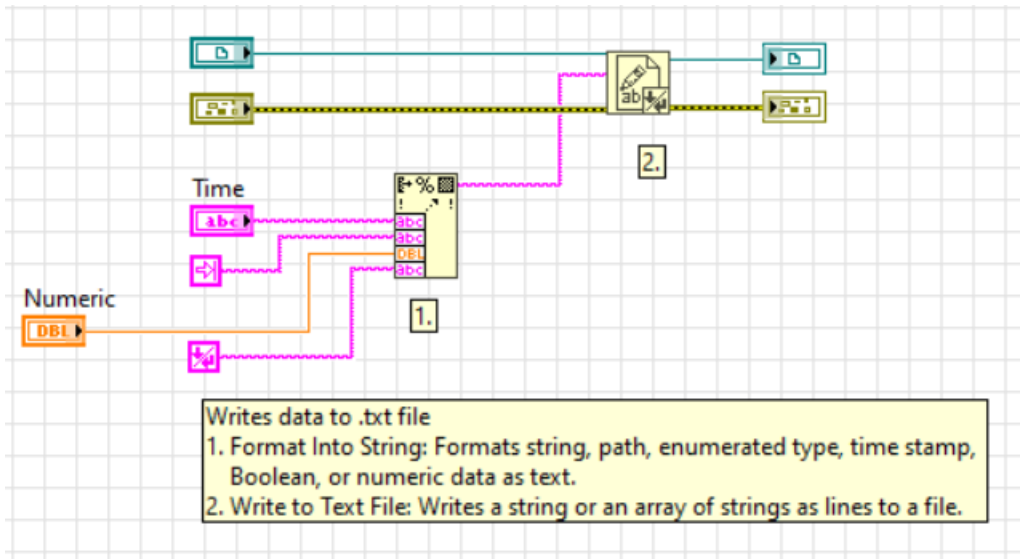


Figure C.6 Write Data File VI

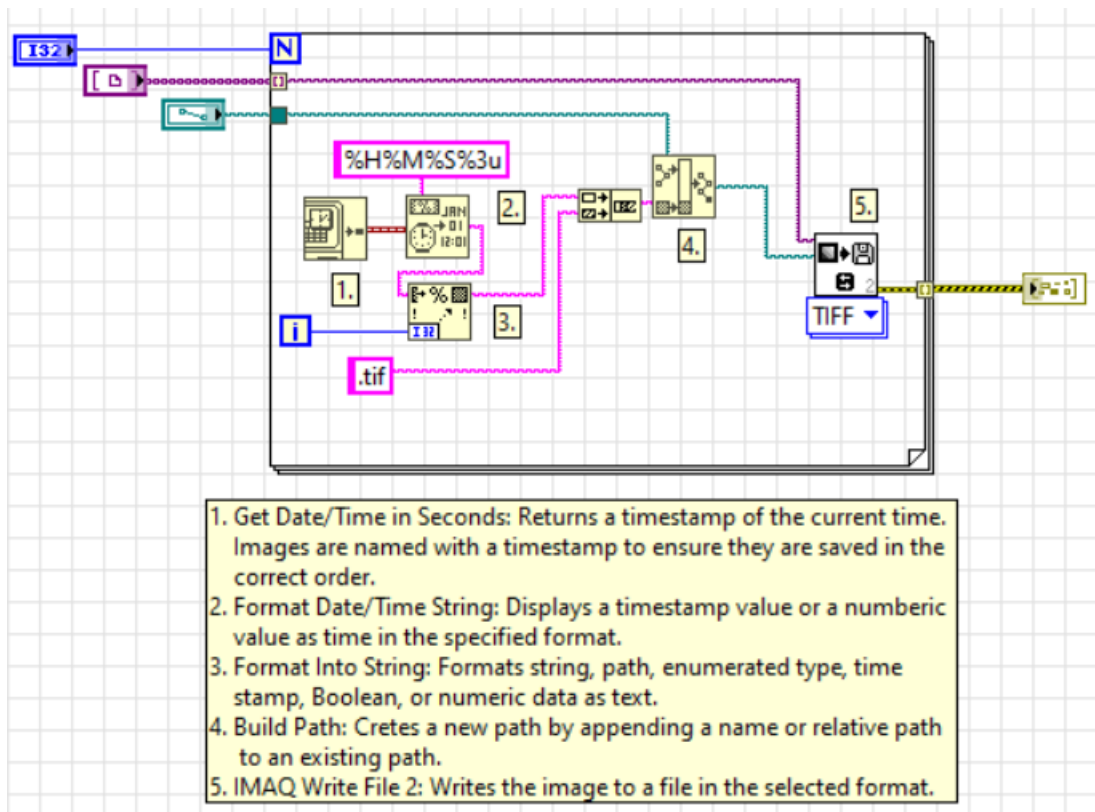


Figure C.7 Save Images to Disk VI

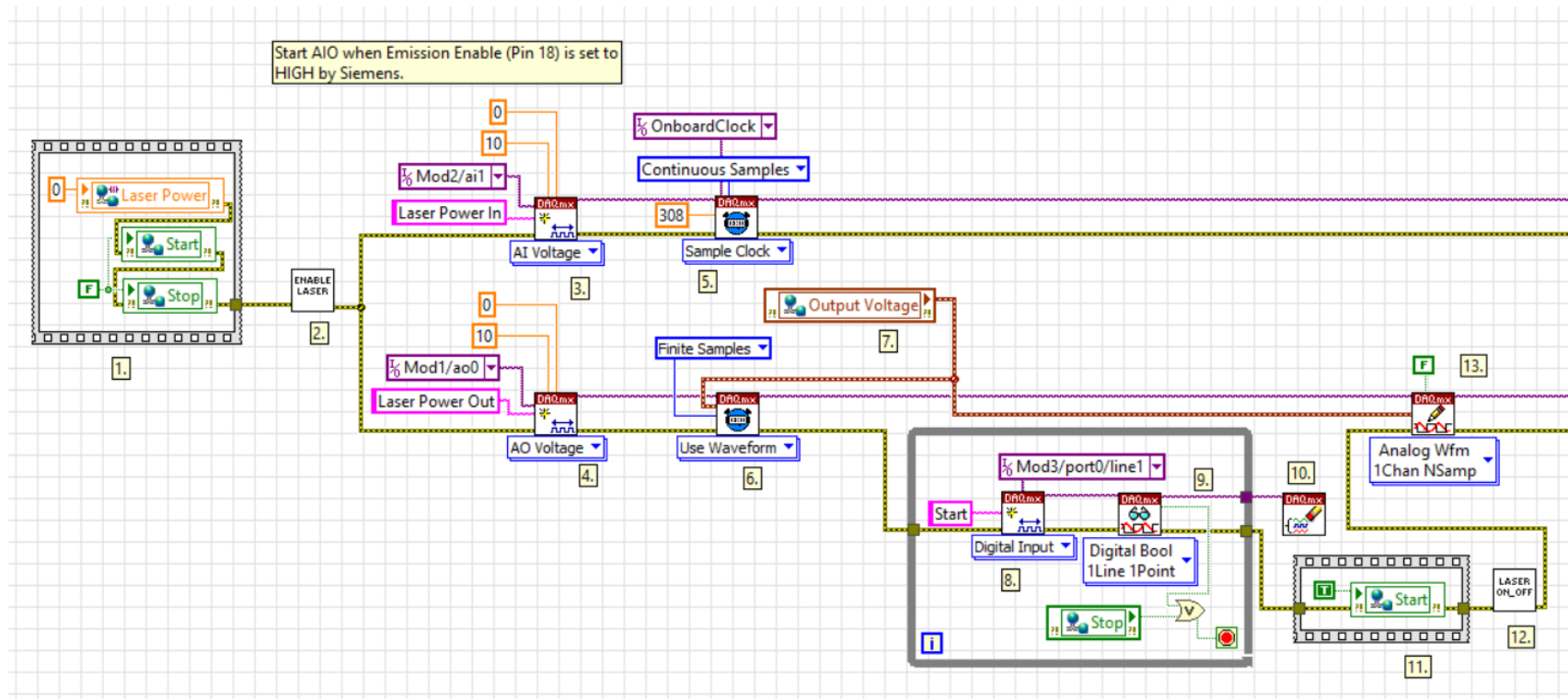


Figure C.8 cRIO Main VI

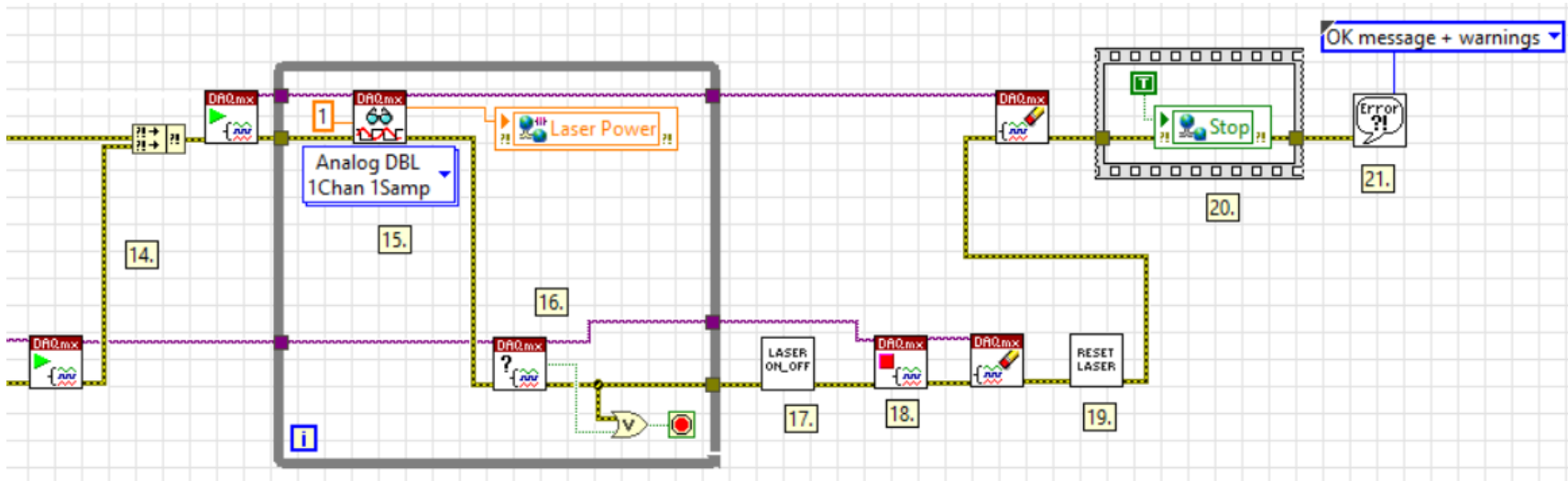


Figure C.9 cRIO Main VI (2)

1. Initialize network shared variable.
2. Enable Laser.vi (SubVI): Primes the laser to emit
3. DAQmx Create Channel (AI-Voltage-Basic).vi: Creates channel(s) to measure voltage.
4. DAQmx Create Channel (AO-Voltage-Basic).vi: Creates channel(s) to generate voltage.
5. DAQmx Timing (Sample Clock).vi: Sets the source of the Sample Clock, the rate of the Sample Clock, and the number of samples to acquire or generate.
6. DAQmx Timing (Use Waveform).vi: Uses the dt component of the waveform input to determine the Sample Clock rate. dt is the time in seconds between samples.
7. Network Shared Variable: Input waveform
8. DAQmx Create Channel (DI-Digital Input).vi: Creates channel(s) to measure digital signals. This While Loop will wait until the Laser ON digital signal is sent from the Siemens controller.
9. DAQmx Read (Digital Bool 1Line 1Point).vi: Reads a single Boolean sample from a task that contains a digital input channel composed of a single line. Stops the While Loop when the Laser ON digital signal is received from Siemens controller.
10. DAQmx Clear Task.vi: Clears the task. Aborts task if necessary and releases any resources the task reserved.
11. Reset the 'Start' variable to TRUE. This will begin data logging in Main.vi on the host PC.
12. Laser On_Off.vi (SubVI): Turns laser emission on.
13. DAQmx Write (Analog Wfm 1Chan NSamp).vi: Writes a waveform to a task that contains a single analog output channel.
14. DAQmx Start Task.vi: Transitions the task to the running state to begin the measurement or generation.
15. DAQmx Read (Analog DBL 1Chan 1Samp).vi: Reads a single floating point sample from a task that contains a single analog input channel. Reads the laser power return from the laser.
16. DAQmx Is Task Done.vi: Queries the status of the task and indicates if it completed execution. VI used to ensure that the specified operation is completed before stopping task. Task is complete when input waveform is finished.
17. Laser On_Off.vi (SubVI): Turns laser emission off.
18. DAQmx Stop Task.vi: Stops the task and returns it to previous state.
19. Reset Laser.vi (SubVI): Resets digital control signals to default values.
20. Network Shared Variable: Set value of variable 'Stop' to TRUE. This stops image acquisition and data logging.
21. Simple Error Handler.vi: Indicates whether an error occurred.

Figure C.10 cRIO Main VI Documentation

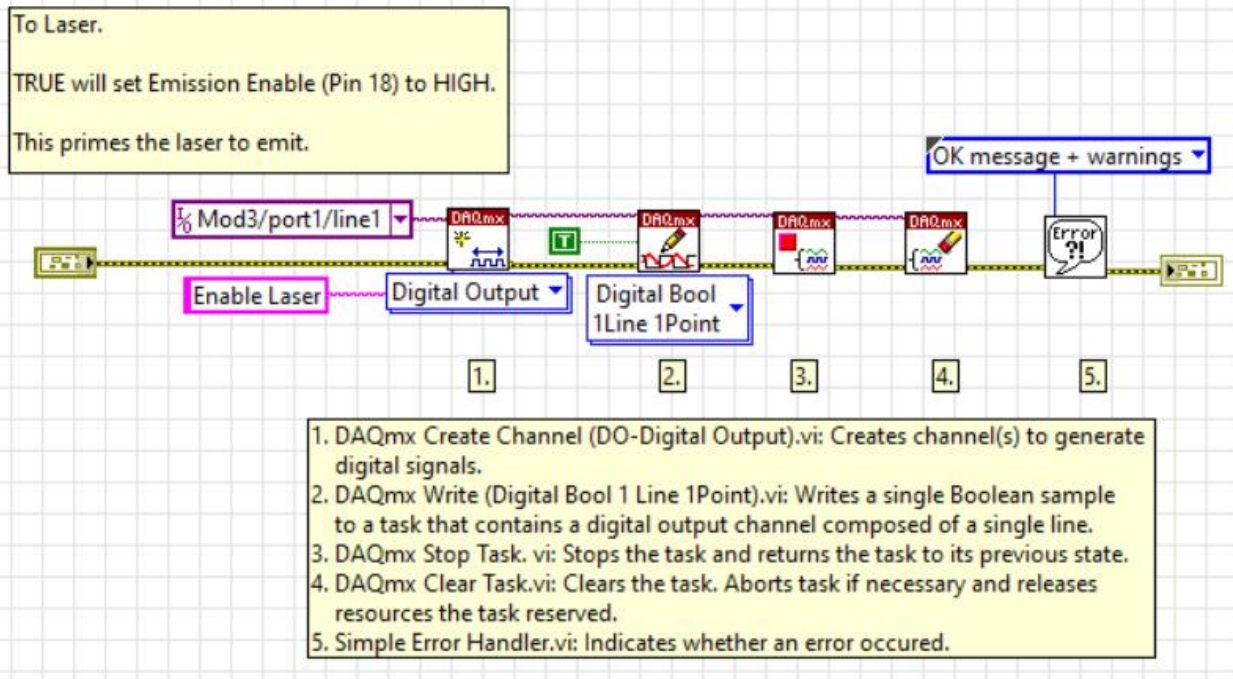


Figure C.11 Enable Laser VI

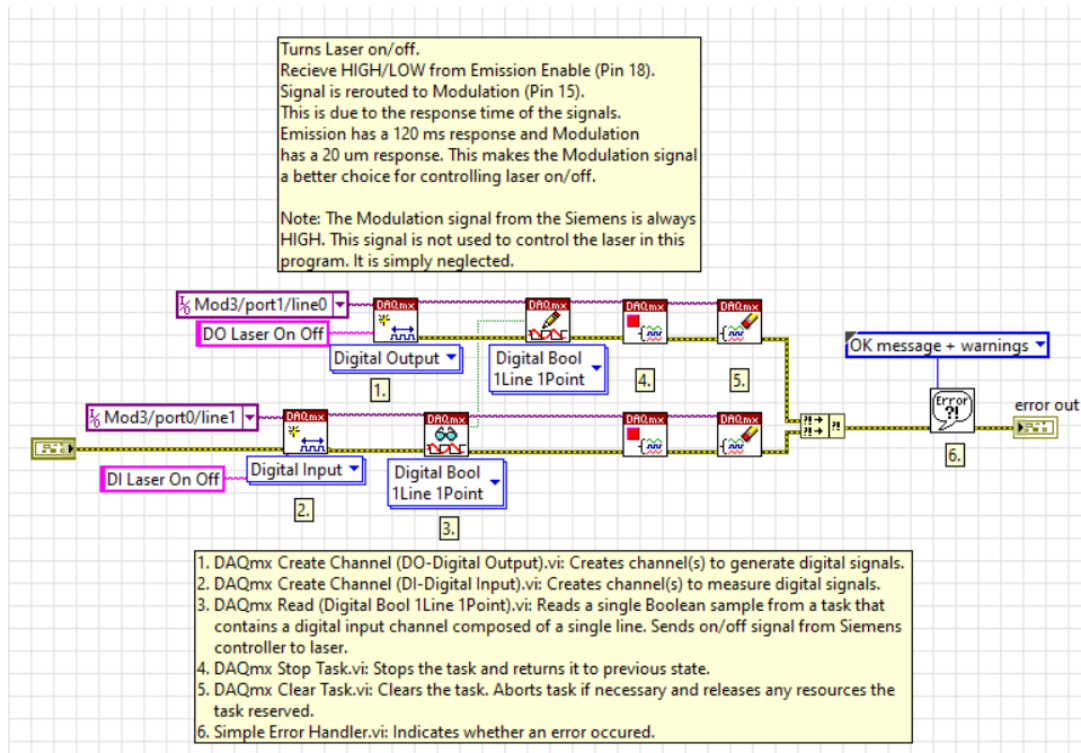


Figure C.12 Laser On/Off VI

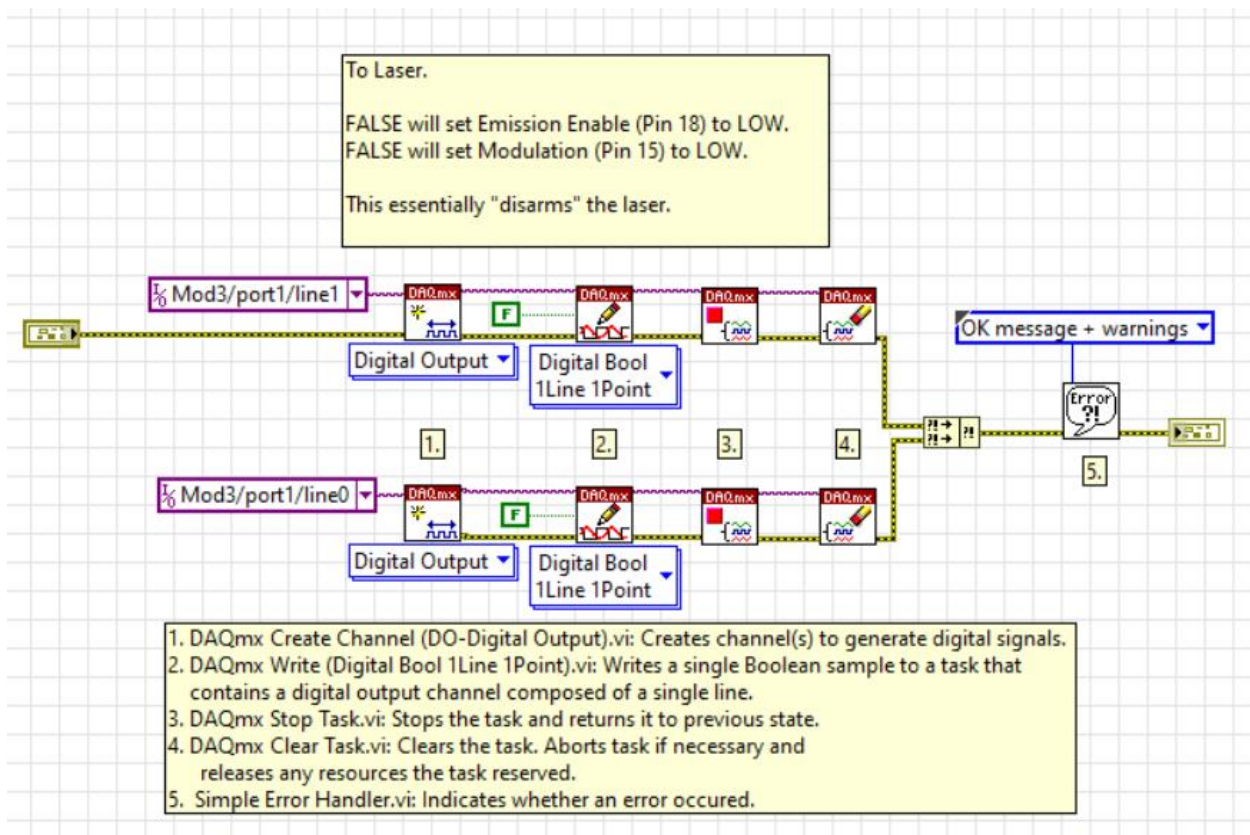


Figure C.13 Reset Laser VI

Appendix D – MATLAB® Scripts

```
%% Create data objects for system identification toolbox.
layer1 = iddata(DiameterLayer1,LaserPowerLayer1,0.0033);
layer2 = iddata(DiameterLayer2,LaserPowerLayer2,0.0033);
layer3 = iddata(DiameterLayer3,LaserPowerLayer3,0.0033);
layer4 = iddata(DiameterLayer4,LaserPowerLayer4,0.0033);
layer5 = iddata(DiameterLayer5,LaserPowerLayer5,0.0033);
layer6 = iddata(DiameterLayer6,LaserPowerLayer6,0.0033);
layer7 = iddata(DiameterLayer7,LaserPowerLayer7,0.0033);
layer8 = iddata(DiameterLayer8,LaserPowerLayer8,0.0033);
layer9 = iddata(DiameterLayer9,LaserPowerLayer9,0.0033);
layer10 = iddata(DiameterLayer10,LaserPowerLayer10,0.0033);



---



%% Generate plots of layer data and layer model.
% Layer model parameters
Kp = [1.0554 1.0772 1.1064 1.1337 1.2076 1.2107 1.2166 1.2272 1.2761 1.2869];
x = 1:1:10;
Tp = [0.254 0.211 0.230 0.210 0.192 0.174 0.190 0.166 0.158 0.198];

% Plot layer model parameters.
figure
subplot(2,1,1)
plot(x,Kp)
grid on
title('Static Gain')
xlabel('Layer')
ylabel('Kp')
subplot(2,1,2)
plot(x,Tp)
grid on
title('Time Constant')
xlabel('Layer')
ylabel('Tp')

% Plot layer data
figure(2)
title('Melt Pool Diameter')
subplot(4,1,1)
plot(DiameterLayer1);hold on;plot(LaserPowerLayer1)
grid on
title('Layer 1')
subplot(4,1,2)
plot(DiameterLayer2);hold on;plot(LaserPowerLayer2)
grid on
title('Layer 2')
subplot(4,1,3)
plot(DiameterLayer2);hold on;plot(LaserPowerLayer2)
grid on
title('Layer 3')
subplot(4,1,4)
plot(DiameterLayer2);hold on;plot(LaserPowerLayer2)
```

```

grid on
title('Layer 4')
legend('Diameter','Laser Power')

figure(3)
title('Melt Pool Diameter')
subplot(4,1,1)
plot(DiameterLayer5);hold on;plot(LaserPowerLayer5)
grid on
title('Layer 5')
subplot(4,1,2)
plot(DiameterLayer6);hold on;plot(LaserPowerLayer6)
grid on
title('Layer 6')
subplot(4,1,3)
plot(DiameterLayer7);hold on;plot(LaserPowerLayer7)
grid on
title('Layer 7')
subplot(4,1,4)
plot(DiameterLayer8);hold on;plot(LaserPowerLayer8)
grid on
title('Layer 8')
legend('Diameter','Laser Power')

figure(4)
title('Melt Pool Diameter')
subplot(4,1,1)
plot(DiameterLayer9);hold on;plot(LaserPowerLayer9)
grid on
title('Layer 9')
subplot(4,1,2)
plot(DiameterLayer10);hold on;plot(LaserPowerLayer10)
grid on
title('Layer 10')
legend('Diameter','Laser Power')

```

```

%% Create Pade approximation of each layer.
% Create transfer function.
s = tf('s');

% Create transcendental fucntion exp(-sT_d).
sys = exp(-0.099*s);

% Specify order of Pade approximation.
sysx = pade(sys,3);

% Generate each layer model.
sys1 = tf(1.055,[0.25385 1])*sysx;
sys2 = tf(1.0772,[0.21084 1])*sysx;
sys3 = tf(1.1064,[0.22969 1])*sysx;
sys4 = tf(1.1337,[0.2095 1])*sysx;
sys5 = tf(1.2076,[1.3051 1])*pade(exp(-0.0122*s),3);
sys6 = tf(1.2107,[0.1739 1])*sysx;
sys7 = tf(1.2166,[0.19029 1])*sysx;

```

```

sys8 = tf(1.2272,[0.16571 1])*sysx;
sys9 = tf(1.2761,[0.15849 1])*sysx;
sys10 = tf(1.2869,[0.19788 1])*sysx;

```

```

%% Tune Mamdani Fuzzy Inference System
% Tunes membership functions (MF) and rule parameters of a Mamdani fuzzy
% inference system (FIS). Uses particle swarm optimization and pattern
% search optimization.

% Load the data.
load('DiameterLayer1.mat')
load('LaserPowerlayer1.mat')

% Name input and output variable.
inname = ["e", "e_dot"];
outname = ["Kp", "Ki"];

% Create input and output data sets.
X = LaserPowerLayer1;
Y = DiameterLayer1;
data = horzcat(LaserPowerLayer1,DiameterLayer1);

% Partition data sets into training data and validation data.
trnX = X(1:2:end,:);
trnX(862:900) = trnX(861);
trnY = Y(1:2:end,:);
trnY(862:900) = trnY(861);

% Extract the range of each data attribute.
dataRange = [min(data)' max(data)'];

% Create a Mamdani FIS.
fisin = mamfis;

% Create input and output variable.
for i = 1:2
    fisin = addInput(fisin,[-100 100], 'Name', inname(i), 'NumMFs', 3);
end
for i = 1:2
    fisin = addOutput(fisin,[-100 100], 'Name', outname(i), 'NumMFs', 9);
end

%% Tune FIS with training data.
% Set tuning options. Method is particle swarm, optimization type is
% learning new rules, restrict maximum number of rules. Number of rules
% will be less then this value.
options =
tunefisOptions('Method','particleswarm','OptimizationType','learning','NumMax
Rules',9);

% Set number of iterations.
options.MethodOptions.MaxIterations = 20;
rng('default')

```

```

% Tune FIS.
fisout1 = tune fis(fisin,[],trnX,trnY,options);

% After learning rules, tune I/O MF.
[in,out,rule] = getTunableSettings(fisout1);
options.OptimizationType = 'tuning';
options.Method = 'patternsearch';
options.MethodOptions.MaxIterations = 40;
options.MethodOptions.UseCompletePoll = true;
rng('default')
fisout = tune fis(fisout1,[in;out;rule],trnX,trnY,options);

```