

**PREDICTING MECHANICAL PROPERTIES
OF FFF PARTS**

Gerardo A. Mazzei Capote

A dissertation submitted in partial fulfillment of
the requirements for the degree of

**Doctor of Philosophy
(Mechanical Engineering)**

at the

UNIVERSITY OF WISCONSIN-MADISON

2021

Approval

The following dissertation, **Predicting Mechanical Properties of FFF parts**, developed at the **University of Wisconsin-Madison** has been approved by:

Signature

Date

Professor Tim A. Osswald
Department of Mechanical Engineering
College of Engineering
University of Wisconsin-Madison

Abstract

Fused Filament Fabrication (FFF) is arguably the most widely available Additive Manufacturing technology at the moment. Offering the possibility of producing complex geometries in a compressed product development cycle and in a plethora of materials, it comes as no surprise that FFF is attractive to multiple industries, including the automotive and aerospace segments. However, the high anisotropy of parts developed through this technique imply that part failure prediction is extremely difficult —a requirement that must be satisfied to guarantee the safety of the final user. Application of a Failure Criterion to predict part failure has been shown to constitute a solution to this problem. However, specialized printing equipment, and a large number of mechanical tests performed under a variety of loading conditions are required to populate the parameters of the failure function - a process that is extremely time consuming and can prove unfeasible if off-axis printing solutions are not available to the user. This research proposal describes a method by which certain mechanical properties of an FFF part can be predicted using machine learning methods. Data extracted from an FFF printer fitted with in-line sensors that capture extrusion force and velocity, as well as additional data stemming from μ CT scans, dimensional changes in the filament geometry, and mechanical tests can be used to train a machine learning system that can predict the expected mechanical performance of an FFF part under certain loading conditions. This resource can significantly reduce the time required to produce a failure envelope for FFF parts, as well as allowing a better comprehension of the relationship between process variables and final mechanical properties. Additionally, such resources clear the path for development of intelligent equipment that can detect flaws mid-print and auto-correct based on the expected performance of the part.

Keywords: FFF, FDM, Failure Criteria, Mechanical Testing, Machine Learning.

Acknowledgments

Thank you Prof. Rudolph and Prof. Osswald for your trust and patience. I couldn't hope for better advisors.

Thank you to Prof. Turng and Prof. Prabhakar for your valuable input and time.

Thank you Tom for your extrusion expertise and your polymer knowledge. The sass was OK too I suppose.

Thank you Luke for building the tools necessary for this whole project to work. That big beard hides a sharp mind -and a lot of patience to answer all my random grammar questions.

Thank you Alec for being an ever-present helping hand, a great classmate, and a good friend. Hope you make ND proud.

Thank you José for putting up with my constant interruptions *en español* from across the desk, and for sharing lab managing duties with me. *Suerte y paciencia para el doctorado.*

Thank you to Thibaut, Colby, Johannes, Peilin, Varun, Ahmed and Brendan. This thesis would look very empty without your contributions!

Thank you to the entire PEC family for creating a great learning environment. I've grown so much with all of you. I hope that you've learned a thing or two with me as well.

Thank you to Ben, Josh, Diego, Iván, Pfeifer, Ian, Abrahan, Maria, Sianna, Conny and Sara for your friendship. Madison was a lot of fun because you were in it.

Gracias a mi familia, a quienes les debo todo.

Gracias a Sonya por tu cariño y paciencia. Que este tiempo que pasamos tan lejos nos traiga en creces momentos juntos. Te amo.

Table of Contents

Front Matter	i
Abstract	i
Acknowledgments	ii
Nomenclature	vi
List of Figures	viii
List of Tables	ix
Introduction	1
1 Background	3
1.1 Additive Manufacturing	3
1.1.1 Advantages and Disadvantages of AM	4
1.2 Fused Filament Fabrication	5
1.2.1 The FFF process	6
1.2.2 Mechanical Properties of FFF parts	8
1.3 Failure Criteria	9
1.3.1 The Stress-Stress Interaction Criterion	10
1.4 Development of SSIC envelope for FFF parts	15
1.5 Machine Learning Techniques	19
2 Part Structural Integrity Evaluation through Failure Criteria	21
2.1 Foreword	21
2.2 Introduction	21
2.3 Experimental Methods	23
2.4 Results	24
2.5 Conclusions	24
3 Trends in Print Force and Extrusion Speed	27
3.1 Foreword	27
3.2 Introduction	27
3.3 Equipment and Materials	30
3.3.1 Printer Setup	30
3.3.2 Materials	31

3.4	Design of Experiments	33
3.5	Results	34
3.5.1	Considerations Derived from Preliminary Tests	34
3.5.2	Results from Modified Tests	37
3.6	Conclusions	44
4	Prediction of Mechanical Properties through Machine Learning	45
4.1	Foreword	45
4.2	Introduction	46
4.3	Experimental Methods	50
4.3.1	Design of Experiments	50
4.3.2	Equipment and methods	50
4.3.3	Experimental Results	54
4.3.4	Statistical Analysis	56
4.3.5	Neural Network Architecture	58
Bibliography		61
A	SABIC Cycloc MG94 Datasheet	67
B	Failure Criterion Calculation Code	69
C	SSIC Predictions Code	71
D	Data Acquisition Code	73
E	Database Assembly Code	78

Symbols and Acronyms

Acronyms

μ CT Micro Computer Tomography

ABS Acrylonitrile Butadiene Styrene

AI Artificial Intelligence

AM Additive Manufacturing

CAD Computer Aided Design

FC Failure Criterion

FDM Fused Deposition ModelingTM

FFF Fused Filament Fabrication

FNM Melt Filled Nozzle Model

GKC Gol'denblat-Kopnov Criterion

MAE Mean Absolute Error

ME Material Extrusion

MFM Melt Film Model

MJF Multi-Jet Fusion

ML Machine Learning

MSE Mean Square Error

NN Neural Network

PA12 Polyamide 12

PBF Powder Bed Fusion

RP	Rapid Prototyping
SLA	Stereolithography
SLS	Selective Laser Sintering
SSIC	Stress-Stress Interaction Criterion
SVM	Support Vector Machines

Symbols

μ^{1112}	SSIC parameter- slope at pure shear failure in the $\sigma_{11} - \tau_{12}$ plane	—
μ^{2212}	SSIC parameter- slope at pure shear failure in the $\sigma_{22} - \tau_{12}$ plane	—
σ	Axial stress	MPa
σ_{11}	Axial stress in the 1-1 direction	MPa
σ_{22}	Axial stress in the 2-2 direction	MPa
σ_{33}	Axial stress in the 3-3 direction	MPa
τ	Shear stress	MPa
τ_{12}	Shear stress in the 1-2 plane	MPa
τ_{13}	Shear stress in the 1-3 plane	MPa
τ_{23}	Shear stress in the 2-3 plane	MPa
S	Shear strength in the 1-2 plane	MPa
S_{45n}	Negative shear strength for 45° specimen	MPa
S_{45p}	Positive shear strength for 45° specimen	MPa
X_c	Compressive strength in the 1-1 direction	MPa
X_t	Tensile strength in the 1-1 direction	MPa
Y_c	Compressive strength in the 2-2 direction	MPa
Y_t	Tensile strength in the 2-2 direction	MPa

List of Figures

1.1	Process flow of AM	3
1.2	The basic FFF machine configuration	6
1.3	Model, toolpath and final part in the FFF process	7
1.4	Typical FFF part mesostructure and its origin	7
1.5	Results from Koch <i>et al.</i> [6]	9
1.6	Comparison of different failure criteria. [31]	10
1.7	Different load directions in an anisotropic material	11
1.8	Interaction slopes available through the SSIC	14
1.9	Failure surface for SLS developed through the SSIC [8]	15
1.10	Comparison of the $\sigma_{22} - \tau_{12}$ interaction for SLS and MJF PA12 parts [34]	16
1.11	failure envelope in the $\sigma_{11}-\sigma_{22}$ plane	17
1.12	Comparison of interaction slopes in the axial-transverse stress planes .	18
1.13	Differences between traditional programming and machine learning. [12]	19
1.14	Taxonomy of ML applications in AM [15]	20
2.1	Off-axis loading conditions	22
2.2	Experimental results from tensile tests	25
2.3	Results from [40]	25
3.1	Bellini model (left) and Osswald, Puentes, Kettinger model (right) [43].	28
3.2	Assembly shown on the left. Schematic of sensor placement on the right.	31
3.3	Extrusion line used to produce ABS filament	32
3.4	Shear rate dependency for a) PLA and b) ABS	32
3.5	Helical cylinder experiment	33
3.6	Raw signal data length (bottom), derived speed (middle), and force (top).	35
3.7	Comparison of processed and unprocessed filament speed data.	35
3.8	Comparison of processed and unprocessed filament speed vs. force within the nozzle of an FFF printer. PLA printed at 230°C.	36
3.9	Alpha plot for 5 trials of ABS printed at 230°C.	38
3.10	Alpha plot for 5 trials of ABS printed at 245°C.	39
3.11	Alpha plot for 5 trials of PLA printed at 230°C.	40
3.12	Comparison of PLA and ABS Force-velocity pairings at a printing temperature of 230°C	40

3.13 Alpha plot for 5 trials of PLA printed at 245°C.	41
3.14 Comparison of experimental data for ABS printed at 230°C with predictions stemming from various ME melting models.	43
4.1 Differences between traditional programming and machine learning. [12]	46
4.2 Schematic of a NN [50]	47
4.3 Iterative process of NN parameter tuning [12]	48
4.4 Schematic of modified ME printer with sensors	51
4.5 Schematic of print experiment	52
4.6 Filament geometry information, acquired through a laser micrometer .	52
4.7 Tensile testing setup	53
4.8 Effect of filament geometry on extrusion force	55
4.9 Preliminary architecture for the NN	59

List of Tables

1.1	Advantages and Disadvantages of Additive Manufacturing	5
1.2	Nomenclature of the GKC parameters	12
1.3	Tensorial components of the GKC	13
1.4	Interaction components attainable through the SSIC [11]	14
2.1	Printing parameters maintained constant.	23
3.1	Number of layers per print head movement speed.	37
3.2	Average print temperature for ABS 245°C trials.	38
3.3	Average print temperature for PLA 245°C trials.	41
3.4	Material properties for ABS MG94 [35].	42
4.1	Controlled variables in the Design of Experiments	50
4.2	Pearson Correlation Matrix	57
4.3	Summary ANOVA table of 0° experiments	57
4.4	Summary ANOVA table of 90° experiments	58

Introduction

Additive Manufacturing (AM) is an umbrella term that encompasses all fabrication techniques where the final geometry of the part is obtained through superposition of material in a layer-by-layer basis [1]. Developed in the 1980s, this manufacturing technique permits immensely shorter part development cycles, since the transition from a 3D *Computer Aided Design* (CAD) to part fabrication only requires one intermediate step: the use of a slicing engine that converts the geometry of the object into machine instructions [1]. For this reason, AM technologies were initially employed exclusively for prototype development and were referred to as *Rapid Prototyping techniques* (RP). However, recent innovations in the field have caused AM to be considered as a legitimate manufacturing technology since it is also capable of reproducing complex geometries unattainable through traditional methods [1].

While offering great advantages over traditional part fabrication methods, AM comes with its own set of limitations and disadvantages: First and foremost, the use of a stratified build approach tends to produce extremely anisotropic parts. Secondly, the geometric accuracy of the object produced is highly dependent of process parameters, particularly, the thickness of the layers. Finally, as of the time of this writing, AM lacks the standardization and scrutiny that are associated to most traditional manufacturing techniques [1].

Fused Filament Fabrication (FFF), also known under the trademark *Fused Deposition Modeling* (FDM™), represents perhaps the most prevalent AM technique in the market due to the advent of low-cost, desktop 3D printers in the early 2010s [2]. Due to the broad availability of machines and relatively low costs of material, there is a surging interest in optimizing FFF to produce small batches of end-user grade parts. Success stories are varied, but examples include vacuum form molds, fixtures, jigs, and tools used to aid assembly lines in the automotive industry [3, 4, 5]. However, this technology still faces the challenges and limitations that currently affect the field of AM as a whole. Namely, anisotropy introduced through the layer-by-layer build approach makes it difficult to assess the expected mechanical behavior of FFF parts when subjected to important mechanical stresses [2]. For these reasons, multiple attempts have been made to characterize the anisotropy of FFF manufactured objects, such as the studies performed by Koch *et al.* [6] and Rankouhi *et al.* [7], which show that the ultimate tensile strength of FFF coupons is sensitive to process parameters such as the layer thickness and, in particular, the orientation in which the plastic strands

are laid during the build process -henceforth referred to as the bead orientation. Literature related to preventing failure through predictive methods in the design stages is scarce. However, a handful of publications exist where this issue was solved through the application of a failure criterion. The reach of this methodology has been fairly limited, given the difficulty of using commercially available AM machines to produce test coupons with unconventional bead orientations necessary to populate the failure surface, as well as the limitations inherent to development of failure criteria. Examples include the developments of failure envelopes for *Polyamide 12* (PA12) used in *Selective Laser Sintering* (SLS) and Multi-Jet Fusion (MJF) [8, 9], and more importantly for this body of work, a failure surface for *Acrylonitrile Butadiene Styrene* (ABS) used in FFF [10]. For the latter, certain test specimens in unconventional configurations had to be produced using a unique off-axis 3D printer developed in-house. In both cases, the researchers utilized a FC that incorporates stress interactions into the calculations of the failure surface, a feature that more recognized criteria, such as the Tsai-Wu model fail to take into account [11].

Additional predictive tools have been pushed to the forefront of engineering applications given recent developments in the fields of statistics, data science, artificial intelligence, worldwide connectivity, and computational hardware. These tools allow designing intelligent systems that can, among many things, detect and correct problems during a production run, identify trends, and more importantly for the objective of this work, predict outcomes or perform classification tasks. These tools have been grouped under the *Machine Learning* (ML) moniker, and are currently being exploited by large companies to make sense of large clusters of data. Machine Learning tools thrive in cases where the inputs and outcomes of a particular phenomena or task are known, but connecting the two through an explicit set of rules or relationships can result extremely complex and time consuming [12] because, in simple terms, ML models are trained, as opposed to explicitly programmed. Their broad range of applications has caused its use to trickle into other segments of engineering, usually in the form of *Neural Networks* or *Support Vector Machines* performing a variety of regression analysis or classification tasks. The field of AM is no stranger to the ML topic. Interest in the subject has been remarked by several authors [13, 14, 15], and it has even been successfully applied to predict certain properties of AM parts produced under various techniques [13, 14, 15, 16].

The set of printing conditions that lead to an optimal part in terms of mechanical properties aren't still fully comprehended and result in extremely complex, multi-variable relations. However, an FFF machine with in-line sensors that allowed monitoring a variety of process-variables, as well as data generated from mechanical tests and ancillary experiments would constitute a perfect case for deployment of a Machine Learning system capable of predicting the mechanical properties of the finished part.

This work apply strategies that allow engineers to solve the issue of unpredictable mechanical behavior of FFF parts — namely the validation of a safety threshold attained through application of failure criteria, and ML techniques to the FFF process in order to predict mechanical properties according to in-line measurements.

1 Background

1.1 Additive Manufacturing

Additive Manufacturing (AM) technologies had their beginnings in the decade of the 1980s. During this time, various independently developed patents were filed across the globe, describing a process that would construct an object by selectively adding layers of material -as opposed to removing excess matter or deforming mass to obtain a desired shape. This represents the core definition of AM: any technology where the final geometry of the manufactured object is obtained through controlled addition of material qualifies as an Additive Manufacturing technique [1].

Advancements in the fields of computing, *Computer Aided Design* (CAD), and controllers, among other technological developments, were necessary to translate the patents into working prototypes, with some eventually becoming the foundations of commercially successful companies -such as 3D Systems in 1986 and Stratasys in 1989 [1, 17, 18]. The basic process of AM has remained largely unchanged from its first iteration in the late 80s: First, a computer model of the object is made using CAD software and exported under the *.stl* file format. Afterwards, the part geometry is stratified, or “sliced”, and translated into machine instructions using a specialized software called *slicing engine*. An AM machine then follows said instructions, commonly referred to as the *toolpath*, to build the object in layers. Finally, the part is available to the user. Depending on either the requirements of the part, or the specifics of the AM technique used, some post-processing may be required [1]. A visual representation of the process is shown in Figure 1.1.

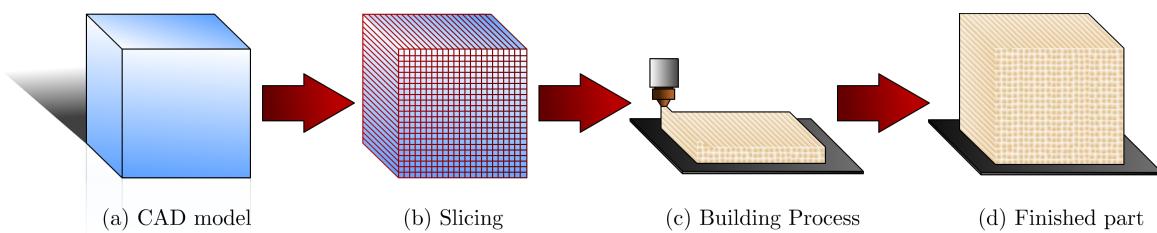


Figure 1.1: Process flow of AM

While all AM technologies operate on the same basic process flow described above, the specifics of each AM technique vary substantially, ranging from processes that use paper and binder, all the way through metal-based, laser tracing technologies. Since this is a rapidly evolving field, no general consensus exists for classifying the multiple AM processes available as of the time of this writing. However, the classification system proposed under the ASTM/ISO 52900 standard [19], has been somewhat accepted by the field and divides AM technologies as follows:

1. **Binder Jetting:** AM techniques where a binding agent is used to selectively promote cohesion in powder materials -generally gypsum, sand or metallic powders [19, 20].
2. **Directed Energy Deposition:** AM processes where a focused thermal energy source (i.e. laser, electron beam, plasma arc) is used to fuse materials as they are being deposited in the build volume. Materials are almost exclusively metals [19, 20].
3. **Material Extrusion:** In this type of AM technology, material is dispensed through a nozzle or orifice. Fused Filament Fabrication belongs to this classification. Materials are almost exclusively thermoplastics [19, 20].
4. **Material Jetting:** AM techniques where build material is deposited selectively in droplets. Materials are usually wax or thermoplastics, but there are examples of metal-based, material jetting techniques [19, 20].
5. **Powder Bed Fusion:** AM processes where portions of a powder bed are selectively fused through application of thermal energy. *Selective Laser Sintering* (SLS) belongs to this category. Materials are usually thermoplastic polymers or metals [19, 20].
6. **Sheet Lamination:** In this type of AM technology, the final part is formed by bonding sheets of material -usually paper or composites [19, 20].
7. **Vat Photopolymerization:** In this AM process, a liquid photopolymer is selectively cured by a light source. *Stereolithography* (SLA), arguably the first AM technology, belongs to this category. Due to the nature of this technique, the only materials used are photopolymers [19, 20].

1.1.1 Advantages and Disadvantages of AM

Since AM processes allow a relatively direct conversion of a CAD model into a constructed object, they were originally exclusively used for prototype development. For this reason, they were initially classified as “*Rapid Prototyping*” (RP) technologies. This terminology is still used today, however, it is being superseded by *Additive Manufacturing* since its potential to become a proper fabrication technique exists [1]. While

being capable of quickly jumping from part design to manufacturing is a great advantage, AM has its own set of drawbacks. Table 1.1 summarizes the most noteworthy set of advantages and disadvantages typical of most AM technologies.

Table 1.1: Advantages and Disadvantages of Additive Manufacturing

Advantages	Disadvantages
Faster product development cycles [1]	Part quality highly dependent on process parameters [1]
No additional tools needed for part fabrication[1]	Stratified build generally results in anisotropic parts [1, 2]
Cost effective for small batches of parts [21, 22, 23]	Costly for production of more than hundreds of parts [21, 22, 23]

Out of all advantages and disadvantages described, the high anisotropy of AM parts is responsible for the slow embrace of AM in highly demanding engineering fields -such as the aerospace and automotive industries. The highly anisotropic mechanical behavior makes it extremely difficult to predict part failure, therefore, it cannot be implemented in engineering applications where catastrophic failure is to be avoided at all costs. Even so, success stories of implementation of AM in industrial environments are abundant. Relatively recent examples include the use of FFF machines to manufacture tools, jigs, and fixtures in a Volkswagen assembly plant in Europe [5]; production of a complex fuel nozzle injector for the LEAP jet engine, using powder based, metal AM by GE [24]; and development and production of highly optimized, 3D printed midsoles for high performance running sneakers by companies as large as New Balance and Adidas [25, 26, 27]. Note that in the cases presented, the main reason behind the usage of AM was either reduction of expenses associated with producing small batches of parts, or the capability of reproducing a unique and complex geometry. This is a trend that is observed in most of the literature describing implementation of AM into industrial scenarios.

While the advantages and disadvantages described here cover the field of AM as a whole, each technique comes with its own set of pros and cons that may make it the preferred method to reproduce a particular product or geometry. This work, however, focuses solely on FFF. The specifics of this process are described in detail in Section 1.2.

1.2 Fused Filament Fabrication

Fused Filament Fabrication (FFF) is an AM technology where the final geometry of the part is obtained through controlled extrusion of a liquid, self-hardening material -usually a thermoplastic polymer in molten state [1]. Originally developed by Stratasys in the 1980s under the still trademarked *Fused Deposition Modeling* (FDM™) moniker, it has recently become one of the most widely used AM techniques due to the advent

of low-cost, desktop FFF machines in the early 2010s caused by the expiration of key patents from Stratasys [1, 2].

1.2.1 The FFF process

At its core, the typical FFF machine consists of a heated build surface commonly referred to as a *build plate*, a specialized tool known as a *printhead*, and the fabrication material -supplied in the form of spools of thermoplastic polymer filament. The printhead is itself composed of a heating element, a nozzle, and some form of driving mechanism that pushes the filament downward. As the thermoplastic material is moved through the heated chamber, polymer melt is formed and extruded through the opening at the tip of the nozzle, producing a *bead*. The molten polymer can then be deposited upon the build plate, where controlled movements of the printhead and the fabrication surface gradually construct the final geometry of the part in a layer-by-layer build approach [1]. The typical setup of an FFF machine can be seen in Figure 1.2. In this example, the printhead moves in the *x-y* plane, while the build plate moves in the *z* direction.

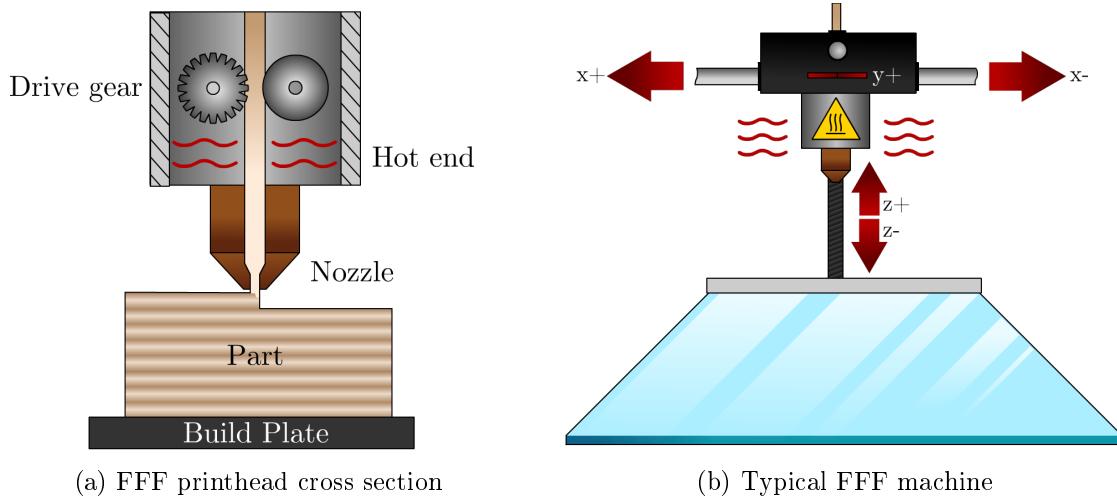


Figure 1.2: The basic FFF machine configuration

Like all AM technologies, the FFF process starts in a computer with a CAD model converted to the *.stl* file format. The geometry is then translated to machine instructions through a *slicing engine*, where the user inputs a plethora of process parameters that include nozzle and build plate temperatures, print speed, layer thickness, and build orientation. Finally the *toolpath* is executed by the FFF printer, building the object in a layer-by-layer basis – sometimes referred to as *2.5D* printing [1, 4]. Figure 1.3 shows an abridged version of the process. The *z* axis indicates the intended build direction. Note how some of the finer details in the original CAD file are lost in the printed part – due in part to the layer height and build orientation selected.

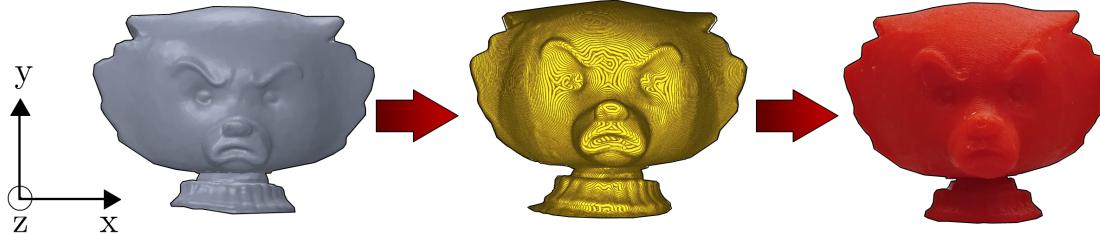


Figure 1.3: Model, toolpath and final part in the FFF process

The process is capable of producing complex geometries that would be otherwise hard to reproduce through other polymer processing techniques, such as injection molding. However, it is bound by the disadvantages described in Section 1.1.1, as well its own unique set of drawbacks. Namely:

- The circular orifice in the nozzle makes FFF incapable of reproducing sharp corners, limits the size of the smallest reproducible feature, and causes the final part to be filled with voids –originating in the junction of round beads. These problems can be seen in Figure 1.4: On the left, a comparison of a 90° corner planned in the toolpath and the final geometry of the printed bead is shown. Note the rounded nature of the turn. On the right, a cross section of an FFF part obtained through *Micro Computer Tomography* (μ CT) shows the voids that form during the printing process.

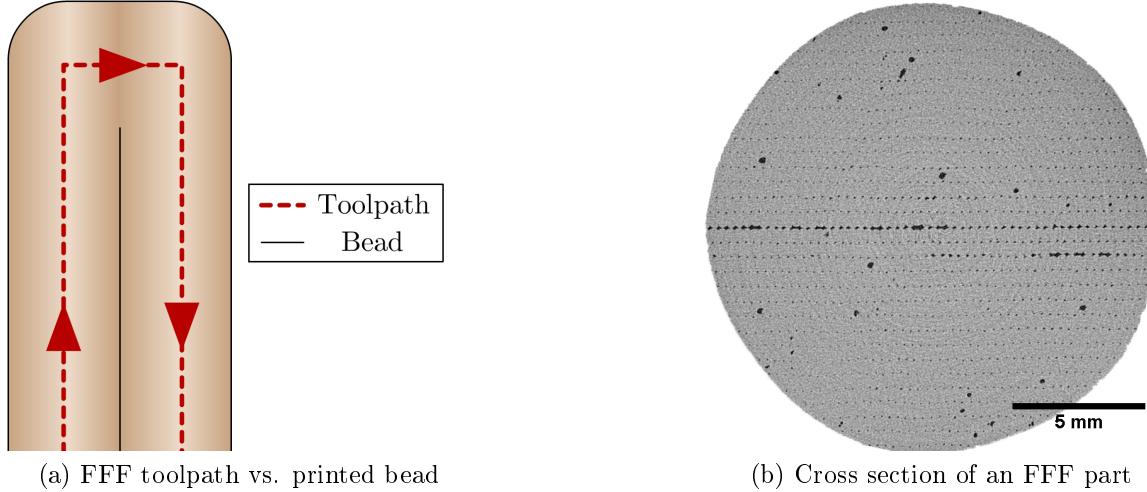


Figure 1.4: Typical FFF part mesostructure and its origin

- The junction of adjacent beads behaves akin to a polymeric weld, and has inferior mechanical properties than the bulk material [2]. This, coupled with the aforementioned voids which can act as stress concentrators, causes FFF parts

to behave in extremely anisotropic manner with diminished mechanical performance when compared to analogous parts obtained through traditional polymer processing technologies – such as injection molding [2].

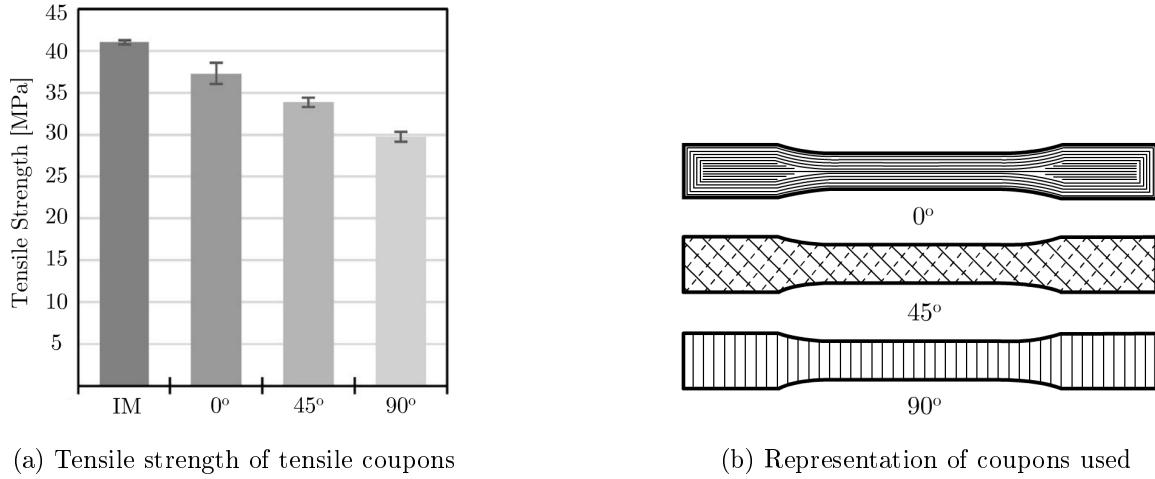
This last disadvantage is responsible for the slow embrace of FFF as a proper manufacturing technique: the high anisotropy of FFF parts imply that predicting part failure becomes extremely difficult and thus, proper part design that guarantees safe operation of the object under important loads is hard to achieve. For this reason, efforts to characterize the mechanical behavior of FFF parts have existed since as early as the 1990s. Recent examples are presented in Section 1.2.2.

1.2.2 Mechanical Properties of FFF parts

Efforts have been made to characterize the mechanical anisotropy of FFF parts. However, due to the lack of testing standards and problems during toolpath planning, most studies focus solely in the tensile mechanical performance of FFF coupons.

Studies performed by Koch *et al.* [6] and Rankouhi *et al.* [7] indicate that the final tensile properties of FFF coupons are particularly sensitive to bead orientation and proper mass output through the nozzle. Other process parameters, such as the layer thickness, have varying degrees of impact upon the final tensile strength of the part. In both studies, tensile coupons were printed with bead orientations of 0° , 45° and 90° in the x - y plane. Results showed that in all the experimental conditions selected, a 0° orientation always behaved closer to the bulk material, whereas a 90° sample always had significantly lower tensile strengths. The 45° samples sat between both extremes. It is important to note that in both studies, toolpath manipulation was necessary to avoid premature failure of the coupons due to stress concentrators originating in void formation due to the elliptical nature of the beads. Figure 1.5 shows some of the results by Koch *et al.* The geometry corresponds to an ASTM Type I Tensile coupon. Injection molded results are denoted *IM* for comparison. Note that the 90° orientation had a tensile strength that was 25% inferior to the IM counterpart, and 20% worse than the 0° oriented FFF coupon. This is a prevalent trend in the consulted bibliography.

Literature for other types of mechanical testing of FFF parts is relatively scarce when compared to tension experiments. Research indicates that the compressive strength of FFF parts tends to be higher than the tensile strength, as well as being less sensitive to process parameters —the bead orientation in particular seems to have a significantly diminished impact upon the compressive strength when compared to its effect upon tensile tests [28, 29]. Shear strength results are virtually non-existent.

Figure 1.5: Results from Koch *et al.* [6]

1.3 Failure Criteria

The increased use of advanced materials in industry has brought upon a necessity to properly characterize their strengths and failure modes. Composites in particular are commonly used in highly demanding engineering fields given that they excel in mechanical properties. However, due to their nature, their behavior is extremely anisotropic. For this reason, it has been of great interest to develop a proper way to model the behavior of anisotropic materials under mechanical stresses as a way to predict part failure – a practice from here on referred to as developing a *failure criterion*.

Early attempts to properly predict failure of anisotropic materials go as far back as 1948 with the Hill model [11]. Further developments led to a plethora of Failure Criteria (FC), such as the Tsai-Hill, Malmeister, Tsai-Wu, Gol'denblat-Kopnov, Puck, and Cuntze to name a few [11, 30]. A wide variety of criteria exists because a model will rarely capture the complete failure behavior of an anisotropic material. To illustrate this point, refer to Figure 1.6, reproduced from work by Sun *et al.* [31] where a composite glass fiber and epoxy laminate was loaded biaxially, in a direction that was either parallel (σ_{11}), perpendicular (σ_{22}) to the fiber, or a combination of both. Positive stresses indicate tensile load, while negative values point to compressive forces. The data, represented by the white squares, does not agree with any of the used models in the fourth quadrant of the graph. This type of behavior is common throughout the literature: Puck's model is great at predicting shear strengthening effects, but doesn't perform well when dealing with combined axial loading scenarios; the Gol'denblat-Kopnov model by contrast is great at predicting axial stress interactions, but falls short when dealing with shear strengthening effects caused by combined shear-axial loadings. These trends point to the limitations of each model: in order to either facilitate calculations, or due to the difficulty of performing combined loading tests,

interaction effects are neglected either by mathematical choice, or indirectly through the inner workings of the failure criterion [11].

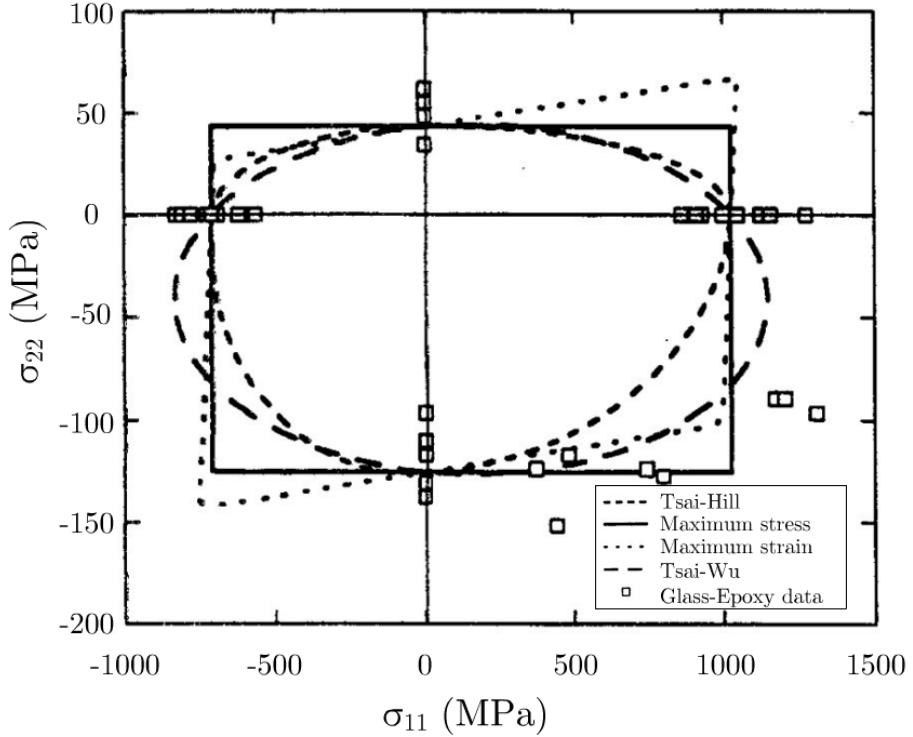


Figure 1.6: Comparison of different failure criteria. [31]

1.3.1 The Stress-Stress Interaction Criterion

The majority of FC fail to completely integrate interaction effects into the modeled failure behavior of anisotropic materials. Work published in 2017 by Paul and Tim Osswald [11] proposed a model that attempts to overcome these limitations by developing a failure function based on the approach described by Gol'denblat and Kopnov in 1965 [32]. The model proposed by Osswald and Osswald, originally titled “A Strength Tensor Based Failure Criterion with Stress Interactions”, will be referred in this work as the Stress-Stress Interaction Criterion (SSIC), and has the following characteristics:

- **Tensor based and purely mathematical:** as opposed to phenomenological or mechanistic models such as the Puck or Cuntze failure criteria.
- **Includes stress interactions that other models neglect.**

To understand the SSIC, it is necessary to describe the model upon which it is based. The Gol'denblat-Kopnov Criterion (GKC) describes a mathematical function

that depends on the stress state of an anisotropic material. Should the computation of this expression exceed a threshold, part failure is to be expected. To that end, a scalar function that depends on stress tensors that completely characterize the state of the material was developed [32]. This function is shown in Equation 1.1, where stresses are denoted σ , and the subindices i,j,k,l denote a particular load direction.

$$f = (F_{ij}\sigma_{ij})^\alpha + (F_{ijkl}\sigma_{ij}\sigma_{kl})^\beta + (F_{ijklmn}\sigma_{ij}\sigma_{kl}\sigma_{mn})^\gamma + \dots \quad (1.1)$$

The terms F_{ij} , F_{ijkl} and F_{ijklmn} represent second, fourth and sixth order tensors respectively. These terms of the equation depend on engineering strength parameters, such as the ultimate tensile and compressive strengths of the material in a particular load direction [11]. Due to the complexity associated with using higher order tensors, Gol'denblat and Kopnov limited their approach to using only the second and fourth order terms. Thus Equation 1.1 is reduced to:

$$f = (F_{ij}\sigma_{ij})^\alpha + (F_{ijkl}\sigma_{ij}\sigma_{kl})^\beta \quad (1.2)$$

In order to attain a linear criterion scalar function, the exponents α and β were assigned values of 1 and $1/2$ respectively. Finally, in plane stress scenarios, the GKC becomes:

$$\begin{aligned} f = F_{11}\sigma_{11} + F_{22}\sigma_{22} + F_{12}\tau_{12} + & (F_{1111}\sigma_{11}^2 + F_{2222}\sigma_{22}^2 + F_{1212}\tau_{12}^2 \\ & + 2F_{1122}\sigma_{11}\sigma_{22} + 2F_{1112}\sigma_{11}\tau_{12} + 2F_{2212}\sigma_{22}\tau_{12})^{1/2} \end{aligned} \quad (1.3)$$

Note that in Equation 1.3 σ and τ denote normal and shear stresses respectively. Figure 1.7 depicts an anisotropic material and all the possible loading directions for reference.

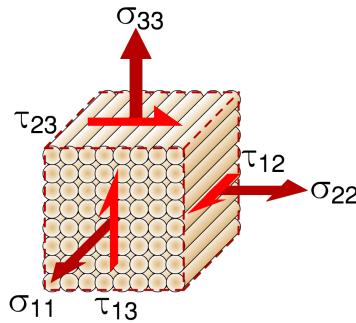


Figure 1.7: Different load directions in an anisotropic material

Per Gol'denblat and Kopnov's design, should the computation of f in Equation 1.3 be greater or equal to 1, part failure is to be expected. However, to simplify calculations, they deliberately assumed the interaction terms F_{1112} and F_{2212} to be

zero. This is an important consideration that will come into play when describing the SSIC.

Most of the terms in the GKC are obtained through mechanical testing of coupons under pure uniaxial loads in the 1 or 2 direction, or pure shear in the 1-2 plane [11]. In these scenarios, f will be equal to 1 at failure, and the stress state will be known to the user, allowing some of the unknown tensorial parameters to be easily calculated. Using F_{11} and F_{1111} as examples, the process would be as follows:

1. The tensile and compressive strength in the 1-1 direction would be obtained through mechanical testing. These values are named X_t and X_c respectively.
2. Under these failure conditions, Equation 1.3 is reduced to the following system of equations:

$$\begin{cases} 1 = F_{11}X_t + (F_{1111}X_t^2)^{1/2} \\ 1 = -F_{11}X_c + (F_{1111}X_c^2)^{1/2} \end{cases}$$

3. F_{11} and F_{1111} can be obtained, yielding $F_{11} = \frac{1}{2}(\frac{1}{X_t} - \frac{1}{X_c})$ and $F_{1111} = \frac{1}{4}(\frac{1}{X_t} + \frac{1}{X_c})^2$.

The only exception to this procedure would be the F_{1122} component, which requires measuring the positive and negative shear strengths of a coupon with reinforcement oriented in 45° . These parameters are named S_{45p} and S_{45n} respectively. Table 1.2 summarizes the nomenclature used for the strength parameters required to completely populate the failure function of the GKC. Table 1.3 summarizes all the tensorial component calculations.

Table 1.2: Nomenclature of the GKC parameters

Parameter	Description
X_t	Tensile strength in the 1-1 direction
X_c	Compressive strength in the 1-1 direction
Y_t	Tensile strength in the 2-2 direction
Y_c	Compressive strength in the 2-2 direction
S_{45p}	Positive shear strength for 45° specimen
S_{45n}	Negative shear strength for 45° specimen
S	Shear strength in the 1-2 plane

One of the assumptions made in the GKC is that the components F_{1112} and F_{2212} in Equation 1.3 are null. While this simplifies the model, it essentially neglects any interactions between axial loads and shear stresses, namely, the $\sigma_{11} - \tau_{12}$ and $\sigma_{22} - \tau_{12}$ interactions. Practically, this causes the failure surface developed through the GKC to under-predict shear strengthening effects exhibited by anisotropic materials loaded in combined axial and shear conditions. The Stress-Stress Interaction Criterion (SSIC) attempts to overcome these limitations by building upon the GKC. For the SSIC, the

Table 1.3: Tensorial components of the GKC

Component	Formula
F_{11}	$\frac{1}{2}(\frac{1}{X_t} - \frac{1}{X_c})$
F_{1111}	$\frac{1}{4}(\frac{1}{X_t} + \frac{1}{X_c})^2$
F_{22}	$\frac{1}{2}(\frac{1}{Y_t} - \frac{1}{Y_c})$
F_{2222}	$\frac{1}{4}(\frac{1}{Y_t} + \frac{1}{Y_c})^2$
F_{12}	0
F_{1212}	$\frac{1}{S^2}$
F_{1122}	$\frac{1}{8}[(\frac{1}{X_t} + \frac{1}{X_c})^2 + (\frac{1}{Y_t} + \frac{1}{Y_c})^2 - (\frac{1}{S_{45p}} + \frac{1}{S_{45n}})^2]$

interaction effects are captured through the use of the slopes of the failure surface at any of the points where the engineering strength is known within a particular stress plane [11]. In this failure scenario, the stress state of the coupon is known and easy to implement into Equation 1.3, where $f = 1$. The resulting expression can then be derived with respect to one of the stresses, allowing for the interaction components to be calculated. This is better illustrated through an example. Assuming the component of interest is F_{2212} , the procedure to calculate it through the SSIC would be as follows:

1. Obtain all the tensorial components possible through the GKC.
2. Using the $\sigma_{22}-\tau_{12}$ stress plane, take the derivative of Equation 1.3 as a function of σ_{22} in the scenario of failure under pure shear ($f = 1$). This yields the expression:

$$0 = F_{22} + [F_{1212}S(\frac{d\tau_{12}}{d\sigma_{22}}) + F_{2212}S] \quad (1.4)$$

where $\frac{d\tau_{12}}{d\sigma_{22}}$ is the slope of the graph at failure under shear. This term is named μ^{2212} in the SSIC and can be obtained by performing combined loading tests.

3. Rearranging Equation 1.4 to solve for the unknown F_{2212} gives the following expression:

$$F_{2212} = -\frac{F_{22}}{S} - F_{1212}\mu^{2212} \quad (1.5)$$

A similar procedure can be followed for any $\sigma_{ii}-\tau_{ij}$ interaction, or even any $\sigma_{ii}-\sigma_{jj}$ components. For this last scenario, the user has four potential choices of slopes to determine the tensorial component of interest. In the SSIC, any slope obtained from a $\sigma_{ii}-\sigma_{jj}$ stress plane is named λ^{iijj} , as opposed to μ^{iiji} for slopes in a $\sigma_{ii}-\tau_{ij}$ reference. A schematic of all possible interaction slopes is shown in Figure 1.8, while Table 1.4 summarizes all the possible interaction factors available through the SSIC, where τ_{ij}^u denotes ultimate shear strength in a particular shear plane.

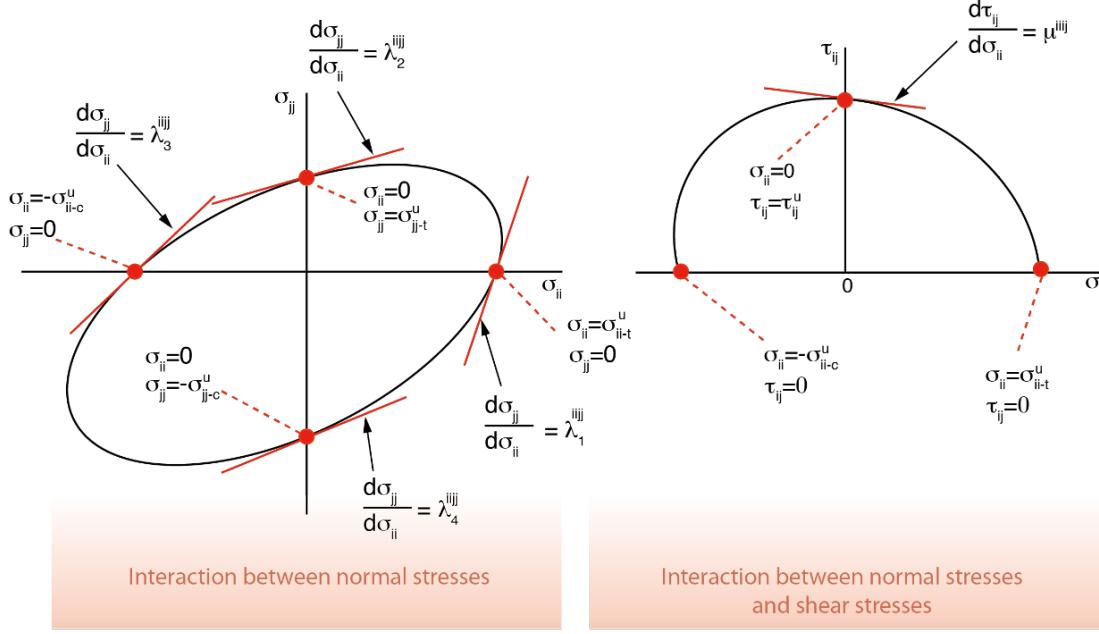


Figure 1.8: Interaction slopes available through the SSIC

Table 1.4: Interaction components attainable through the SSIC [11]

Component	Formula
F_{iiji}	$-\frac{F_{ii}}{\tau_{ij}^u} - F_{ijij}\mu^{iiji}$
F_{iiji} through λ_1^{ijj}	$-\frac{(F_{ii} + F_{jj}\lambda_1^{ijj})F_{iiji}^{1/2} + F_{iiji}}{\lambda_1^{ijj}}$
F_{iiji} through λ_2^{ijj}	$-(F_{ii} + F_{jj}\lambda_2^{ijj})F_{iiji}^{1/2} - F_{iiji}\lambda_2^{ijj}$
F_{iiji} through λ_3^{ijj}	$\frac{(F_{ii} + F_{jj}\lambda_3^{ijj})F_{iiji}^{1/2} - F_{iiji}}{\lambda_3^{ijj}}$
F_{iiji} through λ_4^{ijj}	$(F_{ii} + F_{jj}\lambda_4^{ijj})F_{iiji}^{1/2} - F_{iiji}\lambda_4^{ijj}$

The SSIC offers a way of capturing in a more accurate manner the different failure modes of parts produced through AM technologies. As an example, the model has been successfully implemented by Obst *et al.* in 2018 for SLS manufactured parts produced with PA12 [8, 33]. Their results show how the model was able to capture the τ_{12} - σ_{22} and σ_{11} - σ_{22} interactions. The failure surface obtained, shown in Figure 1.9, was able to capture the interactions between certain axial and transverse stresses. However, due to the limitations of the SLS process, it was not possible to measure the interaction slope between the τ_{12} and σ_{11} directions.

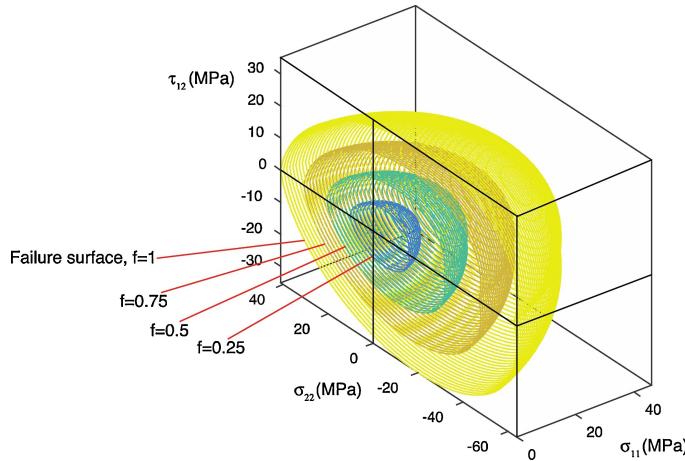


Figure 1.9: Failure surface for SLS developed through the SSIC [8]

Recent work by Osswald *et al.* [34] generated a failure envelope for Multi-Jet Fusion (MJF) parts produced using PA12, and compared it to the surface obtained by Obst *et al.* [8]. Results indicate that, while both techniques are based on Powder Based Fusion (PBF) and use the same material, the envelopes for each AM technology were distinct, serving as proof that these technologies are not as comparable under complex loading conditions as previously assumed. The transverse-axial interaction for the MJF case was significantly less pronounced than for SLS, further reinforcing that each AM technology needs to be studied in a case-by-case basis in terms of mechanical failure characterization.

1.4 Development of SSIC envelope for FFF parts

In 2019, Mazzei Capote *et al.* [10] developed a failure envelope for FFF parts produced using a customized ABS filament produced in-house. Specimens were produced using either a commercially available desktop FFF printer (Lulzbot TAZ5, USA), or a customized 6-axis robotic printing solution whenever the bead orientation was hard to achieve using a 2.5-D machine. The robotic printer was based on a 6-axis robot (ABB IRB-120, Switzerland) and fitted with a stationary printhead mounted on an aluminum frame, chosen to be the same extruder from the traditional printer (LulzBot TAZ Single Extruder Tool Head v2, 0.5 mm nozzle, USA) to minimize machine influence on the results [4]. The final surface obtained showed significant stress interactions in certain directions. Starting with the σ_{11} - σ_{22} plane, it can be seen that the failure envelope has a slight tilt. Refer to Figure 1.11 for a graph showing the calculated failure envelope, including the experimental data for reference. This tilt is evidence of an interaction between the transverse and longitudinal stresses. The conclusion is that FFF parts produced with the print parameters used in the study should show strengthening when loaded bi-axially in compression.

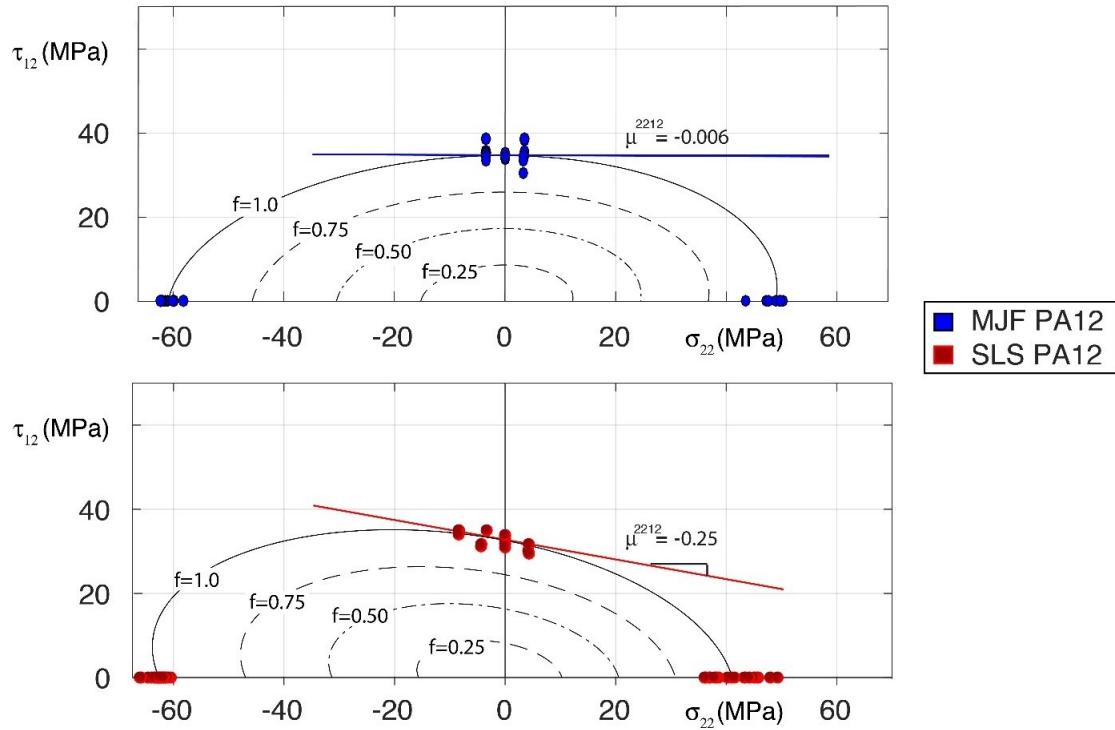


Figure 1.10: Comparison of the $\sigma_{22} - \tau_{12}$ interaction for SLS and MJF PA12 parts [34]

Using the results from combined loading tests plotted in the 11 – 12 and 22 – 12 planes allows visualization of the transverse-axial stress interactions. Beginning with the 11 – 12 plane, it can be seen that the calculated interaction slope μ^{1112} equals 5.2×10^{-3} , a value that's practically zero. Using this parameter, the failure surface shown in Figure 1.12a can be obtained. A dashed line representing μ^{1112} is added for reference. The 22 – 12 plane by comparison reveals a considerable slope. It can be seen through the use of combined loads that there is a slight decrease in the shear strength of the specimens when a tensile load is applied in the 2 – 2 direction. A slope of -0.2 was obtained for μ^{2212} . Figure 1.12b shows the resulting surface with the data and a line with a slope of -0.2 overlaid for reference.

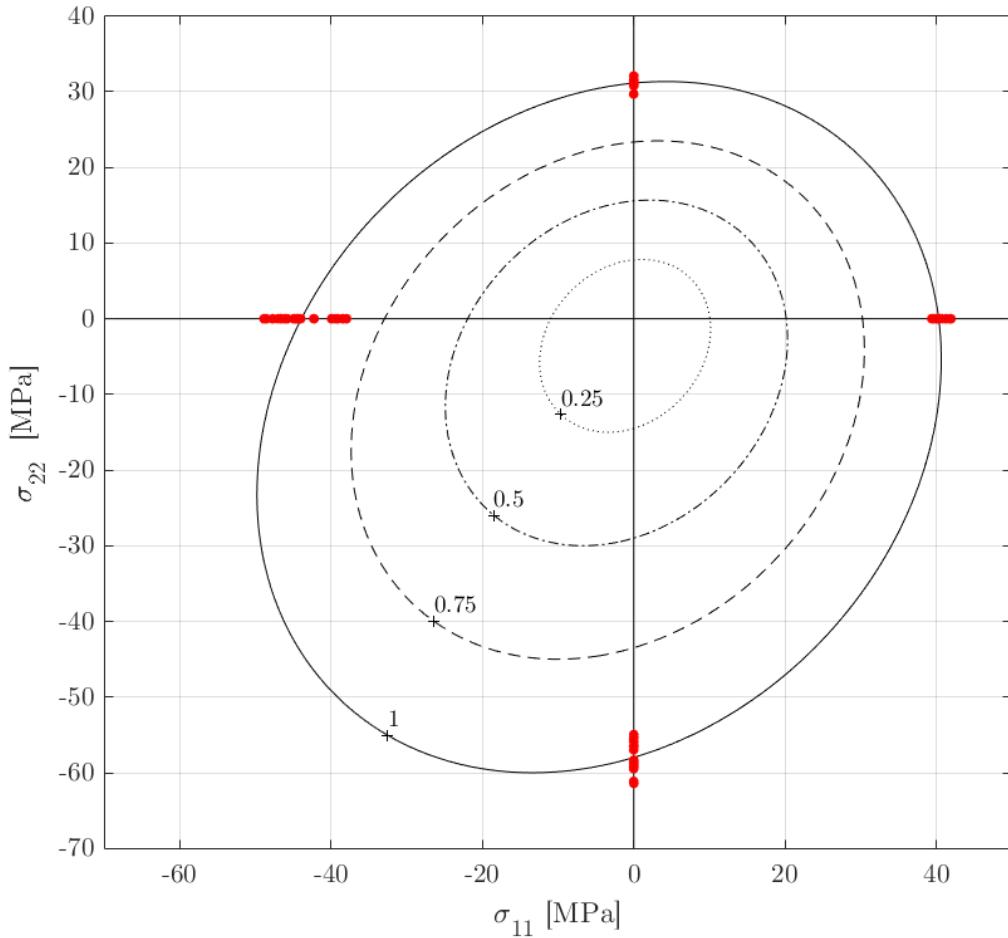


Figure 1.11: σ_{11} - σ_{22} plane including data for reference.

The use of AM technologies to produce small batches of highly customized, complex parts, in a reduced development cycle results extremely attractive. While constructing failure envelopes can help overcome the wariness of industrial segments to design end-user parts, this resource is still not easy to implement, requiring a large number of mechanical tests and specialized equipment to properly map the failure behavior of a particular material. Additional complexity stems from what was shown in Subsection 1.3.1: processing the same material under related AM technologies yields completely different failure envelopes, implying that no generalizations should be made, and each material-process pairing needs to be studied on a case-by-case basis. In general, for AM parts to be adopted, engineers have to be able to confidently assess the probability of part failure under particular loading conditions, predict the expected mechanical properties of AM parts, and understand the underlying physics of the process. None of these conditions are completely met at the time of this work. Section 1.5 outlines

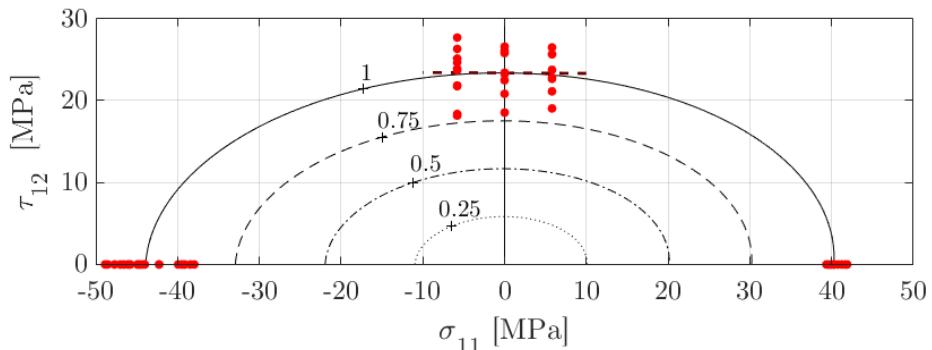
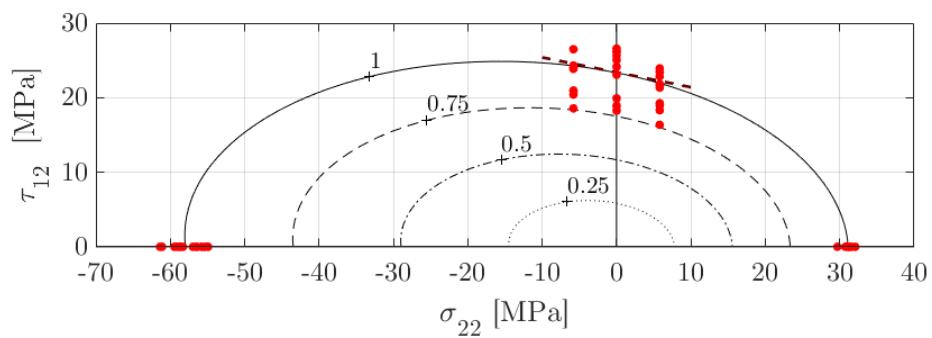
(a) failure envelope in the σ_{11} - τ_{12} plane(b) failure envelope in the σ_{22} - τ_{12} plane

Figure 1.12: Comparison of interaction slopes in the axial-transverse stress planes

machine learning methods, that can be used to predict the mechanical response of FFF parts based on process parameters, with some, if not all of the concepts explored being easily extrapolated to other AM techniques. It should be noted that the two methods are not mutually exclusive. A combination of both FC and machine learning predictive methods can potentially lead to higher adoption rates of AM in industrial scenarios where the final desired application involves complex mechanical loads upon the finished part.

1.5 Machine Learning Techniques

The relationship between printing conditions and final mechanical response of FFF parts is not completely understood, and the physics that govern the process appear to be complex. Multiple attempts have been made to model the physics of the melting process that occurs inside the nozzle, each with their own set of assumptions that do not seem to fully grasp the nature of the small scale extrusion occurring with the FFF technology [35]. While it is understood that processing parameters have an impact on the final dimensions of the extruded polymer beads, there is a clear disconnect between how it relates to the process physics [6]. However, this constitutes an interesting case for development of a Machine Learning (ML) solution, which excel in cases where the inputs and outcomes of a particular phenomena or task are known, but connecting the two through an explicit set of rules or relationships can result extremely complex and time consuming [12]. In this manner, ML models are *trained*, as opposed to explicitly defined, as illustrated in Figure 1.13, where the differences between ML and traditional programming philosophies are compared.

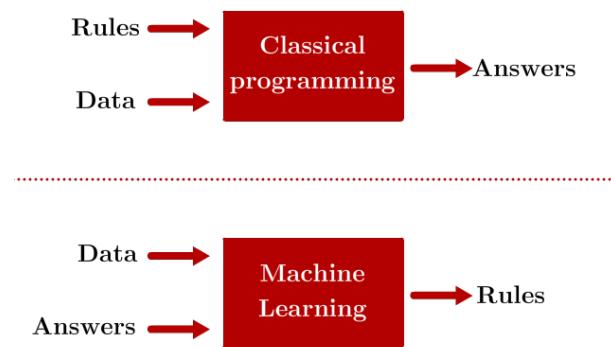


Figure 1.13: Differences between traditional programming and machine learning. [12]

The potential to apply ML solutions in the field of AM has been noted by several authors [14, 15]. Example cases include design-recommendation systems, topology optimization solutions, tolerancing and manufacturability assessment, and material

classification and selection [14]. Machine learning algorithms are varied, and a summary of how they have been recently applied to AM can be seen in Figure 1.14.

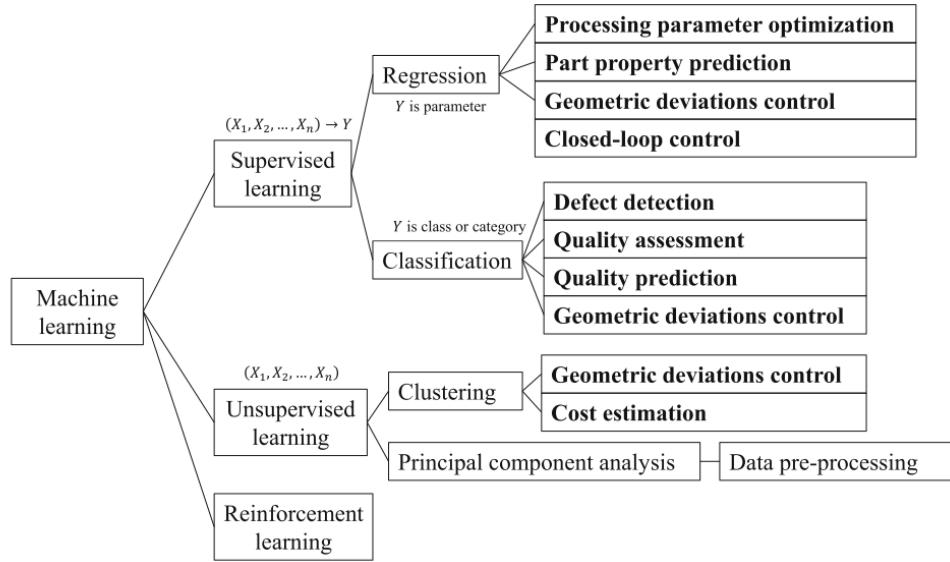


Figure 1.14: Taxonomy of ML applications in AM [15]

Given the factors outlined this far, the fundamental goal of this research is to predict FFF part mechanical performance by finding relations between processing conditions and strength through the use of sensors and machine learning. The success of this project would allow design engineers to confidently assess if a part manufactured through FFF will meet the mechanical requirements imposed by its intended application. This work proposes developing and using a modified printer with force and print speed sensors, as well as mechanical testing to generate data that can be used to train a predictive tool based on ML. This tool can then be used to predict final mechanical properties of the part based on the data generated during the print.

2 Part Structural Integrity Evaluation through Failure Criteria

2.1 Foreword

Fused Filament Fabrication (FFF) is arguably the most widely available Additive Manufacturing (AM) technology at the moment. Offering the possibility of producing complex geometries in a compressed product development cycle and in a plethora of materials, it has gradually started to become attractive to multiple industrial segments, slowly being implemented in diverse applications. However, the high anisotropy of parts developed through this technique renders failure prediction difficult. The proper performance of the part, or even the safety of the final user, can't be guaranteed under demanding mechanical requirements. This problem can be tackled through the development of a failure envelope that allows engineers to predict failure by using the knowledge of the stress state of the part. A previously developed failure envelope for ABS based, FFF parts by use of a criterion that incorporates stress interactions is used in this chapter to predict the structural integrity of FFF produced parts. In the context of this dissertation, the work that follows shows how one can use such envelope to predict mechanical part failure within 10% of the real measured value, and compares how the prediction that stems from the SSIC is more accurate than those derived from simpler but more ubiquitous FC.

2.2 Introduction

FFF's main advantages are its capabilities to produce complex geometries that would otherwise be difficult to achieve, and an extremely short part development cycle, which facilitates rapid design iterations. However, this technology still faces the challenges and limitations that currently affect the entire field of AM. Namely, the anisotropy introduced through the layer-by-layer build approach makes it difficult to assess the expected mechanical behavior of FFF parts when subjected to stresses, and thus, industrial applications are still limited in scope [1]. This is a problem that can be solved through application of a failure criterion (FC) to safely assess if the part is going to perform without failing when subjected to the expected mechanical requirements [30],

[11]. Literature on the topic in the field of AM is scarce, but successful attempts have been published for Selective Laser Sintering (SLS) by Obst et al. [8], Multi-jet Fusion by Osswald et al. [34], and previously by the author of this dissertation for FFF [36, 10]. The applied criterion in all cases was developed by Osswald and Osswald in 2017 [11], through improvements upon the method originally described by Gol'denblat and Kopnov in 1965 [32]. This failure criterion defines a scalar function f that depends on the stress state of the object, as well as strength tensors. Should the calculated value of f exceed 1, part failure is to be expected. For more information related to the mathematical description of this FC, the reader is invited to review Section 1.3 of this document.

In order to experimentally validate the results of the failure surface, combined loads can be generated in the principal directions and compared to those of the envelope at $f = 1$ using destructive testing. According to laminate theory, one can generate a biaxial stress state along the principal material axes by applying off-axis uniaxial stress [37], as shown in Figure 2.1.

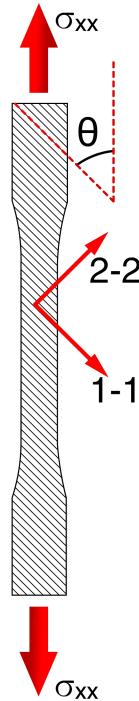


Figure 2.1: Off-axis loading conditions

Using this type of loading condition, the results can be converted to the principal coordinate system by using the following transformation:

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} \cos^2 \theta & \sin^2 \theta & 2 \cdot \cos \theta \cdot \sin \theta \\ \sin^2 \theta & \cos^2 \theta & -2 \cdot \cos \theta \cdot \sin \theta \\ -\cos \theta \cdot \sin \theta & \cos \theta \cdot \sin \theta & \cos^2 \theta - \sin^2 \theta \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ 0 \\ 0 \end{Bmatrix} \quad (2.1)$$

This study proposes verification of the previously developed envelope through this type of uniaxial test. Multiple FFF coupons produced with a variety of raster angles are used to compare the resulting complex stress states in the local coordinate system of the part, to the failure line determined through application of the failure criterion.

2.3 Experimental Methods

The toolpath of tensile coupons was generated using the SciSlice engine [38], following the ASTM D-638 Type I standard geometry [39] due to a lack of a standardized test for AM parts. In order to test a variety of combined loading scenarios, six raster configurations were selected: 0°, 30°, 45°, 60°, 75°, and 90° with respect to the loading direction, as depicted in Figure 2.1. Each orientation was replicated five times. The printing conditions were exactly the same as those used by the authors to generate the original failure envelope. These are shown in Table 2.1. Specimens were printed in a traditional desktop 3D printer (Lulzbot TAZ5, USA), using a customized 2.85 mm ABS filament extruded in-house, based on the Cycolac MG94 material produced by SABIC.

Table 2.1: Printing parameters maintained constant.

Printing Parameter	Value
Nozzle Temperature	220 °C
Bed Temperature	100 °C
Printing Speed	2000 mm/min
Layer Height	0.2 mm
Path Width	0.4 mm
Extrusion Factor	1

Mechanical tests were conducted on an Instron 5967 dual column universal testing machine with a 30 kN load cell, using the recommended testing speed of 5 mm/min, dictated by the ASTM standard [39]. All of the data acquisition was handled through the accompanying Instron Bluehill 3 software. The resulting experimental data was compared to the original failure envelope developed by the authors in previous work. To better visualize the results, the original mathematical formula, expressed in terms of stresses in the local coordinate system of the polymer beads, was translated into the global coordinate system. The transformation involves using the relation shown in Equation 2.1, resulting in the following system of equations. Solving for σ_{xx} allows

the failure surface to be expressed as a function of the raster angle and the tensorial components.

$$\sigma_{11} = \sigma_{xx} \cdot \cos(\theta)^2; \quad \sigma_{22} = \sigma_{xx} \cdot \sin(\theta)^2; \quad \tau_{12} = -\sigma_{xx} \cdot \cos \theta \cdot \sin \theta \quad (2.2)$$

$$1 = F_{11}\sigma_{11} + F_{22}\sigma_{22} + F_{12}\tau_{12} \\ + (F_{1111}\sigma_{11}^2 + F_{2222}\sigma_{22}^2 + F_{1212}\tau_{12}^2 + 2F_{1122}\sigma_{11}\sigma_{22} + 2F_{1112}\sigma_{11}\tau_{12} + 2F_{2212}\sigma_{22}\tau_{12})^{1/2} \quad (2.3)$$

2.4 Results

The maximum stress registered during testing gradually decreased as a function of the angle of the beads, with the highest value belonging to the coupons with a raster orientation parallel to the load direction, and the lowest being the specimens with the beads oriented perpendicular to the direction of the tensile stress. This is in accordance to previous work on the subject, investigating the strength of FFF parts as a function of the raster orientation. The tensile strength in the 0° orientation was approximately 20% higher than that observed for coupons produced with a 90° raster. Results are summarized in Figure 2.2. Error bars represent one standard deviation. The code used to calculate the FC lines and plotting the data can be found in Appendices B and C.

Plotting the failure line determined through the SSIC delimits a safe-unsafe threshold: any stress state above the line will likely result in part failure. For additional comparison, lines determined using the maximum stress criterion are shown, using the X_t , Y_t and S values determined for the SSIC. These are labeled $M1 - 1$, $M2 - 2$ and $M1 - 2$ respectively. Finally, overlaying the experimental data results in Figure 2.3.

It can be seen from the graph that no data point trespasses into the safety zone defined by the SSIC. The maximum deviation between the SSIC line and the experimental data amounts to approximately 8.6%, and this difference could be at least partially attributed to the variability intrinsic to reproducing the same geometry in a process like FFF. By comparison, simply using the maximum stress criterion proves inadequate, given that all the experimental tests fall well into the safe area determined by this criterion. These results give credence to the previously developed failure envelope.

2.5 Conclusions

The field of Additive Manufacturing as a whole can only achieve its maximum potential once standardization is in place, and users can safely assess if the designed part will perform as expected when subjected to important stresses. This body of work shows that the application of a failure criterion for FFF parts produced under a fixed set of print parameters is possible, and that the resulting envelope adequately predicts the overall

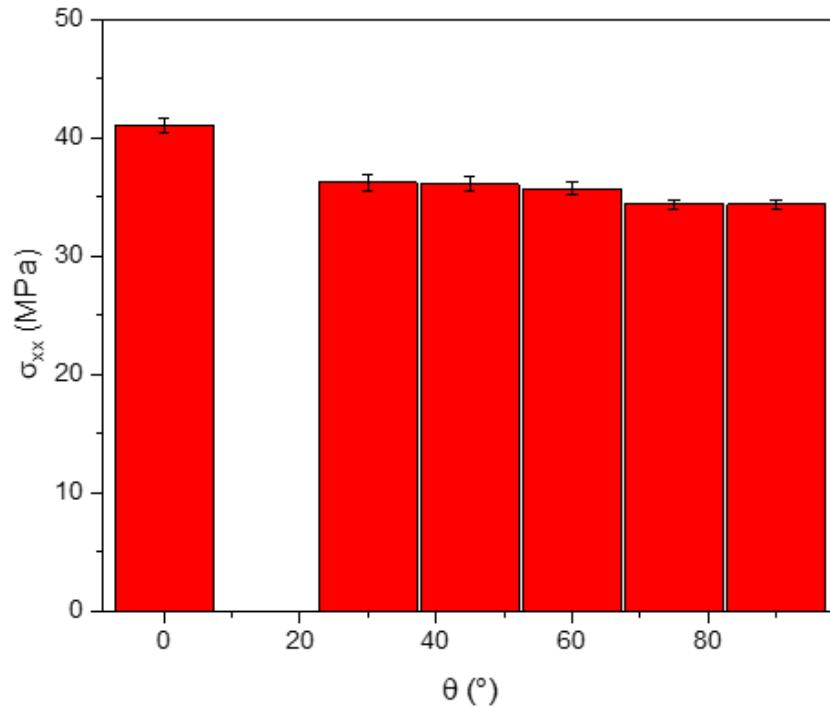


Figure 2.2: Experimental results from tensile tests

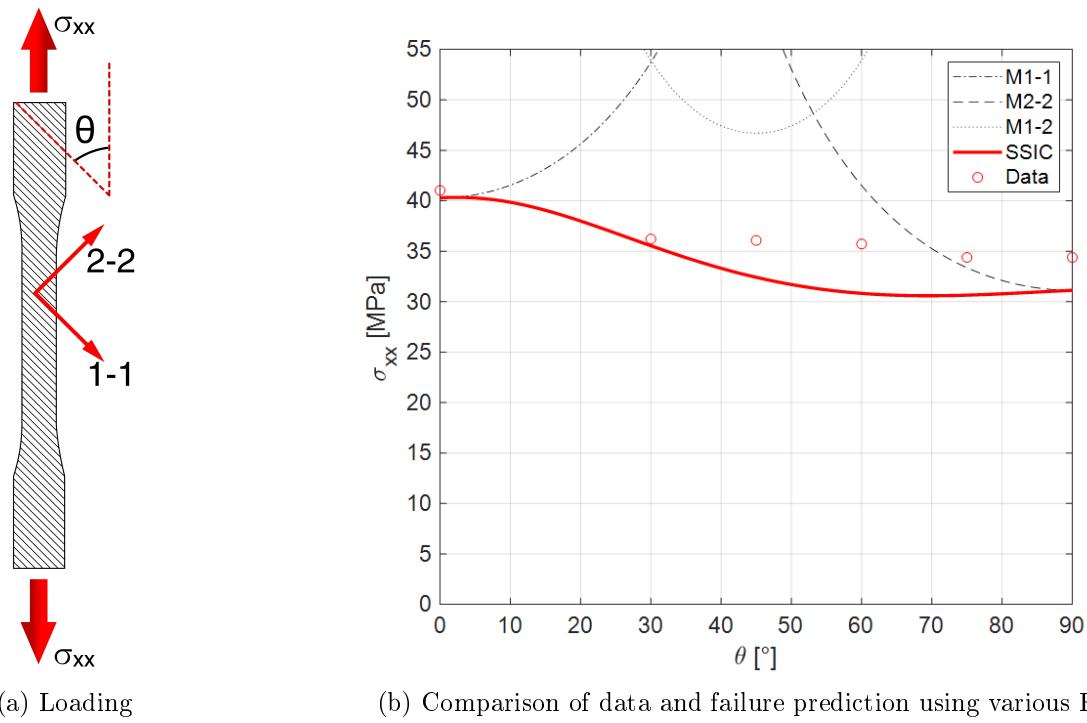


Figure 2.3: Results from [40]

trend of the failure behavior of parts produced by this technique. Results for uniaxial mechanical tests that resulted in complex stress states in the principal directions showed that the threshold determined by the SSIC applied for FFF was appropriate. Future work can expand on the concepts presented in this work by attempting to create a failure envelope for an FFF material that has more drastic differences in tensile strengths in the 1-1 and 2-2 directions, as well as approaching the failure estimation from a probabilistic stand.

3 Trends in Print Force and Extrusion Speed

3.1 Foreword

Material Extrusion (ME), also known as Fused Filament Fabrication (FFF), is the most prevalent Additive Manufacturing (AM) technology due to the broad range of materials and equipment available, generally attainable at a fraction of the cost of other AM processes. While ME has gradually found niche applications outside of the casual user, including implementations in the automotive and aerospace sectors, the slow print speed of the process represents a major pain point of the technology that limits the scope of its adoption. Additionally, process control relies mostly on temperature readings, and there is room for improvement in terms of variations in volumetric throughput and process fluctuations. For this reason, this study uses a customized ME machine with a built-in force sensor and encoder that allows real time acquisition of force and filament print speed data, generating knowledge aimed at maximizing the print speed of ME and improving the understanding of the underlying process physics that can lead to improved print quality and consistency.

In the context of this dissertation, every ML solution begins with a data acquisition and exploration step. This study represents just that, as it allowed the researcher to understand an FFF machine with sensors, develop data filtering techniques, and the deployment of automation protocols that made acquisition of the training and validation data sets of the ML model a lot easier and faster. Finally, and as an added bonus, the results of this study shed some light into the current limitations intrinsic to two of the most widely used melting models for FFF: the Bellini *et al.* model, and the Osswald, Puentes, Kattinger model, as both require measurements of filament force and speed to compare data with theoretical predictions —a feat that was uncommon at the time of this body of work.

3.2 Introduction

One of the major pain points from the Material Extrusion (ME) process is its relatively slow fabrication speed: while compelling for small batches of parts, ME can't come close

to competing with other polymer processing techniques, such as injection molding, which can reproduce a complex part in a matter of seconds, as opposed to hours [21, 22, 23]. The cause of the sluggish nature of ME is rooted in heat transfer and the underlying physics of the process: as the thermoplastic filament is pushed through a heated chamber, polymer melt is formed and extruded through an opening at the tip of the nozzle, producing a bead of material that will be used to gradually reproduce the geometry of the desired part. The problem lies partially on the melt formation within the nozzle: its physics and limitations are not completely understood at the time of this writing, making print-speed optimization difficult and a matter of trial and error. Attempts to curb this issue through mathematical modeling of the melt formation exist, analytically derived through transport phenomena and rheology. Two major attempts exist, differing in the key assumptions made regarding how the molten polymer is formed and extruded. The first, proposed by Bellini, Güceri and Bertoldi in 2004 [41], and the second proposed by Osswald, Puentes and Kattinger in 2018 [42] and improved upon by Colón et al. in 2020 [35]. A schematic representation of both models can be seen in Figure 3.1.

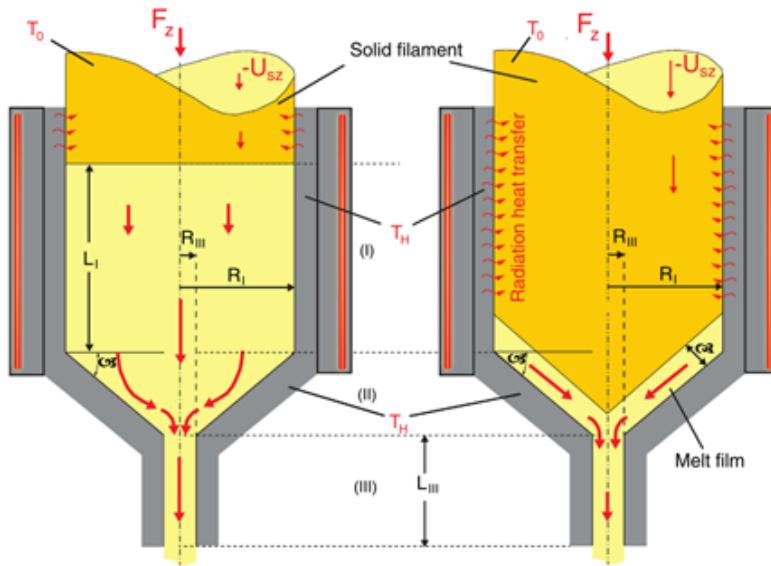


Figure 3.1: Bellini model (left) and Osswald, Puentes, Kattinger model (right) [43].

The Bellini, Güceri and Bertoldi model assumes that a melt pool forms as soon as the filament enters the heated chamber, with the solid filament acting as a piston that allows the flow of the molten polymer [41]. Using the nomenclature shown in Figure 1, the model calculates the pressure requirements to push the molten polymer through three characteristic sections, labeled I, II, and III on the plot. Section I is approximated as pressure flow through a tube. The second section contains a pressure drop associated with the contraction of the nozzle radius, from R_I to R_{III} . Finally, section III is also approximated as pressure flow through a tube. As an example, for

section I the volumetric flow rate can be estimated using the following equations:

$$\eta = m(T) \dot{\gamma}^{n-1} \quad (3.1)$$

$$Q = \left(\frac{n\pi}{3n+1} \right) \left[-\frac{R_1^{3n+1}}{2mL_I^{1/n}} \right] \Delta p_I^{1/n} = A_f U_{sz} \quad (3.2)$$

Where, η , $\dot{\gamma}$, m , n represent the viscosity, shear rate, consistency index, and power-law index for a Power-law viscosity model for the polymeric melt, while Q , Δp , p_1 , R_1 , A_f represent the volumetric flow rate, pressure drop, filament radius, and filament area, respectively. It should be noted that the length L_I is unknown and is a function of the filament movement speed U_{sz} . The model implies that the pressure necessary to extrude material through the capillary at the nozzle tip (region III) is considerably larger than the pressure requirements in regions I and II for a traditional 3D printer – where the nozzle diameter is usually in the order of hundreds of micrometers, while the filament diameter is either 1.75 mm or 2.85 mm [35, 43]. After equating the Filament force F_z to the following equation, one can manipulate the expressions to see that U_{sz} is proportional to $F_z^{1/n}$. For additional details, the reader is invited to carefully examine the publication by Bellini, Güceri and Bertoldi [41].

$$F_z = \Delta P A_f \approx \Delta p_{III} A_f \quad (3.3)$$

The Osswald, Puentes, Kattinger model approaches the behavior of polymer extrusion akin to the formation of a melt film of limited thickness, governed by melting with pressure flow removal [42, 35]. In this approximation, the melt film thickness δ is assumed to be orders of magnitude smaller than the diameter of the filament, and the mathematical relationship that arises from deriving equations rooted in transport phenomena yields the following proportionality. The reader is invited to read the publication by Osswald, Puentes, and Kattinger for the derivation of this mathematical relationship [42].

$$U_{sz} \propto F_z^{1/4} \quad (3.4)$$

Each model has implications for the maximum print-speed possible and the required force necessary to achieve stable printing conditions, as the requisite for extrusion is that the force applied by the filament must be enough to overcome the pressure drop that occurs within the nozzle. However, experimental results that allow discerning between the two models are scarce, mainly because measuring the force exerted by the filament, or the pressure drop within the nozzle during printing is not trivial. Publications by Go, Schiffres, Stevens, and Hart [44], as well as Go and Hart [45] managed to measure filament force as a function of filament feed rate through the use of strain gauges. The results showed that pre-heating the filament before the heated chamber through IR allowed for higher throughputs of material at reduced force requirements. Results also indicated that a higher nozzle temperature would

allow for higher feedrates at lower forces. However, these results were obtained from either a modified printer with specialized modifications that allow preheating of the filament prior to entry to the heating element, or an experimental setup built with strain gages. A traditional printer executing a print in real time was not examined. Additionally, Coogan and Kazmer [46], and Chen and Smith [47] adapted an in-situ rheometer on the print head of the ME machine with the goal of capturing pressure measurements. Serdeczny et al. [48] also developed an experimental device where a custom ME printhead was adapted to house a load-cell, that reported live force readings to a DAQ system, with the goal of relating force readings to melt pressure and extrudate swelling. All of these studies accomplished measuring print force or melt pressure, however, these devices were either an approximation of a real printer, or the measurements were acquired during steady state and with extrusion being discharged upon open air —arguably not completely representative of real printing conditions.

This work used a desktop ME machine, with a customized filament force sensor and encoder that allows real time data acquisition of filament force and filament velocity during a continuous print. This print was repeated over a variety of print conditions, as well as using two different materials. The data and derived conclusions can then be used to improve the design of future equipment in ME that allows faster print speeds, as well as allowing further refinements in the melting models available for ME systems, as each would benefit from having real data pertaining to filament force and speed.

3.3 Equipment and Materials

3.3.1 Printer Setup

For this study, an ME 3D printer (Minilab by FusedForm, Colombia) using a 0.4 mm nozzle and capable of printing 1.75 mm filament was equipped with a customized force sensor and thermistor built into the printhead, as well as an encoder that records the extruded filament length over time. The concentric force sensor was positioned just above the hot end, in a Bowden extruder architecture. These modifications permit recording and visualization of live force, filament speed, and temperature data collected during the printing process, while maintaining the original performance and functionality of the 3D printer. The generated data was collected using an Arduino board sampling at a frequency of 5 Hz, connected to MATLAB for visualization, processing, and logging. The printhead setup and encoder positioning can be seen in detail in Figure 3.2. This setup has the added benefit of allowing detection of filament slippage, if present. As shown in Figure 3.2 the external ring of the force sensor sits on the extruder carrier base, secured by the extruder carrier lid. The modified hot-end sits on the internal ring of the force sensor, secured in place by the hot-end-sensor coupler. A small clearance between the carrier lid and the sensor permits the filament force to be transmitted through the assembly and detected by the DAQ system.

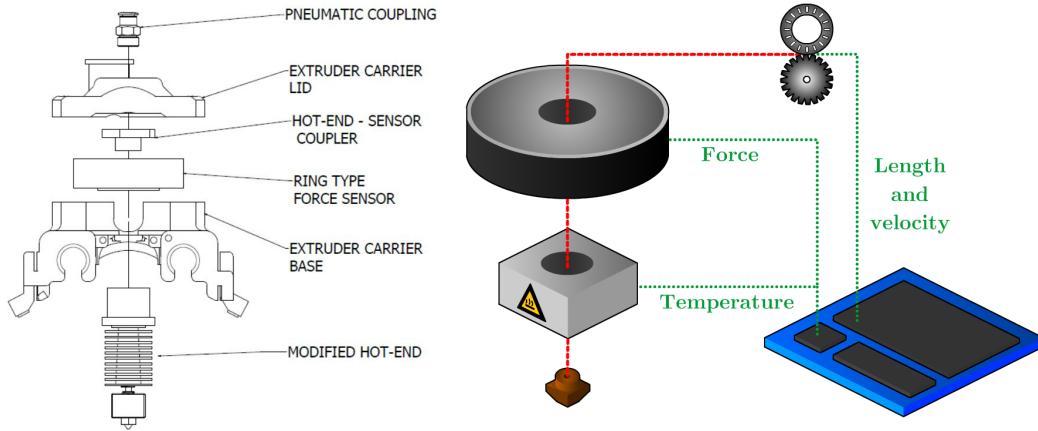


Figure 3.2: Assembly shown on the left. Schematic of sensor placement on the right.

3.3.2 Materials

Two materials were chosen to perform the experiments pertaining to this body of work: a customized ABS filament, extruded in-house, and a commercially available PLA filament, each with a nominal diameter of 1.75 mm. The ABS filament was produced using the SABIC Cycolac™ MG94 material. This is an ABS resin traditionally used for injection molding thin-walled parts, as well as ME filament. With a reported Melt Flow Index of 11.7 g/10 min, it is an ideal resin for both the ME and extrusion processes [49]. The extrusion setup consisted of a single screw extruder (Extrudex EDN 45X30D, Germany) with 45 mm screw diameter and L/D ratio of 30D. The hot melt was extruded at 205 °C through a circular die with a 4.2 mm diameter. It was then guided through a pre-skinner into a vacuum-assisted, heated water bath (Conair, USA) to cool the extrudate whilst minimizing void formation. The solidified filament then passes through a 3-axis laser micrometer (LaserLinc, USA) and a belt puller (Conair, USA) in a control loop that allows adjustment of the pull speed to keep the extrudate within specification. The desired filament dimensions were a diameter of 1.75 mm with a tolerance of ± 0.02 mm. A schematic of the extrusion setup can be seen in Figure 3.3. The PLA filament used was the commercially available "Natural PLA PRO" filament sold by Matterhackers, chosen to minimize the effect of colorants/additives to the composition of the filament. Steps were taken to ensure that all the acquired spools of material came from the same lot as to guarantee that processing conditions during the extrusion process were constant. Figure 3.4 shows the complex viscosity for each material, as measured using a 25 mm parallel plate rheometer and a 1 mm gap [35].

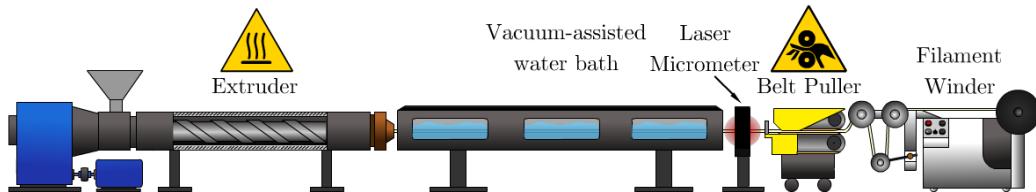


Figure 3.3: Extrusion line used to produce ABS filament

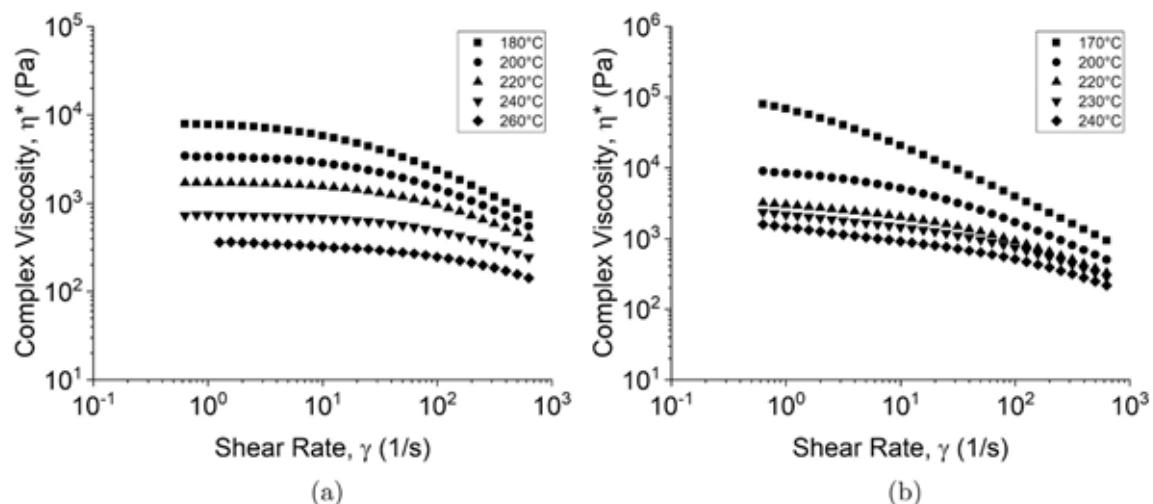


Figure 3.4: Shear rate dependency for a) PLA and b) ABS

3.4 Design of Experiments

A set of preliminary prints were designed with the goal of exploring the raw data, post processing requirements, and detection of lurking variables. In these experiments, the effect of temperature and print speed upon stable printing conditions in terms of required filament force was observed through the execution of several cylindrical toolpath files, where the printhead movement velocity was changed from 15 mm/s in increments of 5 mm/s every 15 layers, each with a thickness of 0.35 mm. To minimize the effects of varying accelerations during the test, a cylindrical geometry with a radius of 75 mm, printed in continuous helical mode was chosen as the benchmark part, as schematized in Figure 3.5. This ensures that changes in filament force and velocity stem mostly from the extrusion process and not due to toolpath considerations. To verify the effect of print temperature upon the required extrusion force, each material was printed at three different temperatures: 200, 215 and 230°C for PLA, and 215, 230 and 245°C for ABS. Close attention was paid to variability between prints performed at the same print conditions, as well as the maximum stable print speed for each material-temperature pairing.

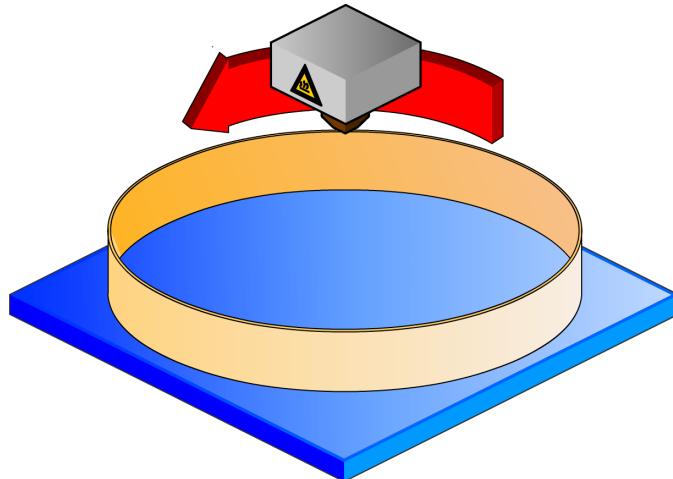


Figure 3.5: Helical cylinder experiment

Once conclusions were drawn from the preliminary set of experiments, the same geometry would be reprinted using a toolpath file that guarantees a comparable number of data points to be collected for each print speed-print temperature-material pairing, as well as utilizing optimal printing conditions and data acquisition considerations.

3.5 Results

3.5.1 Considerations Derived from Preliminary Tests

Preliminary tests show that the physical limitation of the setup lies between 20 and 25 N of force acting upon the filament. At this level of force, slippage occurs in the drive wheel mechanism that drives the material towards the nozzle, causing the mass throughput to become discontinuous. This behavior manifested itself on the data as a sudden drop in the measured filament force and was accompanied by an audible click on the driving wheel. Each material-temperature pairing reached this threshold at different print speeds. Once this phenomenon was observed in a recurring manner, data acquisition was stopped to minimize the number of outliers and noise.

The characteristics of the ME printing process can be best described using a 2D plot, calculating the filament speed against the force within the nozzle. Filament speed is derived from the length data of extruded filament recorded by an encoder, and should not be confused with the printhead movement speed, which is followed by the machine after interpreting gcode.

Figure 3.6 shows the unprocessed signals of length (bottom), the derived speed (middle) and resulting force (top) of a representative test print, made using PLA printed at 230°C. When looking at the plots of unprocessed signals, noise can be observed in the datasets of force and filament speed. Therefore, a combination of visualization and simple signal processing techniques are required to improve the signal to noise ratio of the datasets.

In this example, the first 200 seconds of data are eliminated, as these constitute the time necessary to add the brim to the print, which is necessary to stabilize the cylinder. Datapoints above 1800 seconds are removed as well due to slippage of the filament and high variability within the signal. To remove the amplified noise of the derived speed signal, a moving average filter, applied to the length data showed the best results while keeping most of the signal information. Different parameters for the sliding window of length k across neighboring elements of the length signal have been tested and number of 20 has been found most effective for this application, using a total of 7100 points in the dataset. Deriving the processed length data using the moving average with a sliding window k of 20 shows good results on the filament speed signal in comparison with the unprocessed speed signal. This is better illustrated in Figure 3.7. The oscillating nature of the post processed signal is of unknown origin, but could be related to either the duty cycle of the heater, or small variations in the filament tension that occurred as the toolpath reproduced the circular geometry of the helix. Figure 3.8 shows the comparison of processed and unprocessed force vs. filament speed data, where the reduction of noise is apparent.

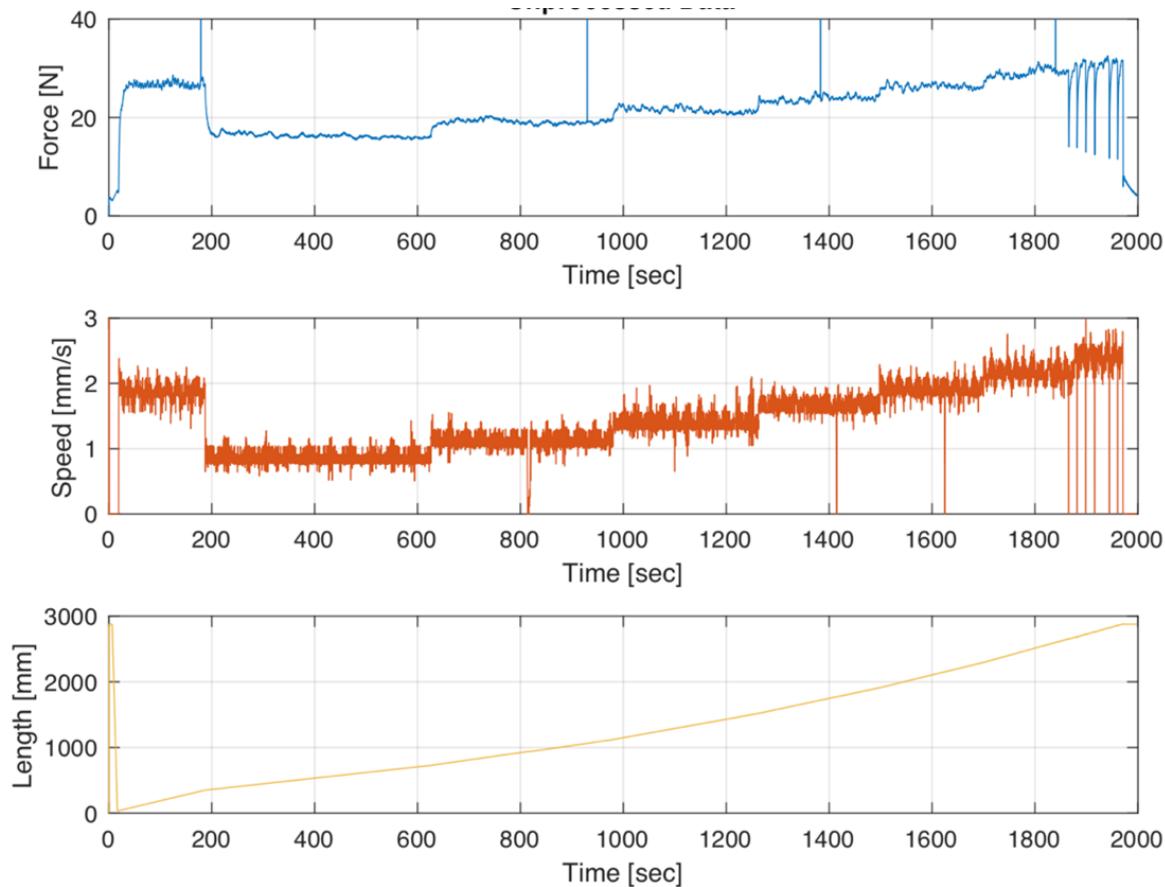


Figure 3.6: Raw signal data length (bottom), derived speed (middle), and force (top).

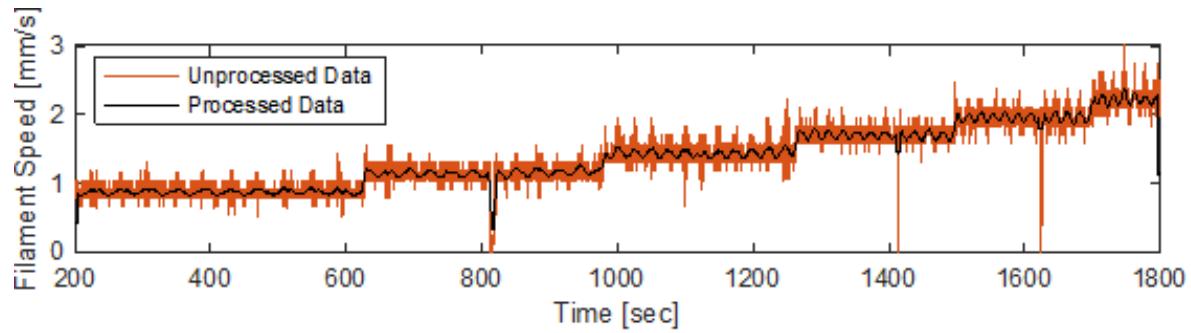


Figure 3.7: Comparison of processed and unprocessed filament speed data.

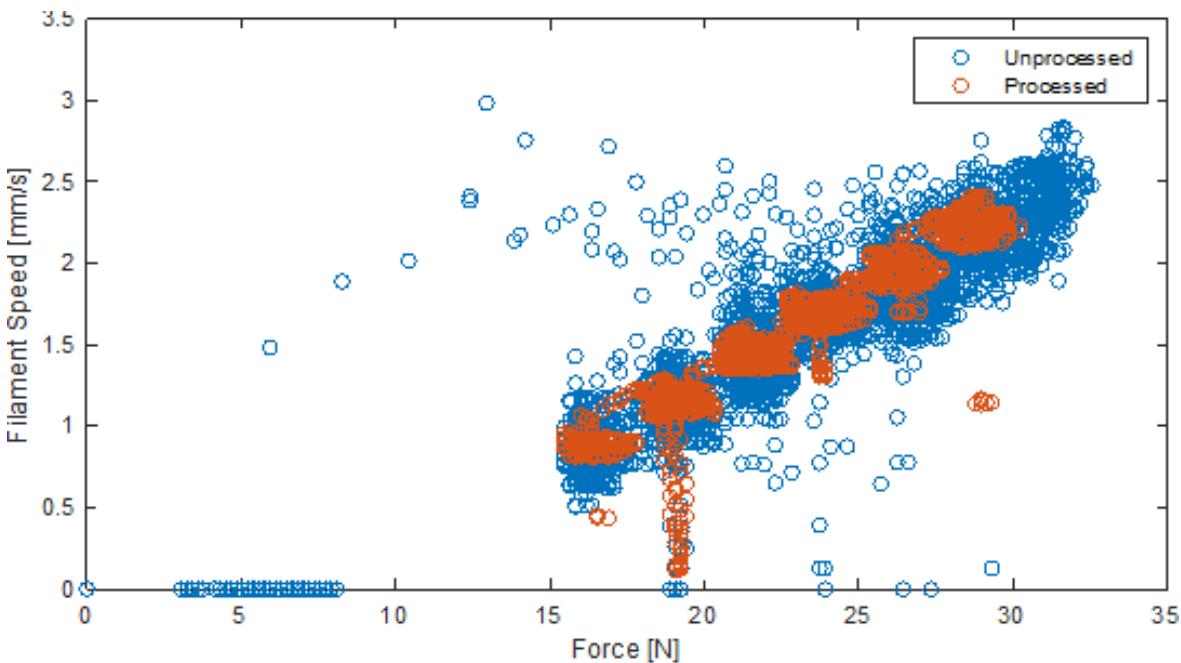


Figure 3.8: Comparison of processed and unprocessed filament speed vs. force within the nozzle of an FFF printer. PLA printed at 230°C.

Analysis of the post-processed data yielded the following general observations:

1. Increasing the hot end temperature allowed the printer to achieve higher filament velocities at lower forces.
2. Printing at 215°C proved to be unreliable and hard to reproduce, even at lower printing speeds.
3. Repeatability of results proved to be challenging. Potential causes were determined to be humidity absorbed by the material due to environmental exposure to moisture, partial clog of the nozzle after prolonged use, and additional filament tension caused by the weight of the filament spool preventing an unwinding motion of the material.

From the conclusions drawn from the preliminary results, the following adjustments were made to the experiment:

1. Filament segments were unspooled prior to any experiment to prevent noise originating from filament tension prior to a spool revolution mid-print.
2. A nozzle cleaning protocol was established, where the printhead would be disassembled, cleaned, and the brass nozzle would be burned at 500°C for 30 minutes to prevent problems arising from nozzle clogging or material degradation. This procedure was executed every time there was a switch in the print material, or every 10 prints, whichever occurred first.

3. The materials would be dried for a minimum of 3 hours prior to the start of any experiment. This is to minimize the influence of absorbed humidity upon the final materials. ABS was dried at 80°C and PLA at 50°C.
4. The toolpath file was modified to ensure that each printhead movement speed possessed the same approximate data collection time, as opposed to the same number of print layers. The selected number of layers per printhead movement speed can be seen in Table 3.1.

Table 3.1: Number of layers per print head movement speed.

Movement speed [mm/min]	Movement speed [mm/s]	Time per layer [s]	Number of layers
900	15	31.41	8
1200	20	23.56	10
1500	25	18.85	13
1800	30	15.71	15
2100	35	13.46	18
2400	40	11.78	20
2700	45	10.47	23
3100	50	9.42	25

3.5.2 Results from Modified Tests

Results obtained using the modified printing protocol yielded mostly reproducible data. Starting with the ABS material, 5 separate replications of the experiment produced Filament Speed versus Force plots that aligned well with each other. Figure 3.9 displays the collected data on the same plot with a transparency filter. Solid areas indicate a denser data cloud. The general trend appears to show an inflection point around the 2mm/s speed mark.

The same experiment performed using 245°C as the printing temperature produced similar results, although with a larger data spread. The general trend confirms what is intuitive: higher printing temperatures allows the machine to reach higher extrusion velocities at lower force requirements. The cause of the spread of the data is not fully understood: monitoring the average printing temperature for each trial showed variability of up to 6°C between replicates. It is unknown if the variations between each

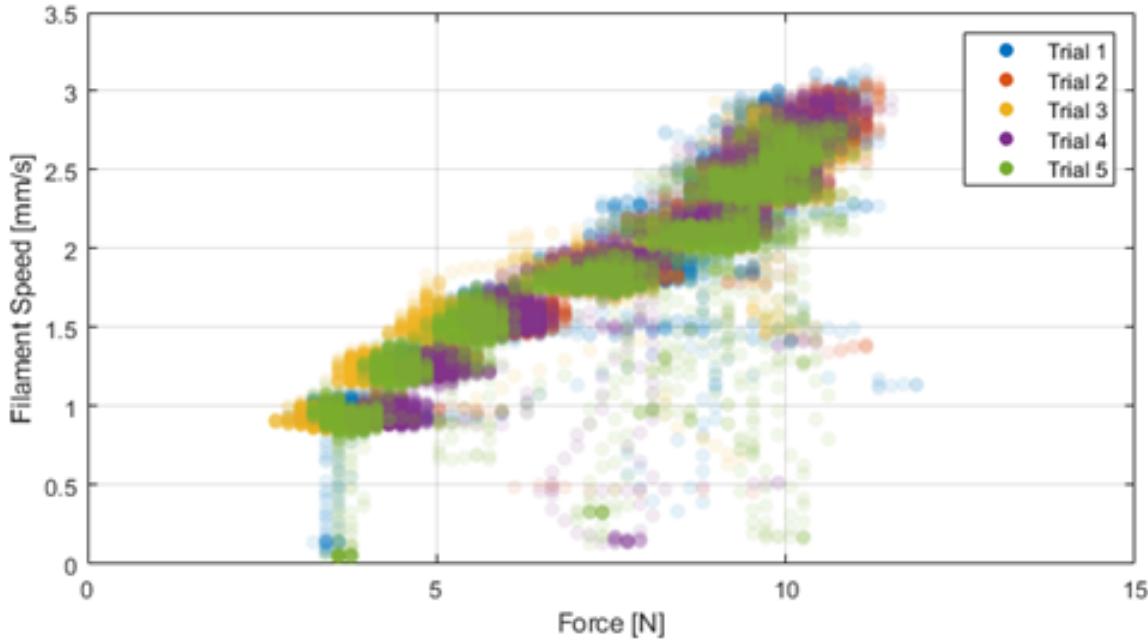


Figure 3.9: Alpha plot for 5 trials of ABS printed at 230°C.

print replication are caused exclusively by the temperature variations. Variability in the average printing temperature between replicates was most likely caused by approaching the printer's upper printing temperature limit, causing the setup difficulty at sustaining the 245°C set value. These results are summarized in Figure 3.10. Note how once again, around the 2mm/s mark there appears to be a change in the trend of the data. Table 3.2 displays the recorded average temperature for each trial.

Table 3.2: Average print temperature for ABS 245°C trials.

Trial	Average Print Temperature [°C]
1	241.38
2	231.81
3	241.23
4	251.60
5	241.21

For the PLA material, the most notable difference in behavior with respect to the ABS plots was observed in the general trend of the data clusters. For PLA printed at 230°C, an inflection point in the behavior of the data was not observed. This is

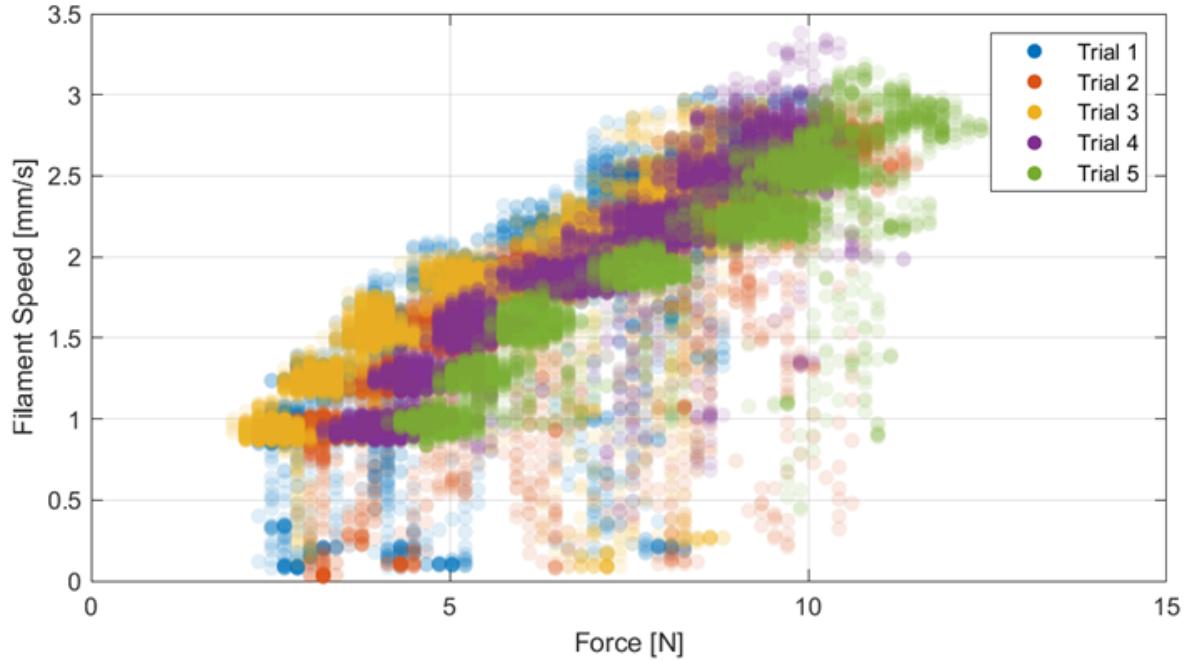


Figure 3.10: Alpha plot for 5 trials of ABS printed at 245°C.

illustrated in Figure 3.11. Additionally, the data clusters did not overlap as consistently as with ABS. It is unknown if this difference stems from material properties, or other factors not taken into consideration for this study, such as fluctuations in the filament diameter.

Comparing both materials shows that the PLA based material generally requires higher forces to achieve comparable filament velocities to its ABS counterpart. This can be seen in Figure 3.12, where a characteristic cluster of data was selected for each material and plotted on the same graph for comparative purposes. This result is particularly interesting, given that in general the complex viscosity of ABS tends to be either higher or similar to that of PLA when compared at a fixed shear rate and temperature, as can be seen in Figure 3.4.

Finally, the 245°C PLA prints faced the same issues seen with ABS at 245°C: the experiment resulted in force-velocity data clusters that agreed in trends, but not values. This can be seen in Figure 3.13, where trials 1, 2, and 3 match closely, but trials 4 and 5 are shifted to the left. Analyzing the average printing temperature for each trial reveals variations in the printing temperature, but no conclusion can be reached as to the cause of the shift in the data. Repetitions of the experiment yielded similar results, indicating that there are other factors at play other than the effect of the printing temperature. The average values for each printing temperature are shown in Table 3.3.

An important observation that can be derived from these experiments is that the data does not fully agree with the Melt Filled Nozzle model proposed by Bellini, Güceri and Bertoldi [41], nor the Melt Film Model with shear thinning behavior, as modified by

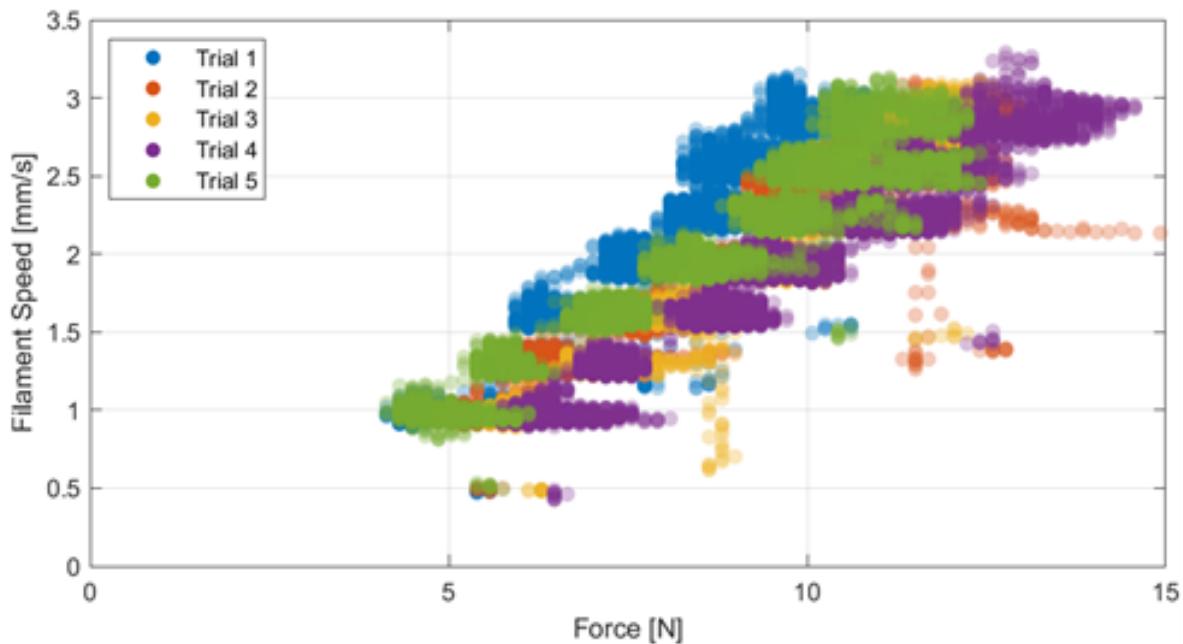


Figure 3.11: Alpha plot for 5 trials of PLA printed at 230°C.

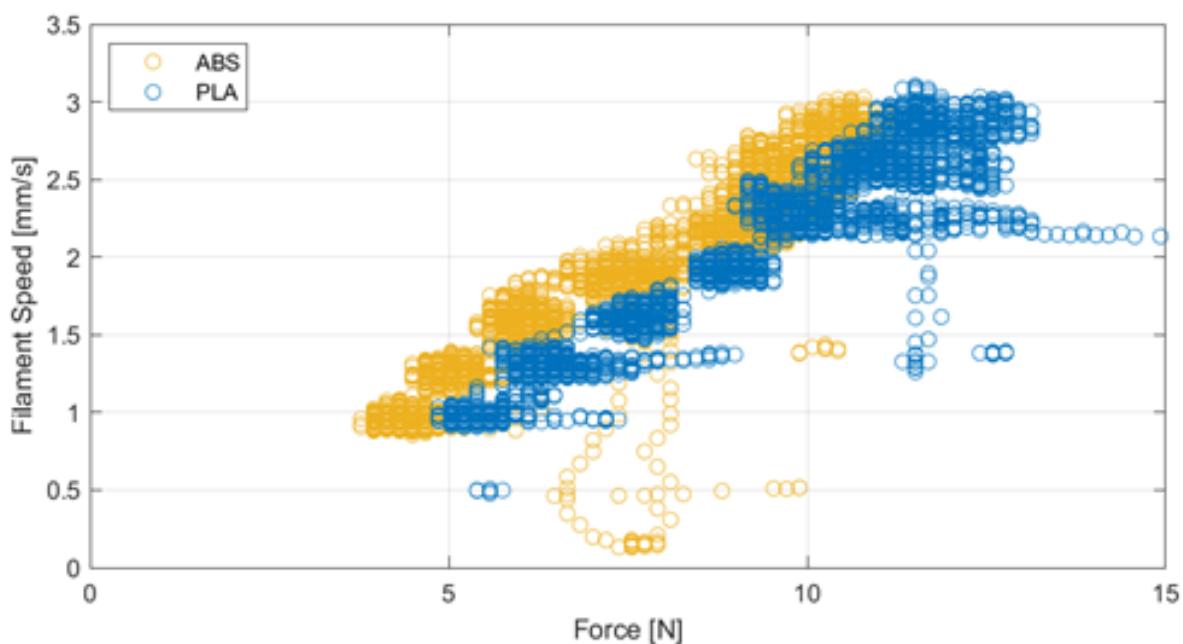


Figure 3.12: Comparison of PLA and ABS Force-velocity pairings at a printing temperature of 230°C

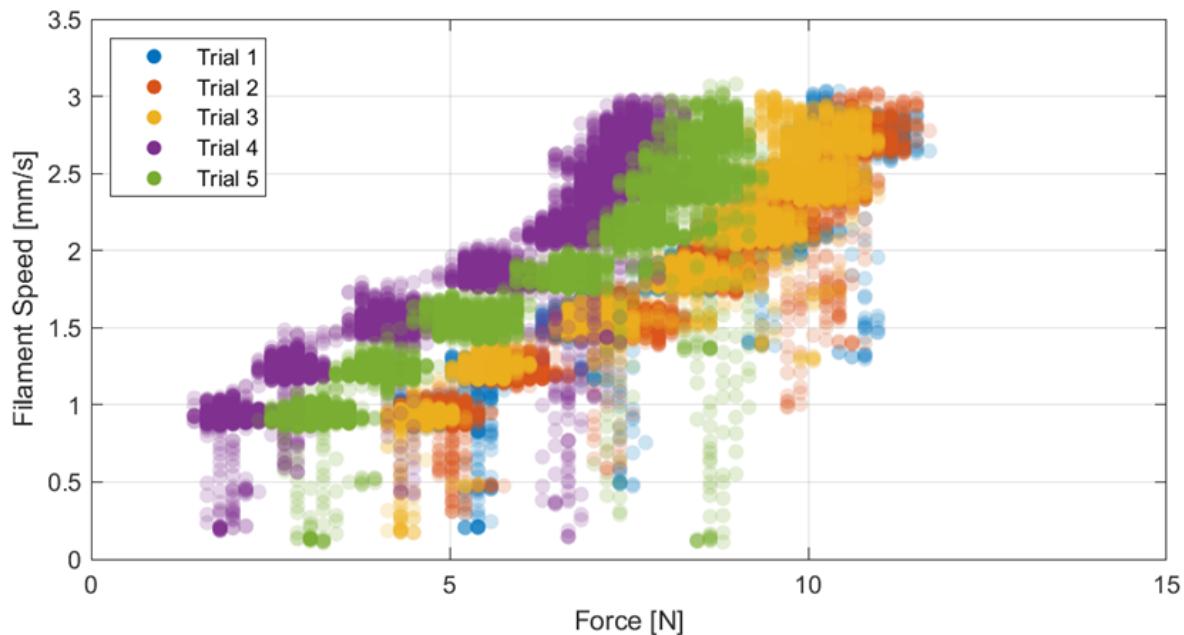


Figure 3.13: Alpha plot for 5 trials of PLA printed at 245°C.

Table 3.3: Average print temperature for PLA 245°C trials.

Trial	Average Print Temperature [°C]
1	241.38
2	231.81
3	241.23
4	251.59
5	241.21

Colón et al. [35]. Unfortunately, the physical limitations of the setup do not allow for experimentations at higher temperatures or printing speeds, as the drive wheel of the printer would slip, or the heater in the printer would not be able to sustain the elevated temperature while printing. Figure 3.14 shows how the data for a representative case of ABS printed at 230°C compares to the predictions of the Melt Filled Nozzle (FNM) and Melt Film (MFM) models [41, 42, 35]. ABS printed at 230°C was chosen for this comparison as it proved to be the most consistent condition in all replications of the experiment. The Melt Film model estimations were calculated using two different melting temperatures (T_m): the glass transition temperature of the polymer (T_g) and $T_g + 40K$, as recommended by Colón et al. during his calculations [35]. This was done since ABS is an amorphous polymer and it does not have a sharp solid-melt transition point as in the case of semi-crystalline polymeric materials. This last point represents a pain point for this model, as the equations favor sharp melt transitions. Note how the data approximates in magnitude the MF calculations using a T_m of 145°C. On the other hand, the FN model fails to capture the observed behavior in both trend and magnitude. For more details pertaining to how each model estimation was calculated, please refer to the manuscript by Colón et al., where the MG94 ABS material was used to derive the parameters necessary to fit each model [35]. Material properties used for the calculations are shown in Table 3.4, including the power law model parameters necessary to approximate the shear thinning behavior of the material [35].

Table 3.4: Material properties for ABS MG94 [35].

Material Property	Value
Melt density	954 kg/m^3
Solid density	1060 kg/m^3
Specific heat	1381 $J/kg - K$
Thermal conductivity	0.17 $W/m - K$
Heat of fusion	0 J/kg
m	$4.977 \times 10^4 Pa - s$
n	0.2847

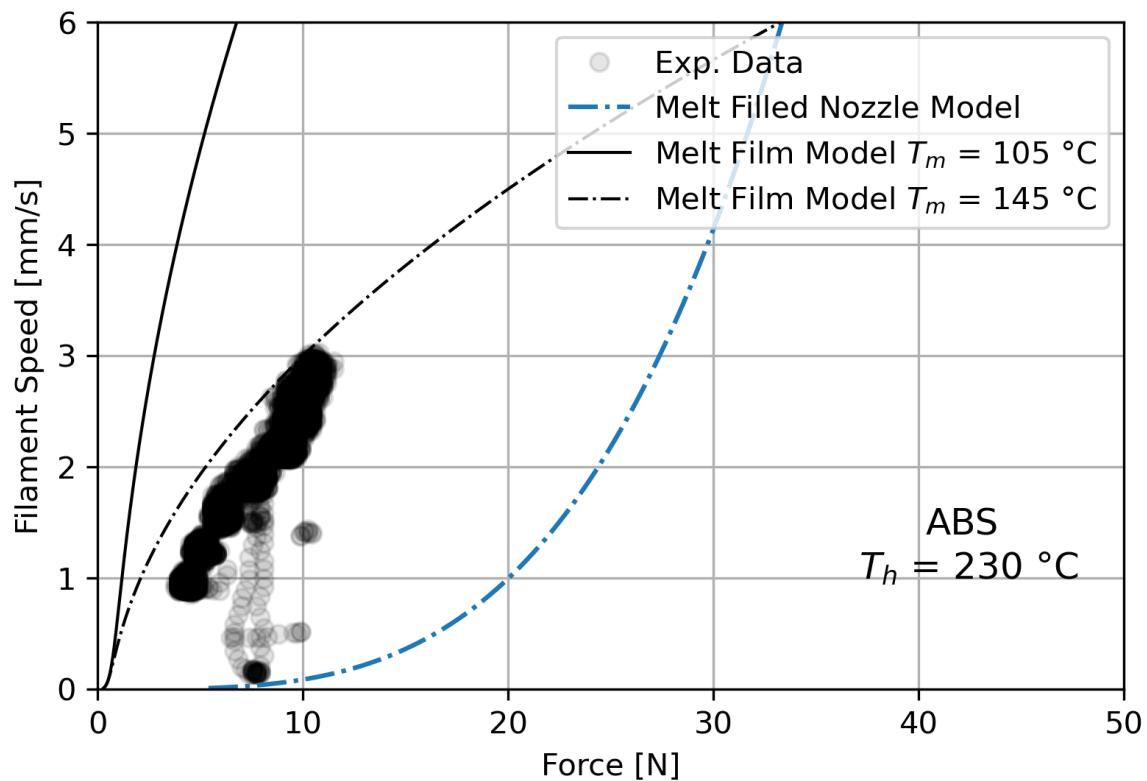


Figure 3.14: Comparison of experimental data for ABS printed at 230°C with predictions stemming from various ME melting models.

3.6 Conclusions

The experiments presented in this study show that a lot is still not fully understood regarding the underlying physics of the ME process. The differences in behavior shown between the ABS and PLA filaments prove that each material needs to be studied in a case-by-case basis to optimize part build time through maximization of print speed. Using the 230°C printing temperature of this study as an example, ABS can achieve approximately 0.5 mm/s of extra filament print speed at comparable printing conditions for the PLA material, even though their complex viscosities are comparable. Additionally, the trends observed in the data do not fully match any of the more renowned melting models available in literature. However, additional testing is necessary given that higher print temperatures and printing forces could not be achieved due to the limitations of the setup. Future work involving a different filament feeding mechanism, and longer and more potent heating sections within the printhead could lead to more comprehensive force-velocity curves. Additionally, these trials involved “best case scenario” prints that circumvent accelerations, decelerations, and filament retractions — all of which occur in real prints and potentially have tangible implications upon the behavior of the material. Ultimately a setup similar to the one used here could not only be used to properly calibrate a print to minimize part build time, but also predict other characteristics, such as mass throughput or even final part properties using resources such as machine learning algorithms. Other potential applications include development of smart systems that can auto correct or cancel failing prints according to the real-time monitoring of the filament speed and applied force.

4 Prediction of Mechanical Properties through Machine Learning

4.1 Foreword

The use of AM technologies to produce small batches of highly customized, complex parts in a reduced development cycle results extremely attractive to all industries. However, for AM parts to be fully adopted in industrial scenarios, engineers have to be able to confidently assess the structural integrity of the finished part under its intended loading conditions. This requirement is unfortunately not fully possible at the time this work was produced, partly because the mechanical properties of AM tend to be anisotropic, and partly because the relationships that exist between processing parameters, underlying physics of the process, and final mechanical part properties aren't fully comprehended. However, these obstacles present an interesting case for the application of Machine Learning (ML) techniques, where the inputs and outputs of a particular phenomenon are known, but there's a lack of explicit rules that indicate a relationship between the two.

This work uses the Material Extrusion (ME) process as a case study for the application of ML techniques to predict the final mechanical properties of a printed part. Experimental work involved producing a variety of tensile coupons, developed under various printing conditions, and where the filament extrusion speed, filament extrusion force, and printing temperature were measured in real time using machines fitted with in-line sensors. These specimens were then tested up to tensile failure, and the collective data of printing parameters, measured process indicators, and mechanical tests results were used to train a Neural Network capable of predicting the tensile failure stress.

In the context of this dissertation, this represents an alternative method for part failure prediction to construction and evaluation of a failure envelope. However, it should be noted that both methods are not mutually exclusive, and as will be discussed, the author believes they can, and should, be combined.

4.2 Introduction

The set of printing conditions that lead to an optimal part in terms of mechanical properties are not fully comprehended due to the nuances associated with the interacting effects of the processing conditions, material behavior, paired with a commonplace lack of standardization in the field of AM as a whole. However, recent advancements in processing power and algorithms have made it easier than ever to deploy Machine Learning (ML) solutions, and the intricacies of the processing-properties relationships of AM techniques represent an interesting case for development of a ML system. These excel in cases where the inputs and outcomes of a particular phenomena or task are known, but connecting the two through an explicit set of rules or relationships can result extremely complex and time consuming [12]. In this manner, ML models are *trained*, as opposed to explicitly programmed, as illustrated in Figure 4.1, where the differences between ML and traditional programming philosophies are compared.

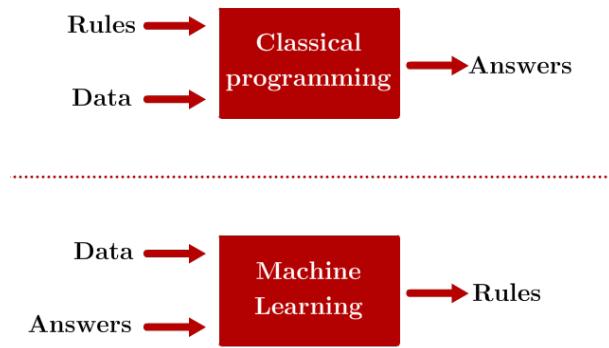


Figure 4.1: Differences between traditional programming and machine learning. [12]

The potential to apply ML solutions in the field of AM has been noted by several authors [14, 15]. Example cases include design-recommendation systems, topology optimization solutions, tolerancing and manufacturability assessment, and material classification and selection [14]. The specific algorithm applied for each case varied wildly depending on the nature of the task, but in general, Support Vector Machines (SVM) and Neural Networks (NN) appear to be the most prevalent solutions.

A Neural Network (NN) algorithm is effectively a facsimile of how biological neurons establish connections and communications with each other. In summary, the inputs of the problem are fed to a layer of nodes, or "neurons". Each node has itself a variety of connections to other neurons, and an associated weight and activation threshold, which if surpassed, triggers information transfer to its connections in subsequent layers. Finally, the information reaches the network stratus that estimates the outcome of whatever phenomena the model is trying to characterize, traditionally named the

output layer [12, 50, 51]. The weights and activation thresholds of each node are iteratively tuned as the NN architecture is exposed to a training data set, while also being compared to a separate set of data points used for validation. Once the accuracy of the model reaches its desired value, the underlying communication between the neurons is capable of making predictions based on what the input layer is perceiving. A schematic of a NN can be seen in Figure 4.2. The particular NN shown in this image is called a Deep Neural Network, as the number of layers of nodes surpasses three [50]. This type of architecture tends to be reserved for computationally complex tasks, such as text recognition or image processing.

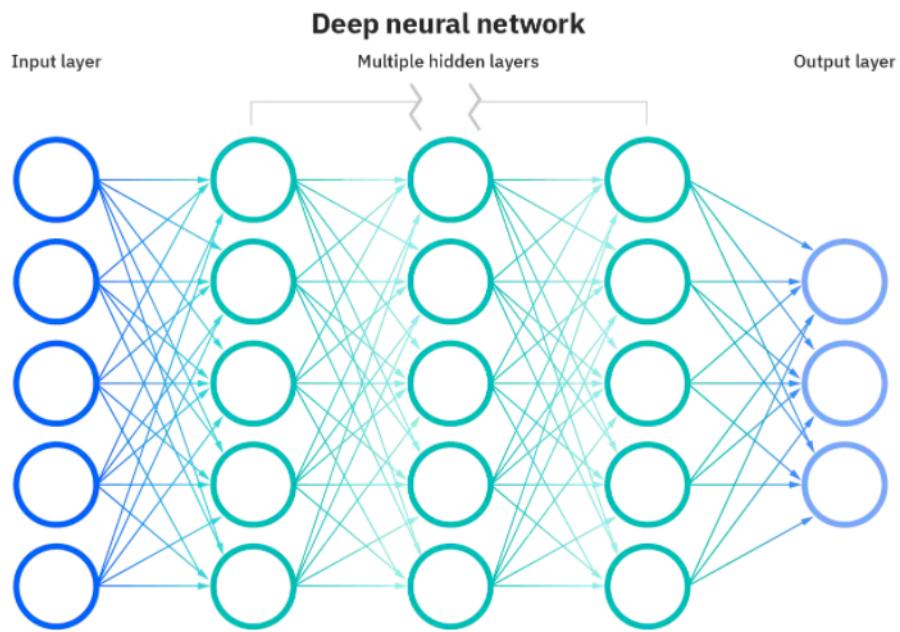


Figure 4.2: Schematic of a NN [50]

The capability of NNs to model complex behaviors is rooted in the mathematical operations that happen behind the scenes. Each neuron behaves effectively as its own mini linear regression model, represented in Equation 4.1. Here X_i and W_i represent one of the node's m inputs and its associated weight respectively.

$$\sum_{i=1}^m W_i X_i + bias = z \quad (4.1)$$

The weighed sum of the inputs can then be used as is, or passed through an activation function. This signals the generation of an output that can then be used at face value, or transmitted to subsequent nodes if a threshold is surpassed. Assuming for the purposes of this example that the threshold is zero, and the activation function is the Heaviside step function, the output of a neuron can be computed as:

$$\text{Heaviside}(z) = \begin{cases} 1 & \text{if } z \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

Concatenating nodes in a forward fashion creates the concept of levels, or *layers* in a network. When all neurons in a layer are fully connected to the nodes in the previous level, this is typically named a *Dense* layer. Arranging more than one dense layer in series results in a NN [12, 51].

The weights of each neuron are iteratively tuned in a process that involves penalizing the model using a loss function, that compares the predictions of the model with true output values using example data. This process is effectively an optimization task where the goal is to minimize the loss function. A schematic of the process can be seen in Figure 4.3, using a two layer network architecture as an example.

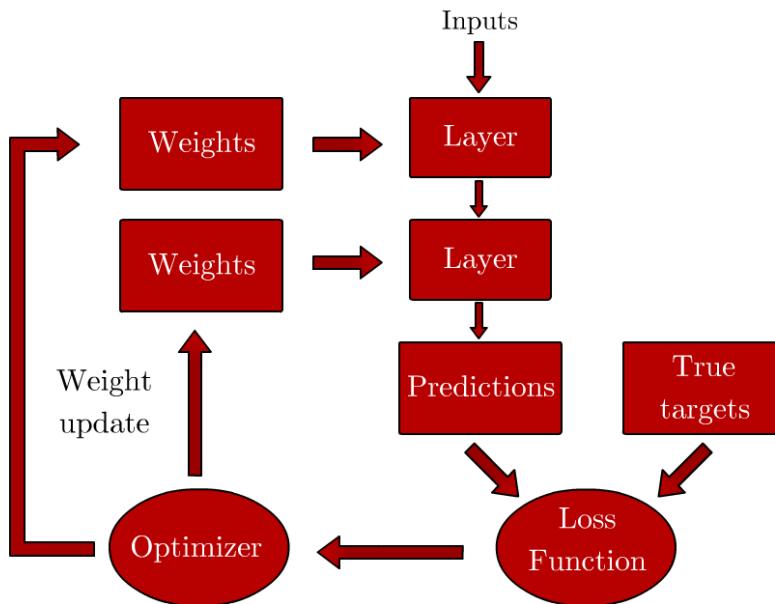


Figure 4.3: Iterative process of NN parameter tuning [12]

Unfortunately, while NNs are powerful, highly adaptable predictive models, the contributions of each input are usually hidden behind a veil of complexity that makes it almost impossible for the user to comprehend what is happening behind the scenes —fittingly, much in the same way we do not fully understand how brains and cognition work. Luckily, recent developments in interpretability of models allow a peek inside the inner machinations of a NN, normally considered a "black box" model. Of particular interest, is the application of Shapley values to aid in the understanding of how a trained ML model is affected by each of its inputs.

A Shapley value is a concept that stems from game theory. To paraphrase and over-simplify the formal definition, if a coalition C collaborates to produce a value V ,

the Shapley value is simply how much each member of the coalition contributed to its final output [52, 53]. For instance, it can determine the average contribution of each player, to the output of the team in a game; how much profit did each members of a corporation produce throughout a year; and more importantly for the contents of this work, *how much is each input of a model contributing to its output*. The formal mathematical definition is computationally expensive and escapes the scope of this work, but a numerical approximation using Monte-Carlo sampling, proposed by Štrumbelj et al. [54] states that, for any model f capable of making a prediction using the vector x with M features as an input, the Shapely value ϕ_j for that particular input can be approximated as follows:

$$\frac{1}{M} \sum_{m=1}^M (\hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m)) = \hat{\phi}_j \quad (4.3)$$

Where $\hat{f}(x_{+j}^m)$ and $\hat{f}(x_{-j}^m)$ represent predictions made by the model using two synthetic datapoints, constructed using a random entry from the database, called z . The former term is the instance of interest, but all values in the order after feature j are replaced by feature values from the sample z . Similarly, the latter term, $\hat{f}(x_{-j}^m)$ is constructed in the same manner as $\hat{f}(x_{+j}^m)$, but it also replaces feature j by its counterpart from datapoint z [53].

To facilitate its applicability to ML, Lundberg and Lee created the SHAP (Shapley Additive Explanations) method to explain individual predictions. The SHAP method effectively models "any explanation of a model's prediction as a model itself", termed an *explanation model* [55]. Thus, any model f can be approximated using an explanation model g using the equation below, where z' represents a simplified version of input z , such that $z = h_x(z')$:

$$g(z') \approx f(h_x(z')) \quad (4.4)$$

Finally, for the purposes of SHAP, the explanation model of choice has each effect ϕ_i attributed to a feature, and the sum of the effects of all feature attributions approximates the outputs of the original model f . This explanation model is represented by the equation below.

$$g(z') = \phi_0 + \sum_{m=1}^M \phi_i z'_i \quad (4.5)$$

Using Equations 4.3 and 4.5, one can use Shapley values to identify the importance of features in a respective model. This allows the user to make decision pertaining to the architecture of the model, as well as appreciating the impact a particular input can have on the output of the model —something that would be difficult to do without this resource. These mathematical principles are built into the SHAP Python library out of the box [55].

Given the factors outlined this far, the fundamental goal of this research is to predict ME part mechanical performance by training a NN with data generated through the use of sensors built into a 3D printer. This tool can then be used to predict final mechanical properties of the part based on the data generated during the print. The features selected as controlled variables for the Design of Experiments (DoE) were Layer Height (LH), Nozzle Diameter (ND), and Print Speed (PS). These parameters were chosen based on previous research that shows that these slicing parameters had a tangible impact upon the final tensile strength of ME coupons [6, 7]. Coupons are to be printed in both 0 deg and 90 deg orientations so the model can predict both the highest and lowest possible mechanical properties of each particular printing condition. Additional attention will be paid to the changes in required print force as these parameters are varied to produce the mechanical test coupons. The models produced through this research will be analyzed using Shapley values through the SHAP method to draw conclusions regarding the impact of the selected features of the model.

4.3 Experimental Methods

4.3.1 Design of Experiments

The target values of the predictive NN are the required average extrusion force (F) for the print, Tensile Strength of the coupon (σ_t), and the Elastic Modulus of the sample (E). In order to capture a variety of printing conditions, the selected controlled variables were varied in a three level, full-factorial experimental layout, which can be seen in Table 4.1. Each printing condition was replicated 2 times to account for variability in the samples, and all combinations were reproduced using both a 0° and 90° orientation.

Table 4.1: Controlled variables in the Design of Experiments

Variable	Levels
Layer Height (LH) [mm]	0.1, 0.2, 0.4
Nozzle Diameter (D_N) [mm]	0.3, 0.4, 0.8
Print Speed (PS) [mm/min]	1200, 2400, 3600

4.3.2 Equipment and methods

A set of 4 identical customized ME 3D printers (Minilab by FusedForm, Colombia) fitted with sensors capable of recording the force exerted by the filament upon the nozzle, discrete measurements of temperature, and changes in the extruded length over

time were used to reproduce a variety of tensile coupons. A schematic of the printer setup can be seen in Figure 4.4. The data was collected using an Arduino board sampling at a frequency of 5 Hz, connected to MATLAB for visualization, processing, and logging. A Butterworth filter was applied to amplify the signal-to-noise ratio of the outputs of the system. The force sensor and encoder have an innate uncertainty of $\pm 20g$, and 200 Pulses per Revolution respectively. The code for the data acquisition and filtering can be found in Appendix D. A random set of printing conditions were reproduced on all printers to monitor printer-to-printer variability.

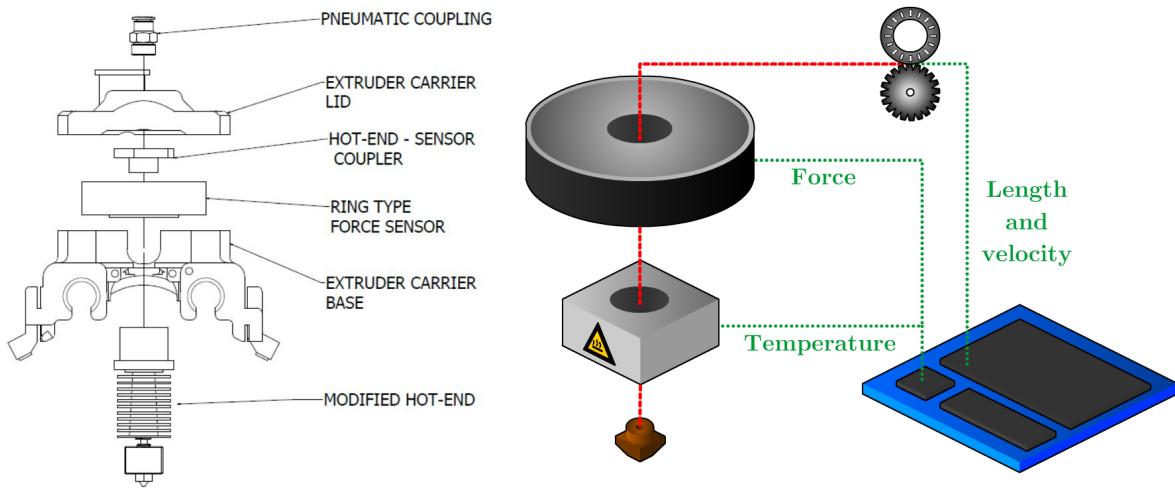


Figure 4.4: Schematic of modified ME printer with sensors

To minimize the impact of lurking variables, the filament was extruded in-house, using the SABIC Cycloc MG94 ABS material, using an extrusion line that maintained the filament diameter within the target of 1.75 ± 0.05 mm using a vacuum assisted water bath, a laser micrometer, and a conveyor belt in a control loop. Each filament was dried for a minimum of 3 hours at $65^\circ C$ and kept in a drybox attached to the printer during manufacturing. Bed leveling was performed after 10 prints, or each time a change in material or nozzle was necessary. Nozzles were burned off in a furnace every time they were switched to minimize the influence of clogging.

Each print consisted of four rectangular tensile coupons of dimensions 25 mm by 100 mm by 3.2 mm, chosen to strike a balance between being relatively quick to print, having at least 50 mm of gauge length, and fitting in the jaws of the tensile testing equipment. Each coupon was printed in a part-by-part manner (as opposed to having a single layer of the print construct a slice of all coupons), to approximate real printing behavior as much as possible. Since a single experimental run yielded four coupons, post-processing was required to separate the force-data-speed pairings for each specimen. A pause was introduced between each part that would allow discerning when one specimen print was finished, and the next started. A schematic of the process is shown in Figure 4.5.

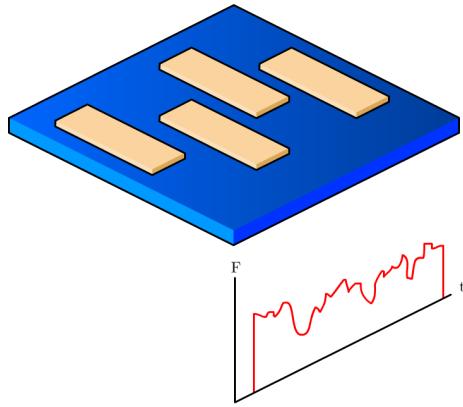


Figure 4.5: Schematic of print experiment

Additionally as an exploratory experiment, the geometric information of the filament was collected and paired to the rest of the information stemming from the in-line measurements of a handful of prints, to assess if variations in the filament geometry resulted in notable changes in the required print force. This data was attained through the use of a laser micrometer and a conveyor belt, pulling the material at a constant, known speed. The process yielded discrete measurements of the filament diameter and ovality as a function of filament length and time. A schematic of the process can be seen in Figure 4.6.

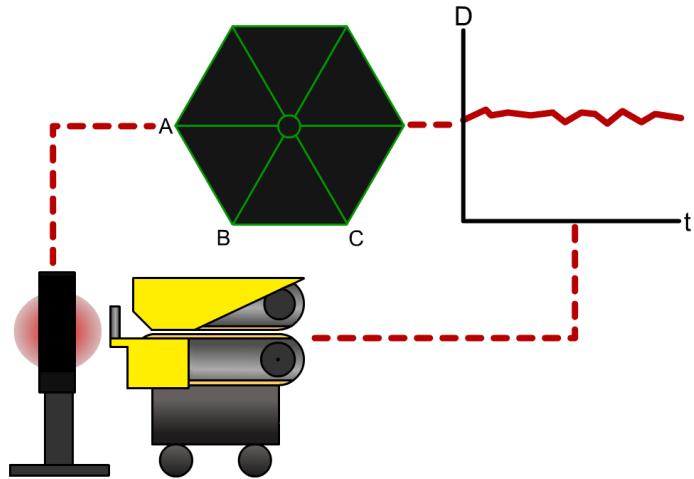


Figure 4.6: Filament geometry information, acquired through a laser micrometer

Tensile testing was performed using an Instron 5967 dual column universal testing machine, fitted with a 30 kN load cell. All data acquisition was handled through the accompanying Instron Bluehill 3 software. A movement speed of 5 mm/min was used

to deform the 50 mm gage section of the specimens, with all deformations being logged using an extensometer. To protect the samples from excessive gripping force, emery cloth tabs were used [2]. This setup can be seen in Figure 4.7. The final step of the process involved matching the printing data to its corresponding mechanical test results, and assembling everything in a single database. This was done using a Python code that would search the directory of the raw data and construct a *.csv* file including all the pertaining information of an experimental run in a row. This step is necessary for the ML code to quickly read and understand the information that it is attempting to model. This code can be found in Appendix E.



Figure 4.7: Tensile testing setup

An additional metric that can aid in identifying underlying systematic issues or relationships between the print and its mechanical properties is comparing the expected filament speed against its experimentally measured counterpart. The theoretical filament speed can be calculated using a simple volumetric balance. The volume of an extruded bead can be approximated as $ND.D_N.l$, where l represents the bead length. Given that mass must be conserved, this implies that the volume of the bead V_{out} must be equal to the volume supplied by the incoming filament V_{in} resulting in the equation below, where D_f and L represent the filament diameter and length supplied to extrude the bead respectively:

$$V_{out} = LH.D_N.l = V_{in} = \frac{\pi \cdot D_f^2 \cdot L}{4} \quad (4.6)$$

Manipulating Equation 4.6 to solve for L results in:

$$L = \frac{LH \cdot D_N \cdot l \cdot 4}{\pi \cdot D_f^2} \quad (4.7)$$

Now, one can use the expected time to print a bead of dimensions l to solve for the expected filament speed S_t .

$$t = \frac{l \cdot 60s}{PS} \quad (4.8)$$

Finally, combining Equations 4.7 and 4.8 yields the theoretical filament speed, S_t . The equation includes a conversion factor to solve in units of mm/s .

$$S_t = \frac{4 \cdot LH \cdot D_N \cdot PS}{\pi \cdot D_f^2 \cdot 60s} \quad (4.9)$$

4.3.3 Experimental Results

Effect of Variations in Filament Diameter upon Extrusion Measurements

The effect of fluctuations in the filament geometry upon the required extrusion force is unknown, but are expected to be proportional given the current filament extrusion models. As an exploratory experiment, the data stemming from the measurements of the laser micrometer were matched to its corresponding data readings resulting from the printer sensors. The condition chosen to test this experiment was $LH = 0.2$ mm, $D_N = 0.4$ mm, $PS = 2400$ mm/min. Matching the filament diameter and ovality measurements to the corresponding speed and force readings from the sensors resulted in inconclusive results. This can be seen in Figure 4.8, where the force remains nearly constant despite the variations of filament diameter and ovality.

Pairing the geometric information of the filament to its corresponding print data proved to be an extremely time consuming task, given that both sets of information came from different devices, and that in order to get proper matching, the starting point of the print had to be accurately pinpointed as length zero. The inconclusive nature of these results, and the additional time required to further explore this phenomena lends itself well to be explored as future work, with a device that measures the filament geometry directly within the printer, as to facilitate exploring trends over more printing conditions.

Printer Sensor Data Analysis

Generally speaking, the average measured filament speed was very consistent within a particular print condition. Standard deviation for filament print speed data varies between 6.6×10^{-4} and 1.8 mm/s. In contrast, the measured print force varied drastically between experimental runs, and even within the same print. As an example, four print coupons produced in a single print using $LH = 0.2$ mm, $D_N = 0.4$ mm, $PS = 2400$ mm/min had average forces of 792.66, 611.52, 516.86, and 554.91 g respectively.

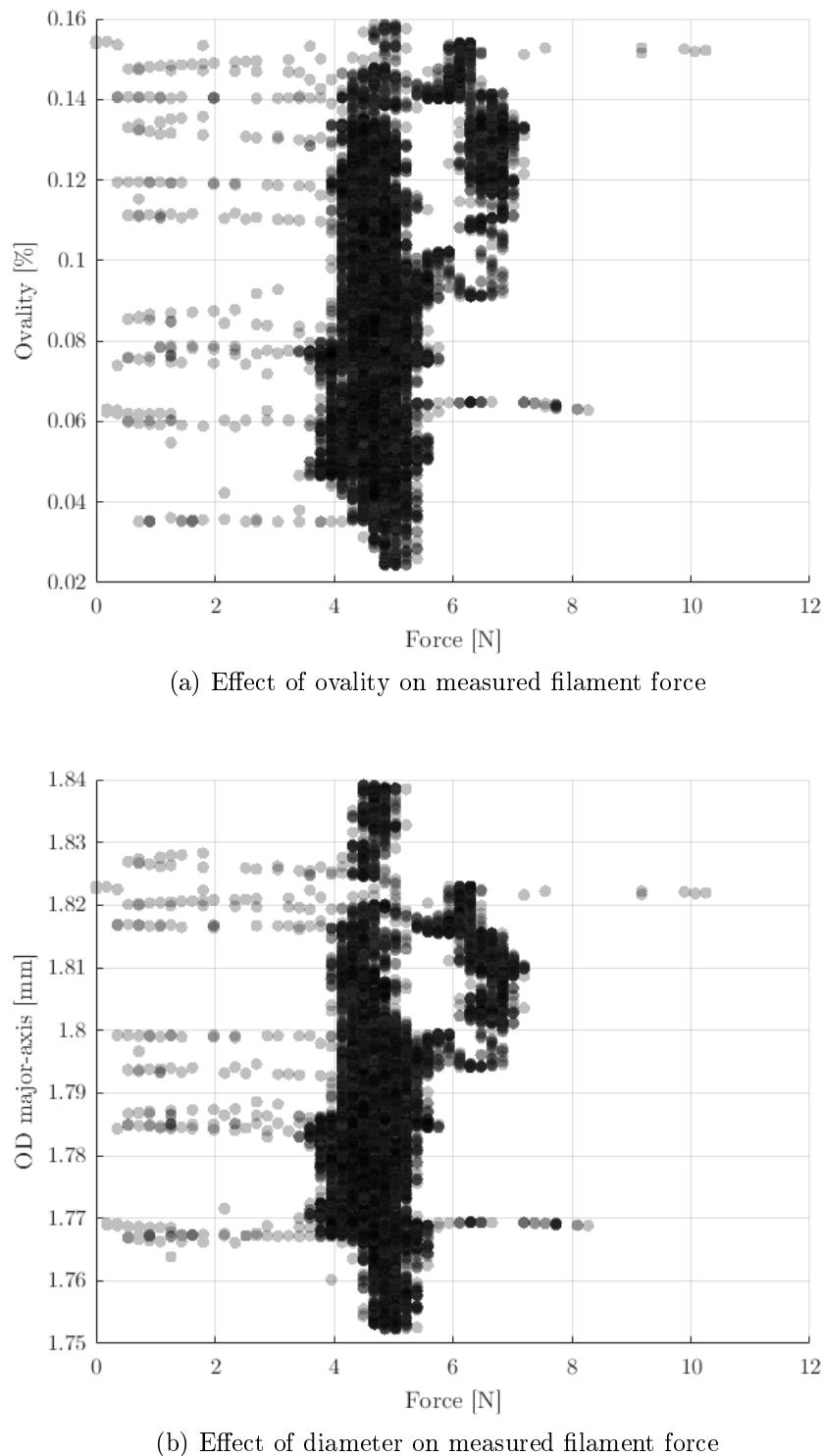


Figure 4.8: Effect of filament geometry on extrusion force

The standard deviation of the measured print force ranged between 16 and 475 g, with higher variability observed in prints involving the smallest nozzle diameter combined with the largest layer height. The conditions selected to test experimental reproducibility between printers showed no higher variability than fluctuations of measured force and filament speed within the same printer, with the highest discrepancy between printers in the range of 100 g and 1×10^{-3} mm/s respectively.

For most conditions, the measured filament speed approximated the theoretical value fairly closely, with most specimens achieving a real-to-expected speed ratio between 0.95 – 0.97. However, a couple of generalizations can be made for certain combinations of experimental values that had ratios below 0.9. These involved the fastest *PSs* combined with a *LH* that was larger than the *D_N*, namely, a 0.3 mm*N_D* paired with either a 0.4 mm or 0.8 mm *LH*. It is believed that this is caused by having filament force requirements that go beyond the capabilities of the machine, resulting in slippage at the drive wheel of the system. This effect was present in more conditions in the 90° orientation, in particular, conditions that involved the 3000mm/min gantry speed. This is theorized to be caused by a shorter travel distance between contiguous beads, implying that the acceleration of the system was not sufficient to attain the desired filament speed before the toolpass required a change in direction.

Tensile Testing Results

Box-whisker plots were used to identify outliers in the data using the Interquartile Range method (IQR). Once an outlier was detected, the entry of the database was replaced with the average of that particular condition. This was done as a way to avoid data-loss without introducing noise or bias to the expected response. An example plot can be seen in Figure.

4.3.4 Statistical Analysis

Analysis of the data was performed in two ways. First using the entire data set to develop a Pearson correlation matrix, and secondly, analyzed as a full-factorial experimental design, separating the data set in two: one for each printing orientation.

The Pearson correlation matrix allows discerning whether variables and a response may possess a linear correlation with each other, with a coefficient of 1 (or -1) indicating perfect linear correlation. It should be noted that a low Pearson index does not mean no correlation, it simply implies that the relationship may not be linear in nature [51]. The resulting matrix can be seen in 4.3.4. Here it can be seen that out of all responses of interest, only the force shows high likelihood of linear correlation with the *LH*, *D_N*, *PS*, and *S*.

The factorial experimental analysis of variance (ANOVA) yields results relating to how the input variables can have direct or interacting effects upon the measured response. Starting with the 0° orientation, the ANOVA test determined that using a 95% confidence interval, the print force is directly affected by the main effects of

Table 4.2: Pearson Correlation Matrix

	O	LH	D_N	PS	S	F	E	σ_T
O	1	-0.056	0.063	0.042	0.003	0.044	-0.108	-0.835
LH	-0.056	1	-0.018	-0.011	0.6	0.583	-0.097	-0.077
D_N	0.063	-0.018	1	0.051	0.522	-0.367	-0.232	-0.267
PS	0.042	-0.011	0.051	1	0.464	0.403	-0.081	-0.1
S	0.003	0.6	0.522	0.464	1	0.398	-0.209	-0.176
F	0.044	0.583	-0.367	0.403	0.398	1	0.015	-0.014
E	-0.108	-0.097	-0.232	-0.081	-0.209	0.015	1	0.215
σ_T	-0.835	-0.077	-0.267	-0.1	-0.176	-0.014	0.215	1

the selected control variables, as well as its two and three way interaction effects. A similar conclusion can be drawn for the tensile strength. Interestingly, only the Nozzle Diameter and the Print Speed have tangible main and interaction effects upon the Elastic Modulus. Results can be seen in Table 4.3, where * denotes a p-value lower than 0.001.

Table 4.3: Summary ANOVA table of 0° experiments

Effect	F		E		σ_T	
	f	p-value	f	p-value	f	p-value
LH	349.41	*	2.17	0.117	382.56	*
D_N	361.45	*	11.30	*	3.44	0.034
PS	213.43	*	9.37	*	9.17	*
$LH * D_N$	15.05	*	1.17	0.326	16.10	*
$LH * PS$	44.50	*	3.21	0.014	30.16	*
$D_N * PS$	4.77	0.001	1.22	0.304	35.95	*
$LH * D_N * PS$	3.97	*	0.67	0.714	53.64	*

Applying the same type of analysis to the 90° population yields similar results. Here, at a 95% confidence interval, PS , $LH * D_N$, and $LH * PS$ do not play a significant role in the response of the Young's modulus, while the LH does not affect σ_T . All other

Table 4.4: Summary ANOVA table of 90° experiments

Effect	F		E		σ_T	
	f	p-value	f	p-value	f	p-value
LH	416.87	*	3.37	0.039	0.37	0.693
D_N	122.68	*	25.24	*	234.96	*
PS	166.16	*	2.84	0.064	41.39	*
$LH * D_N$	13.09	*	0.62	0.649	25.51	*
$LH * PS$	66.31	*	2.33	0.063	21.86	*
$D_N * PS$	22.28	*	13.77	*	14.30	*
$LH * D_N * PS$	41.20	*	5.90	*	16.71	*

main and interaction effects are considered significant at the selected $\alpha = 0.05$ value. Results of the ANOVA test can be seen in Table 4.4.

4.3.5 Neural Network Architecture

All NNs described in this section were developed using the TensorFlow platform for Python, using the Keras API. A primary iteration of the algorithm was trained using a stratified, random split approach that maintained the ratio of 0° and 90° samples to avoid accidental introduction of biases to the NN. This ratio was estimated to be 51 : 49 respectively. The split was set to the typical 80 – 20 split, where 80% of the data goes to train the model, and the remaining 20% is stored separately and used to validate the model and check for over or underfitting. In order to facilitate the convergence of the algorithm to its optimal configuration, the data was normalized by using Equation 4.10 prior to being used by the model. In this computation, X is a feature, μ is the mean of the feature values, and σ is the standard deviation. This ensures all entries to the dataset have a comparable scale.

$$X' = \frac{X - \mu}{\sigma} \quad (4.10)$$

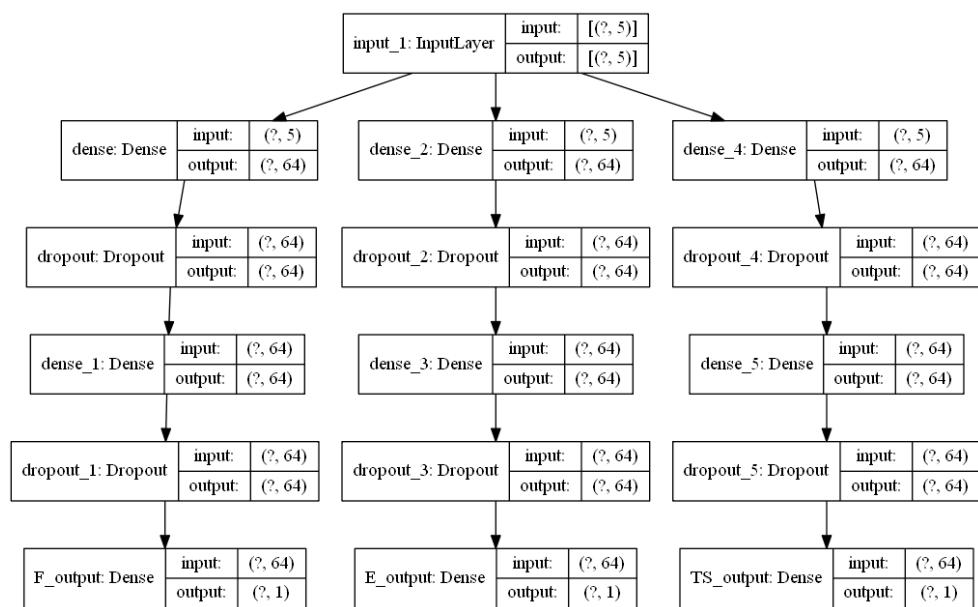


Figure 4.9: Preliminary architecture for the NN

Bibliography

- [1] Ian Gibson, David Rosen, and Brent Stucker. *Additive Manufacturing Technologies*. 2nd Ed. Springer, 2015. ISBN: 978-1-4939-2112-6. DOI: [10.1007/978-1-4939-2113-3](https://doi.org/10.1007/978-1-4939-2113-3). URL: <http://link.springer.com/10.1007/978-1-4939-2113-3>.
- [2] G A Mazzei Capote, A Redmann, C Koch, and N Rudolph. “Towards a Robust Production of FFF End-User Parts with Improved Tensile Properties”. In: *Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium*. Austin, TX, 2017, pp. 507–518.
- [3] Cole Hartman and Veronica de la Rosa. *Benefits of 3D Printing Vacuum Form Molds*. 2014. URL: <http://studiodfathom.com/wp-content/uploads/Vacuum-Forming-White-Paper-F001-5-1-2014.pdf> (visited on 02/06/2018).
- [4] Luke Van Hulle. “Robotic Off-Axis Fused Filament Fabrication”. Master Thesis. University of Wisconsin-Madison, 2017.
- [5] Caspar de Vries. *Volkswagen Autoeuropa: Maximizing production efficiency with 3D printed tools, jigs, and fixtures*. 2017. URL: <https://ultimaker.com/en/stories/43969-volkswagen-autoeuropa-maximizing-production-efficiency-with-3d-printed-tools-jigs-and-fixtures> (visited on 02/02/2018).
- [6] Carsten Koch, Luke Van Hulle, and Natalie Rudolph. “Investigation of mechanical anisotropy of the fused filament fabrication process via customized tool path generation”. In: *Additive Manufacturing* 16 (2017), pp. 138–145. ISSN: 22148604. DOI: [10.1016/j.addma.2017.06.003](https://doi.org/10.1016/j.addma.2017.06.003). URL: <http://dx.doi.org/10.1016/j.addma.2017.06.003>.
- [7] Behzad Rankouhi, Sina Javadpour, Fereidoon Delfanian, and Todd Letcher. “Failure Analysis and Mechanical Characterization of 3D Printed ABS With Respect to Layer Thickness and Orientation”. In: *Journal of Failure Analysis and Prevention* 16.3 (2016), pp. 467–481. ISSN: 15477029. DOI: [10.1007/s11668-016-0113-2](https://doi.org/10.1007/s11668-016-0113-2).

- [8] P. Obst, M. Launhardt, D. Drummer, P.V. Osswald, and T.A. Osswald. “Failure criterion for PA12 SLS additive manufactured parts”. In: *Additive Manufacturing* 21.March (2018), pp. 619–627. ISSN: 22148604. DOI: [10.1016/j.addma.2018.04.008](https://doi.org/10.1016/j.addma.2018.04.008). URL: <http://linkinghub.elsevier.com/retrieve/pii/S221486041830099X>.
- [9] P. V. Osswald, P. Obst, G. A. Mazzei Capote, M. Friedrich, D. Rietzel, and G. Witt. *Failure Criterion for PA 12 Multi-Jet Fusion Additive Manufactured Parts*. 2020.
- [10] Gerardo A. Mazzei Capote, Natalie M. Rudolph, Paul V. Osswald, and Tim A. Osswald. “Failure surface development for ABS fused filament fabrication parts”. In: *Additive Manufacturing* 28.April (2019), pp. 169–175. ISSN: 22148604. DOI: [10.1016/j.addma.2019.05.005](https://doi.org/10.1016/j.addma.2019.05.005). URL: <https://doi.org/10.1016/j.addma.2019.05.005>.
- [11] Paul V. Osswald and Tim A. Osswald. “A strength tensor based failure criterion with stress interactions”. In: *Polymer Composites* (2017). ISSN: 15480569. DOI: [10.1002/pc.24275](https://doi.org/10.1002/pc.24275). arXiv: [1206.4529](https://arxiv.org/abs/1206.4529). URL: <http://onlinelibrary.wiley.com/doi/10.1002/pc.24275/full>
- [12] François Chollet. *Deep learning with Python*. 1st ed. Shelter Island: Manning, 2018. ISBN: 9781617294433.
- [13] Xinbo Qi, Guofeng Chen, Yong Li, Xuan Cheng, and Changpeng Li. “Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives”. In: *Engineering* 5.4 (2019), pp. 721–729. URL: <http://www.sciencedirect.com/science/article/pii/S2095809918307732>.
- [14] S. S. Razvi, S Feng, A. Narayanan, Y. T. Lee, and P. Witherell. “A REVIEW OF MACHINE LEARNING APPLICATIONS IN ADDITIVE MANUFACTURING”. In: *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Anaheim, 2019.
- [15] Lingbin Meng, Brandon McWilliams, William Jarosinski, Hye Yeong Park, Yeon Gil Jung, Jehyun Lee, and Jing Zhang. “Machine Learning in Additive Manufacturing: A Review”. In: *Jom* (2020). ISSN: 15431851. DOI: [10.1007/s11837-020-04155-y](https://doi.org/10.1007/s11837-020-04155-y). URL: <https://doi.org/10.1007/s11837-020-04155-y>.
- [16] Anoop K. Sood, Raj K. Ohdar, and Siba S. Mahapatra. “Experimental investigation and empirical modelling of FDM process for compressive strength improvement”. In: *Journal of Advanced Research* 3.1 (2012), pp. 81–90. ISSN: 20901232. DOI: [10.1016/j.jare.2011.05.001](https://doi.org/10.1016/j.jare.2011.05.001). URL: <http://dx.doi.org/10.1016/j.jare.2011.05.001>.
- [17] 3D Systems. *Our Story*. 2017. URL: <https://www.3dsystems.com/our-story> (visited on 02/15/2018).

- [18] Stratasys. *About us.* 2017. URL: <http://www.stratasys.com/corporate/about-us> (visited on 02/15/2018).
- [19] ASTM International and International Organization for Standardization. *ASTM 52900:2015-Standard Terminology for Additive Manufacturing- General Principles-Terminology.* West Conshohocken, 2015. URL: <https://www.astm.org/Standards/ISOASTM52900.htm>.
- [20] 3D Hubs. *What is 3D Printing?- The definitive guide to additive manufacturing.* 2018. URL: <https://www.3dhubs.com/what-is-3d-printing{\#technologies> (visited on 02/26/2018).
- [21] Martin Baumers, Phill Dickens, Chris Tuck, and Richard Hague. “The cost of additive manufacturing: Machine productivity, economies of scale and technology-push”. In: *Technological Forecasting and Social Change* 102 (2016), pp. 193–201. ISSN: 00401625. DOI: <10.1016/j.techfore.2015.02.015>. URL: <http://dx.doi.org/10.1016/j.techfore.2015.02.015>.
- [22] Brett P. Conner, Guha P. Manogharan, Ashley N. Martof, Lauren M. Rodomsky, Caitlyn M. Rodomsky, Dakesha C. Jordan, and James W. Limperos. “Making sense of 3-D printing: Creating a map of additive manufacturing products and services”. In: *Additive Manufacturing* 1 (2014), pp. 64–76. ISSN: 22148604. DOI: <10.1016/j.addma.2014.08.005>. URL: <http://dx.doi.org/10.1016/j.addma.2014.08.005>.
- [23] Barry Berman. “3-D printing: The new industrial revolution”. In: *Business Horizons* 55.2 (2012), pp. 155–162. ISSN: 00076813. DOI: <10.1016/j.bushor.2011.11.003>. URL: <http://dx.doi.org/10.1016/j.bushor.2011.11.003>.
- [24] GE Additive. *GE Additive Manufacturing in Alabama: The Future Is Now.* 2016. URL: <https://www.ge.com/additive/press-releases/ge-additive-manufacturing-alabama-future-now> (visited on 03/01/2018).
- [25] New Balance. *The Future of Running is Here.* 2016. URL: <https://www.newbalance.com/article?id=4041> (visited on 03/07/2018).
- [26] Michelle Matisons. *Futurecraft 3D: Adidas Says It Will Turn to 3D Printed Shoes in Near Future.* 2015. URL: <https://3dprint.com/99620/futurecraft-3d-adidas-shoes/> (visited on 03/07/2018).
- [27] Sarah Saunders. *adidas and Carbon Announce Partially 3D Printed AlphaEDGE 4D LTD Shoe.* 2018. URL: <https://www.3dprint.com/203543/adidas-carbon-alphaedge-4d-ltd/> (visited on 03/07/2018).
- [28] Sung Hoon Ahn, Michael Montero, Dan Odell, Shad Roundy, and Paul K. Wright. “Anisotropic material properties of fused deposition modeling ABS”. In: *Rapid Prototyping Journal* 8.4 (2002), pp. 248–257. ISSN: 1355-2546. DOI: <10.1108/13552540210441166>. URL: <http://www.emeraldinsight.com/doi/10.1108/13552540210441166>.

- [29] C. S. Lee, S. G. Kim, H. J. Kim, and S. H. Ahn. "Measurement of anisotropic compressive strength of rapid prototyping parts". In: *Journal of Materials Processing Technology* 187-188 (2007), pp. 627–630. ISSN: 09240136. DOI: [10.1016/j.jmatprotec.2006.11.095](https://doi.org/10.1016/j.jmatprotec.2006.11.095).
- [30] Paul Osswald. "Comparison of Failure Criteria of Fiber Reinforced Polymer Composites". Term Thesis. Munich: Technische Universität München, 2015.
- [31] C.T. Sun, B.J. Quinn, and J. Tao. "Comparative Evaluation of Failure Analysis Methods for Composite Laminates." In: *U.S. Department of Transportation - May* (1996), p. 133. URL: <http://trid.trb.org/view.aspx?id=523207>.
- [32] I. I. Gol'denblat and V.A. Kopnov. "Strength of Glass-Reinforced Plastics in the Complex Stress State". In: *Mekhanika Polimerov* 1.2 (1965), pp. 70–78.
- [33] P. Obst, M. Launhardt, D. Drummer, P. V. Osswald, and T. A. Osswald. "FAILURE CRITERION FOR SLS ADDITIVE MANUFACTURED PARTS". In: *SPE ANTEC® Anaheim 2017*. Anaheim, 2017, pp. 49–54.
- [34] P. V. Osswald, P. Obst, G. A. Mazzei Capote, M. Friedrich, D. Rietzel, and G. Witt. "Failure Criterion for PA 12 Multi-Jet Fusion Additive Manufactured Parts". In: *Additive Manufacturing* July (2020). ISSN: 2214-8604. DOI: [10.1016/j.addma.2020.101668](https://doi.org/10.1016/j.addma.2020.101668). URL: <https://doi.org/10.1016/j.addma.2020.101668>.
- [35] José Luis Colón Quintana, Stefan Hiemer, Nancy Granda Duarte, and Tim Osswald. "Implementation of shear thinning behavior in the fused filament fabrication melting model: Analytical solution and experimental validation". In: *Additive Manufacturing* 37.October 2020 (2020). ISSN: 22148604. DOI: [10.1016/j.addma.2020.101687](https://doi.org/10.1016/j.addma.2020.101687).
- [36] Gerardo A Mazzei Capote. "Defining a failure surface for Fused Filament Fabrication parts using a novel failure criterion". Master thesis. Madison, WI: University of Wisconsin- Madison, 2018.
- [37] Ronald F. Gibson. *Principles of Composite Material Mechanics*. 4th Ed. CRC Press, 2016. ISBN: 0824753895. DOI: [10.1201/b19626](https://doi.org/10.1201/b19626).
- [38] Luke Van Hulle. *SciSlice- The Scientific Slicer*. 2017. URL: <https://github.com/VanHulleOne/SciSlice>.
- [39] ASTM International. "Standard test method for tensile properties of plastics". In: *ASTM International* 08 (2014), pp. 46–58. DOI: [10.1520/D0638-14.1](https://doi.org/10.1520/D0638-14.1). URL: [#0](http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Standard+Test+Method+for+Tensile+Properties+of+Plastics).
- [40] Gerardo A. Mazzei Capote, Alec Redmann, and Tim A. Osswald. "Validating a Failure Surface Developed for ABS Fused Filament Fabrication Parts through Complex Loading Experiments". In: *Journal of Composites Science* 3.2 (2019). DOI: <https://doi.org/10.3390/jcs3020049>.

- [41] Anna Bellini, Selçuk Güceri, and Maurizio Bertoldi. "Liquefier Dynamics in Fused Deposition". In: *Journal of Manufacturing Science and Engineering* 126.2 (2004), p. 237. ISSN: 1087-1357. DOI: [10.1115/1.1688377](https://doi.org/10.1115/1.1688377). URL: <https://asmedigitalcollection.asme.org/manufacturingscience/article/126/2/237/446054/Liquefier-Dynamics-in-Fused-Deposition>.
- [42] Tim A. Osswald, John Puentes, and Julian Kattinger. "Fused filament fabrication melting model". In: *Additive Manufacturing* 22.April (2018), pp. 51–59. ISSN: 22148604. DOI: [10.1016/j.addma.2018.04.030](https://doi.org/10.1016/j.addma.2018.04.030). URL: <https://doi.org/10.1016/j.addma.2018.04.030>.
- [43] Paul Oehlmann, Paul Osswald, Juan Camilo Blanco, Martin Friedrich, Dominik Rietzel, and Gerd Witt. "Modeling Fused Filament Fabrication using Artificial Neural Networks". In: *Production Engineering* (2021). ISSN: 18637353. DOI: [10.1007/s11740-021-01020-y](https://doi.org/10.1007/s11740-021-01020-y). URL: <https://doi.org/10.1007/s11740-021-01020-y>.
- [44] Jamison Go, Scott N. Schiffres, Adam G. Stevens, and A. John Hart. "Rate limits of additive manufacturing by fused filament fabrication and guidelines for high-throughput system design". In: *Additive Manufacturing* 16 (2017), pp. 1–11. ISSN: 22148604. DOI: [10.1016/j.addma.2017.03.007](https://doi.org/10.1016/j.addma.2017.03.007). URL: <http://dx.doi.org/10.1016/j.addma.2017.03.007>.
- [45] Jamison Go and A. John Hart. "Fast Desktop-Scale Extrusion Additive Manufacturing". In: *Additive Manufacturing* 18 (2017), pp. 276–284. ISSN: 22148604. DOI: [10.1016/j.addma.2017.10.016](https://doi.org/10.1016/j.addma.2017.10.016). URL: <http://dx.doi.org/10.1016/j.addma.2017.10.016>.
- [46] Timothy J. Coogan and David O. Kazmer. "In-line rheological monitoring of fused deposition modeling". In: *Journal of Rheology* 63.1 (2019), pp. 141–155. ISSN: 0148-6055. DOI: [10.1122/1.5054648](https://doi.org/10.1122/1.5054648). URL: <http://dx.doi.org/10.1122/1.5054648>.
- [47] Jingdong Chen and Douglas E. Smith. "A low-cost approach for characterizing melt flow properties of filaments used in fused filament fabrication additive manufacturing". In: *Solid Freeform Fabrication 2019: Proceedings of the 30th Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference, SFF 2019* (2019), pp. 1756–1768.
- [48] Marcin P. Serdeczny, Raphaël Comminal, David B. Pedersen, and Jon Spangenberg. "Experimental analysis of the filament feeding force in material extrusion additive manufacturing". In: *Proceedings of the Joint Special Interest Group meeting between euspen and ASPE Advancing Precision in Additive Manufacturing (2019)* (2019), pp. 141–145.
- [49] SABIC. *SABIC Cycloc MG94 Datasheet*. 2016.
- [50] IBM Cloud Education. *What are Neural Networks*. 2020. URL: <https://www.ibm.com/cloud/learn/neural-networks> (visited on 10/17/2021).

- [51] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. 2nd ed. Sebastopol: O'Reilly Media, 2019. ISBN: 9781492032649.
- [52] L S Shapley. "17. A Value for n-Person Games". In: *Contributions to the Theory of Games (AM-28), Volume II*. Ed. by Harold William Kuhn and Albert William Tucker. Princeton University Press, 2016, pp. 307–318. DOI: [doi:10.1515/9781400881970-018](https://doi.org/10.1515/9781400881970-018). URL: <https://doi.org/10.1515/9781400881970-018>.
- [53] Christoph Molnar. *Shapley Values*. 2021. URL: <https://christophm.github.io/interpretable-ml-book/shapley.html> (visited on 11/15/2021).
- [54] Erik Štrumbelj and Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions". In: *Knowledge and Information Systems* 41.3 (2014), pp. 647–665. ISSN: 0219-3116. DOI: [10.1007/s10115-013-0679-x](https://doi.org/10.1007/s10115-013-0679-x). URL: <https://doi.org/10.1007/s10115-013-0679-x>.
- [55] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Ed. by I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: <https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>.

A SABIC Cycloc MG94 Datasheet

Reproduced from SABIC's Datasheet Document [49].

Superior flow, injection molding ABS. Good impact. For thin-wall applications.

TYPICAL PROPERTIES ¹	TYPICAL VALUE	Unit	Standard
MECHANICAL			
Tensile Stress, yld, Type I, 5 mm/min	460	kgf/cm ²	ASTM D 638
Tensile Stress, brk, Type I, 5 mm/min	350	kgf/cm ²	ASTM D 638
Tensile Strain, yld, Type I, 5 mm/min	2	%	ASTM D 638
Tensile Strain, brk, Type I, 5 mm/min	18	%	ASTM D 638
Tensile Modulus, 5 mm/min	25200	kgf/cm ²	ASTM D 790
Flexural Stress, yld, 1.3 mm/min, 50 mm span	800	kgf/cm ²	ASTM D 790
Flexural Modulus, 1.3 mm/min, 50 mm span	26700	kgf/cm ²	ASTM D 790
Hardness, Rockwell R	113	-	ASTM D 785
Tensile Stress, yield, 50 mm/min	46	MPa	ISO 527
Tensile Stress, break, 50 mm/min	35	MPa	ISO 527
Tensile Strain, yield, 50 mm/min	2.3	%	ISO 527
Tensile Strain, break, 50 mm/min	40	%	ISO 527
Tensile Modulus, 1 mm/min	2450	MPa	ISO 527
Flexural Stress, yield, 2 mm/min	70	MPa	ISO 178
Flexural Modulus, 2 mm/min	2500	MPa	ISO 178
IMPACT			
Izod Impact, notched, 23°C	24	cm-kgf/cm	ASTM D 256
Izod Impact, notched, -30°C	8	cm-kgf/cm	ASTM D 256
Instrumented Impact Total Energy, 23°C	214	cm-kgf	ASTM D 3763
Instrumented Impact Total Energy, -30°C	50	cm-kgf	ASTM D 3763
Izod Impact, notched 80°10°4 +23°C	16	kJ/m ²	ISO 180/1A
Izod Impact, notched 80°10°4 -30°C	7	kJ/m ²	ISO 180/1A

TYPICAL PROPERTIES ¹	TYPICAL VALUE	Unit	Standard
IMPACT			
Charpy 23°C, V-notch Edgew 80*10*4 sp=62mm	18	kJ/m ²	ISO 179/1eA
 THERMAL			
Vicat Softening Temp, Rate B/50	98	°C	ASTM D 1525
HDT, 0.45 MPa, 3.2 mm, unannealed	95	°C	ASTM D 648
HDT, 1.82 MPa, 3.2mm, unannealed	82	°C	ASTM D 648
CTE, -40°C to 40°C, flow	8.82E-05	1/°C	ASTM E 831
CTE, -40°C to 40°C, xflow	8.46E-05	1/°C	ASTM E 831
CTE, -40°C to 40°C, flow	8.82E-05	1/°C	ISO 11359-2
CTE, -40°C to 40°C, xflow	8.46E-05	1/°C	ISO 11359-2
Vicat Softening Temp, Rate B/50	98	°C	ISO 306
Vicat Softening Temp, Rate B/120	100	°C	ISO 306
HDT/Be, 0.45MPa Edgew 120*10*4 sp=100mm	89	°C	ISO 75/Be
HDT/Ae, 1.8 MPa Edgew 120*10*4 sp=100mm	76	°C	ISO 75/Ae
Relative Temp Index, Elec	60	°C	UL 746B
Relative Temp Index, Mech w/impact	60	°C	UL 746B
Relative Temp Index, Mech w/o impact	60	°C	UL 746B
PHYSICAL			
Specific Gravity	1.05	-	ASTM D 792
Mold Shrinkage, flow, 3.2 mm (5)	0.5 - 0.8	%	SABIC Method
Melt Flow Rate, 230°C/3.8 kgf	11.7	g/10 min	ASTM D 1238
Melt Viscosity, 240°C, 1000 sec-1	1720	poise	ASTM D 3825
Density	1.04	g/cm ³	ISO 1183
Melt Flow Rate, 220°C/5.0 kg	12	g/10 min	ISO 1133
Melt Flow Rate, 220°C/10.0 kg	42	g/10 min	ISO 1133
FLAME CHARACTERISTICS			
UL Recognized, 94HB Flame Class Rating (3)	1.52	mm	UL 94

CYCOLAC™ Resin MG94

Americas: COMMERCIAL

PROCESSING PARAMETERS	TYPICAL VALUE	Unit
Injection Molding		
Drying Temperature	80 - 90	°C
Drying Time	2 - 4	hrs
Drying Time (Cumulative)	8	hrs
Maximum Moisture Content	0.1	%
Melt Temperature	205 - 245	°C
Nozzle Temperature	205 - 245	°C
Front - Zone 3 Temperature	205 - 225	°C
Middle - Zone 2 Temperature	200 - 210	°C
Rear - Zone 1 Temperature	190 - 200	°C
Mold Temperature	50 - 70	°C
Back Pressure	0.3 - 0.7	MPa
Screw Speed	30 - 60	rpm
Shot to Cylinder Size	50 - 70	%
Vent Depth	0.038 - 0.051	mm

B Failure Criterion Calculation Code

```

1 %-----%
2 %
3 %          Evaluation of Failure Criteria Models
4 %
5 %-----%
6 %
7 % Define Strength Parameters
8 %
9
10    Xt=46.4;
11    Xc=61.5;
12    Yt=41.1;
13    Yc=62.6;
14    S=32.6;
15    S45=34.0;
16    mu1=-0.04;
17    mu2=-0.25;
18    lambda=0.38;
19 %
20 % Set up a grid within the desired limits in MPa for S2 (x) and
21 % T12 (y). Within this grid values of the failure criteria will
22 % be calculated using values of S2 and T12 on a given S1 plane
23 %
24    x=-70:0.5:70;
25    y=-70:0.5:70;
26    z=0:0.5:70;
27    [S1,S2,T12]=meshgrid(x,y,z);
28
29 %
30 % Inquire the desired model from the user
31 %
32    reply = input('Which model would you like to use? [Tsai, TsaiM, Gol or GolM]: ', 's');
33 %
34 % Compute the appropriate model based on user's input, or "break"
35 % if non-existing model
36 %
37 %
38    if strcmp(reply,'Tsai')
39        disp('Evaluating the Tsai-Wu Criteria')
40        L1='Tsai-Wu Criteria';
41        criteria=Tsai(Xt,Xc,Yt,Yc,S,S1,S2,T12);
42    elseif strcmp(reply,'TsaiM')
43        disp('Evaluating the Modified Tsai Criteria')
44        L1='Modified Tsai Criteria';
45        criteria=TsaiM(Xt,Xc,Yt,Yc,S,mu1,mu2,lambda,S1,S2,T12);
46    elseif strcmp(reply,'Gol')
47        disp('Evaluating the Gol-denblat Criteria')
48        L1='Gol-denblat Criteria';
49        criteria=Gol(Xt,Xc,Yt,Yc,S,S45,S1,S2,T12);
50    elseif strcmp(reply,'GolM')
```

```

51 disp('Evaluating the Modified Gol-denblat Criteria')
52 L1='Modified Gol-denblat Criteria';
53 criteria=GolM(Xt,Xc,Yt,Yc,S,S45,mu1,mu2,lambda,S1,S2,T12);
54 else
55 disp('Model not available')
56 %break;
57 end
58 %
59 % Plot the failure criteria at values of 1., 0.75, 0.5 and 0.25
60 % interpolating on the S1-S2 grid
61 %
62 cvals = linspace(0,1,5)
63 Sx = [];
64 Sy = [];
65 Sz = -70:70;
66 figure(1)
67 %colormap(3)
68 contourslice(S1,S2,T12,criteria,Sx,Sy,Sz,cvals);
69 view(3);
70 axis([-70,70,-70,70,-70,70]);
71 daspect([1,1,1]);
72 box on
73 axis tight
74
75 %contour(S2,T12,criteria,[1,1],'-k','showtext','off')
76
77 set(gca,'FontSize',16)
78 hold on
79 % contour(S2,T12,criteria,[1,0.75],'-k','showtext','off')
80 hold on
81 % contour(S2,T12,criteria,[1,0.5],'.k','showtext','off')
82 hold on
83 % contour(S2,T12,criteria,[1,0.25], '--k','showtext','off')
84 hold on
85 % grid on
86 % grid on
87 xlabel('\sigma_{11}', 'fontsize',20) % x-axis label
88 ylabel('\sigma_{22}', 'fontsize',20) % y-axis label
89 zlabel('\tau_{12}', 'fontsize',20) % z-axis label
90
91 % colormap(gray)
92
93 % hold on
94 % x1=[-70,50];
95 % y1=[0,0];
96 % plot(x1,y1,'k')
97 % hold on
98 % x2=[0,0];
99 % y2=[-70,50];
100 % plot(x2,y2,'k')
101 % title (L1,'fontsize',20)
102 print('Figure(1)', '-depsc')
103

```

C SSIC Predictions Code

```

1 %FailSurf test
2 %clear all
3 clc
4 test =1; %Change number to calculate desired surface
5 if (test==1)
6     %G. Mazzei Master, Plots using Max Stress criterion
7     Xt=40.29;    Xc=43.91;    Yt=31.13;    Yc=57.96;    S=23.35;
8     S45p=20.80;    S45n=38.17;    mu1112=-0.0052;    mu2212=-0.2;
9     lambda=0;
10    x=-80:0.5:80;
11    y=-80:0.5:80;
12    z=-80:0.5:80;
13    [s1,s2,t12]=meshgrid(x,y,z);
14    [F_11,F_1111,F_22,F_2222,F_12,F_1212,F_1122,F_1112,F_2212]=FailSurf.tens(Xt,Xc,Yt,Yc,S,S45p,S45n,mu1112,
15        mu2212,lambda);
16    f=FailSurf.criteria(s1,s2,t12,F_11,F_1111,F_22,F_2222,F_12,F_1212,F_1122,F_1112,F_2212);
17    %g=FailSurf.graph(s1,s2,t12,f,2);
18    % hold on
19    % box on
20    % grid on
21    % axis tight
22    % daspect([1,1,1]); %data aspect ratio of 1:1
23    X=0:1:90;
24    T=X*pi/180;
25    c=cos(T);
26    s=sin(T);
27    S_1x=Xt./c.^2;
28    S_2x=Yt./s.^2;
29    S_12x=S./(s.*c);
30    Sx=(-1.*sqrt((F_1111-2.*F_1122-1.*F_1212+F_2222).*(c.^4)+(2.*F_1112-2.*F_2212).*s.*c.^3+(2.*F_1122+
31        F_1212-2.*F_2222).*c.^2+2.*F_2212.*c.*s+F_2222)+(F_11-1.*F_22).*c.^2+F_12.*c.*s+F_22)./((F_11
32        .^2-2.*F_11.*F_22-1.*F_12.^2+F_22.^2-1.*F_1111+2.*F_1122+F_1212-1.*F_2222).*c.^4+(2.*F_11.*F_12-2.*
33        F_12.*F_22-2.*F_1112+2.*F_2212).*s.*c.^3+(2.*F_11.*F_22+F_12.^2-2.*F_22.^2-2.*F_1122-1.*F_1212+2.*
34        F_2222).*c.^2+(2.*F_12.*F_22-2.*F_2212).*s.*c+F_22.^2-1.*F_2222));
35    plot(X,S_1x,'-k');
36    axis ([0 90 30 60])
37    grid on
38    box on
39    hold on
40    plot(X,S_2x,'-k');
41    plot(X,Sx,'-r','LineWidth',2)
42    xlabel('\theta [deg]', 'FontSize', 16) % x-axis label
43    ylabel('\sigma_{xx} [MPa]', 'FontSize', 16) % y-axis label
44    set(gca,'FontSize',14) %Sets the font size for axes
45    set(gca,'FontName','Arial') %Sets the font size for axes
46    plot([0 30 45 60 75 90], [41.03 36.21 36.07 35.72 34.39 34.39], 'ro')
47    %plot(15,35.57,'rx')
48    legend({'M1-1','M2-2','M1-2','00C','Data'})
49 elseif (test==2)

```

```

46 %Solving for sx as a function of theta
47 %Define tensorial components to avoid runnin calc. twice
48 tic
49 F_11=0.001023099495182;
50 F_1111=0.000566295072456799;
51 F_1112=-3.427843350651909e-05;
52 F_1122=-1.017023854135421e-04;
53 F_12=0;
54 F_1212=0.001834113595826;
55 F_22=0.007435037777640;
56 F_2212=4.840568372012957e-05;
57 F_2222=6.095129420020747e-04;
58 T=0:1:90;
59 theta=(pi/180).*T;
60 s=sin(theta);
61 c=cos(theta);
62 Sx=(-1.*sqrt((F_1111-2.*F_1122-1.*F_1212+F_2222).*(c.^4)+...
63 (2.*F_1112-2.*F_2212).*s.*c.^3+(2.*F_1122+F_1212-2.*F_2222).*c.^2+2.*F_2212.*c.*s+F_2222)+...
64 (F_11-1.*F_22).*c.^2+F_12.*c.*s+F_22)./((F_11.^2-2.*F_11.*F_22-1.*F_12.^2+F_22.^2-1.*F_1111+2.*F_1122+...
   F_1212-1.*F_2222).*c.^4+...
65 (2.*F_11.*F_12-2.*F_12.*F_22-2.*F_1112+2.*F_2212).*s.*c.^3+(2.*F_11.*F_22+F_12.^2-2.*F_22.^2-2.*F_1122...
   -1.*F_1212+2.*F_2222).*c.^2+...
66 (2.*F_12.*F_22-2.*F_2212).*s.*c+F_22.^2-1.*F_2222);
67 end

```

D Data Acquisition Code

```

1 run('close_Serial.m')
2 comPort='com4';% com 42 arduino viejo
3 [force.s,fserialFlag]=setupSerial(comPort);
4 %calCo=calibrateForce(force.s);
5 calCo.g=0.0627;
6
7 calCo.offset=0;
8 load('thermistor_lookup.mat')
9
10 format short
11
12 % Open a new figure — add start/stop and close serial buttons
13 % initialize the figure that will be plot if it does not exist
14 %clear button
15 if(~exist('h','var')||~ishandle(h))
16     h=figure(1);
17     ax=axes('box','on');
18 end
19 if (~exist('button','var'))
20     button=uicontrol('Style','pushbutton','String','Stop',...
21         'pos',[0 0 50 25],'parent',h,...
22         'Callback','stop_call_vector','UserData',1);
23 end
24 % clos_call function calls everytime it is pressed
25 if(~exist('button2','var'))
26     button=uicontrol('Style','pushbutton','String','Close Serial Port',...
27         'pos',[250 0 150 25],'parent',h,...
28         'Callback','closeSerial','UserData',1);
29 end
30
31 buf_len=2;
32 plot_window=60; % 30 data points
33 % Rolling plot of data
34 % buf_len=100;
35 % index=1:buf_len;
36 % force_data=zeros(buf_len,1);
37 % length_data=zeros(buf_len,1);
38 tic
39 % data Collection and plotting
40 % while the figure window is open
41 while (get(button,'UserData'))
42 index=1:buf_len;
43 % force_data=zeros(buf_len,1);
44 % length_data=zeros(buf_len,1);
45 % Get the new value from acceleometer
46 [forces, length, temp1, temp2]=readData(force,calCo, thermistor_lookup);
47 forces
48 length
49 % Update the rolling plot. Append the new reading to the end of the
50 % rolling plot data. Drop the first value

```

```

51     actual_time=toc;
52     force_data(buf_len,1:3)=[buf_len,actual_time, forces];
53     length_data(buf_len,1:3)=[buf_len,actual_time, length];
54     speed_data(buf_len,1)=buf_len;
55     speed_data(buf_len,2)=actual_time;
56     speed_data(buf_len,3)=(length_data(buf_len,3)-length_data(buf_len-1,3))/(length_data(buf_len,2)-
57         length_data(buf_len-1,2));
58     speed_data_2(buf_len,1)=buf_len;
59     speed_data_2(buf_len,2)=actual_time;
60     speed_data_2(buf_len,3)=movmean(speed_data(buf_len,3),10);
61     temp1_data(buf_len,1:3)=[buf_len,actual_time,temp1];
62     temp2_data(buf_len,1:3)=[buf_len,actual_time,temp2];
63     if (buf_len<=plot_window)
64         d_min=1;
65     t_ini=0;
66     else
67         d_min=buf_len-plot_window;
68     t_ini=(force_data(d_min,2));
69     end
70
71     subplot(5,1,1);
72     plot(force_data(d_min:buf_len,2),force_data(d_min:buf_len,3),'r');
73     axis([t_ini ceil(actual_time) min(force_data(d_min:buf_len,3))-5 max(force_data(d_min:buf_len,3))+5]);
74     xlabel('time')
75     ylabel('F_z(g)')
76
77     subplot(5,1,2);
78     plot(length_data(d_min:buf_len,2),length_data(d_min:buf_len,3),'r');
79     axis([t_ini ceil(actual_time) min(length_data(d_min:buf_len,3))-5 max(length_data(d_min:buf_len,3))+5]);
80     xlabel('time')
81     ylabel('L_z(mm)')
82
83     subplot(5,1,3);
84     plot(speed_data(:,2),speed_data(:,3),'r');
85     % plot(speed_data_2(:,2),speed_data_2(:,3),'b');
86     % axis([t_ini ceil(actual_time) min(speed_data(d_min:buf_len,3))-1 max(speed_data(d_min:buf_len,3))+1]);
87     axis([t_ini ceil(actual_time) min(speed_data_2(d_min:buf_len,3))-1 max(speed_data_2(d_min:buf_len,3))+
88           +1]);
89     xlabel('time')
90     ylabel('U_z(mm/s)')
91
92     % subplot(6,1,4);
93     % plot(speed_data(:,2),speed_data(:,3),'r');
94     % plot(speed_data_2(:,2),speed_data_2(:,3),'b');
95     % axis([t_ini ceil(actual_time) min(speed_data(d_min:buf_len,3))-1 max(speed_data(d_min:buf_len,3))+
96           +1]);
97     % axis([t_ini ceil(actual_time) min(speed_data_2(d_min:buf_len,3))-1 max(speed_data_2(d_min:buf_len,3))+
98           +1]);
99     xlabel('time')
100    ylabel('U2_z(mm/s)')
101
102    subplot(5,2,7);
103    plot(temp1_data(d_min:buf_len,2),temp1_data(d_min:buf_len,3),'r');
104    axis([t_ini ceil(actual_time) 0 300]);
105    % plot(temp1_data(:,2),temp1_data(:,3),'r');
106    % axis([0 ceil(actual_time) 0 300]);
107    xlabel('time')
108    ylabel('T1 (C)')
109    subplot(5,2,8);
110    plot(temp2_data(:,2),temp2_data(:,3),'r');
111    axis([0 ceil(actual_time) 0 300]);
112    xlabel('time')
113    ylabel('T2 (C)')
114    drawnow;
115    buf_len=buf_len+1;

```

```

112 end
113
114 %set ':' for sake of validating vector and csv creation. will need to
115 %change to '4285:29735'
116
117
118 mean_temp = mean(temp2_data(:,3));    %section of data that corresponds to bulk of print, ok
119 mean_speed = mean(speed_data_2(:,3));   %section of data that corresponds to bulk of print, ok
120 mean_force = mean(force_data(:,3));     %section of data that corresponds to bulk of print, NOT OK- data
121      should not be averaged! what else to do?
122
123 % d1 = designfilt('lowpassiir','FilterOrder',12, ...
124 %     'HalfPowerFrequency',0.03,'DesignMethod','butter');
125 %
126 % temp_filt= filtfilt(d1,data(:,2));
127
128 to_csv = [mean_temp(:), mean_speed(:), mean_force(:)]    %matrix with all vectors to be output to csv
129
130 writematrix(to_csv,'temp_speed_force.csv')  %where do we want this csv to be saved to
131
132
133 %% Extra section to create csv using data from print for export and import to python
134 %maybe just add this directly above? right after the loop ends!
135
136 length_values = length_data(:,3);    %vector of length values
137 diff_length_values = diff(length_values);  %difference in length along vector
138
139 eps = .1; %threshold
140 t_diff = (abs(diff_length_values) >= eps); % create all ones and zeros vector
141 d_diff = diff(t_diff); %difference along ones and zeros path
142
143 startIndex = find(d_diff < 0)+1;
144 endIndex = find(d_diff > 0)+1;
145 endIndex(1)=[];           %doing this gets rid of the first zero set from before the print begins
146
147 clear length      %need to do this because 'length' is a function, but also has been defined as a variable
148
149 final_index = startIndex(end);
150
151 if length(startIndex) ~= length(endIndex)
152     startIndex(end)=[];
153 end
154
155 duration = endIndex-startIndex+1;
156
157 m_zeros = [startIndex,endIndex,duration];
158 m_zeros_copy = m_zeros;
159
160
161 clear length
162
163 for index = length(startIndex):-1:1
164     duration(index);
165     if duration(index) < 80 || duration(index)>125
166         m_zeros(index,:)= [];
167     end
168 end
169
170 %start
171 %here is where I add the digital filter section
172 d2 = designfilt('lowpassiir','FilterOrder',12, ...
173     'HalfPowerFrequency',0.002,'DesignMethod','butter');
174
175 % y = filtfilt(d2,force_data(:,3));

```

```

176
177 %temp_data_filtered = filtfilt(d2,temp2_data(:,3));           comment out
178 %speed_data_filtered = filtfilt(d2,speed_data_2(:,3));       comment out
179 %force_data_filtered = filtfilt(d2,force_data(:,3));
180
181
182 %mean_force1 = mean(force_data_filtered(m_zeros(1,2):m_zeros(2,1),1));
183 %mean_force2 = mean(force_data_filtered(m_zeros(2,2):m_zeros(3,1),1));
184 %mean_force3 = mean(force_data_filtered(m_zeros(3,2):m_zeros(4,1),1));
185 %mean_force4 = mean(force_data_filtered(m_zeros(4,2):final_index(1),1));
186 %here is where I end the digital filter section
187 %end
188
189
190 %mean_temp1 = mean(temp1_data(m_zeros(1,2):m_zeros(2,1),3));
191 %mean_temp2 = mean(temp1_data(m_zeros(2,2):m_zeros(3,1),3));
192 %mean_temp3 = mean(temp1_data(m_zeros(3,2):m_zeros(4,1),3));
193 %mean_temp4 = mean(temp1_data(m_zeros(4,2):final_index(1),3));
194 %mean_temp5 = mean(temp2_data(m_zeros(5,2):m_zeros(6,1),3));
195
196 %mean_speed1 = mean(speed_data_2(m_zeros(1,2):m_zeros(2,1),3));
197 %mean_speed2 = mean(speed_data_2(m_zeros(2,2):m_zeros(3,1),3));
198 %mean_speed3 = mean(speed_data_2(m_zeros(3,2):m_zeros(4,1),3));
199 %mean_speed4 = mean(speed_data_2(m_zeros(4,2):final_index(1),3));
200 %mean_speed5 = mean(speed_data_2(m_zeros(5,2):m_zeros(6,1),3));
201
202 % mean_force1 = mean(force_data(m_zeros(1,2):m_zeros(2,1),3));
203 % mean_force2 = mean(force_data(m_zeros(2,2):m_zeros(3,1),3));
204 % mean_force3 = mean(force_data(m_zeros(3,2):m_zeros(4,1),3));
205 % mean_force4 = mean(force_data(m_zeros(4,2):final_index(1),3));
206 % mean_force5 = mean(force_data(m_zeros(5,2):m_zeros(6,1),3));
207
208 %mean_temp = [mean_temp1;mean_temp2;mean_temp3;mean_temp4];
209 %mean_speed = [mean_speed1;mean_speed2;mean_speed3;mean_speed4];
210 %mean_force = [mean_force1;mean_force2;mean_force3;mean_force4];
211
212 %to_csv = [mean_temp(:,1), mean_speed(:,1), mean_force(:,1)]; %matrix with all vectors to be output to csv
213 to_csv = [speed_data_2(:,1), speed_data_2(:,3), force_data(:,3)];
214 to_csv_2 = speed_data;
215 %writematrix(to_csv,'filtered_speed_force.csv'); %where do we want this csv to be saved to
216 writematrix(to_csv,'raw_speed_force.csv')
217 writematrix(to_csv_2,'raw_speed.csv');
218
219
220 %% Section 4:
221
222 % Troubleshooting section
223
224 % FORCE PLOTS
225
226 figure('Name','Force')
227
228 plot(force_data(:, 1), force_data(:,3))
229
230 ylim([0 1800])
231 xlabel('index')
232 ylabel('Force/g')
233 hold on
234
235 %plot(force_data(:, 1), force_data(:,3))
236
237 % LENGTH AND SPEED PLOTS
238
239 figure ('Name','Length & Speed')
240

```

```
241 subplot(2,1,1); plot(length_data(:,1), (length_data(:,3)))
242 xlabel('index')
243 ylabel('Length/mm')
244 axis tight
245
246 subplot(2,1,2); plot(speed_data(:,1), (speed_data(:,3)))
247 xlabel('index')
248 ylabel('Filament Speed/mm/s')
249
250 axis tight
251 %% Section 5:
252 % FORCE v.s. SPEED PLOTS
253
254 %figure ('Name','NO filter & filter')
255
256 %xlim([2 5])
257 scatter(speed_data(:,3),force_data(:,3), 'MarkerFaceAlpha',0.2, 'MarkerEdgeAlpha',0.2) %Without filter
258
259 xlabel('Filament Speed/mm/s')
260 ylabel('Force/g')
261
262 %axis tight
263
264 %subplot(2,1,2); scatter(speed_data(:,3),force_data(:,3), 'MarkerFaceAlpha',0.2, 'MarkerEdgeAlpha',0.2)%With
265 %filter
266 %axis tight
267
268
269 %%
270 clear force_data length_data index forces length temp1_data temp2_data speed_data
```

E Database Assembly Code

```
#####
#      3D Printing Research Ingest
#      Programmed by Evan Grubis, 2021
#      ### Denotes Comments for the software
#      # Denotes commented sections for various reasons
#####

# INSTRUCTIONS:
# Download the "Data Collection" folder on Box using the three dots (...)
# Unzip the file using your operating system
# Run the program and select the unzipped [Datapath] folder

outputfilename = 'data.csv'

#####
#      Import required libraries
#
#####

import os
import re
import glob
import csv
import pandas as pd
import tkinter as tk
from tkinter import filedialog

#####

###      Define functions
###

#####

### Asks the user to select the root file path to the data
def getrootfilepath():
    downloads = os.getcwd()                      # finds the downloads folder
    downloads = downloads.split("\\\\")
    downloads = downloads[0] + "/" + downloads[1] + "/" + downloads[2] + "/Downloads"
    root = tk.Tk()
    root.withdraw()
    root_path = filedialog.askdirectory(title='Please_select_the_unzipped_[Data_Collection]_folder',
                                         initialdir=downloads)    # creates dialogue window
    return root_path                                # returns the selected folder

### Reads all the filenames in a given directory
def get_files(directory):
    filenames = glob.glob(directory + "/0_degree_orientation/*/*.csv")
```

```

# searches 0 degree folder
filenames = filenames + glob.glob(directory + "/90_degree_orientation/*.csv")
# searches 90 degree folder
print(str(len(filenames)) + ".CSV_files_discovered")
# outputs how many CSV files were found
return filenames

### Splits, cleans, and extracts useful info from a filepath
def get_info(infopath):
    returnme = []
    # array to return
    infopath = re.split('[.]', infopath)[0]
    # gets rid of the '.csv'
    infopath = infopath.split('\\')
    # swap \ for /
    tensilepath = infopath[0] + "/" + infopath[1]
    # create the tensile data path for later
    end = infopath[2]
    infopath = infopath + end.split('_')
    # split at underscores
    cleanpath = infopath[0] + '/' + infopath[1] + '/' + infopath[2] + '.csv'
    # reconstructs filepath
    data = pd.read_csv(cleanpath, sep=',', header=None).values
    # reads value from filepath
    infopath = infopath[3:]
    # 0           1           2           3           4           5           6
    # ReplicateNumber_Length_Width_LayerHeight_NozzleDiameter_PrintSpeed_Orientation

    #
    #
    # Orientation LH ND PS AT FS MF M TS Run Sample

    i = 1
    tensiledata = glob.glob(tensilepath + "/*/*.csv") # init variable to keep track of sample
    if len(tensiledata) > 0: # reads the tensile data
        tensiledata = tensiledata[0].split('\\') # if the tensile data exists
        tensiledata = tensiledata[0] + "/" + tensiledata[1] + "/" + tensiledata[2] # swap \ to /
        tensdata = pd.read_csv(tensiledata, sep=',', header=None, skiprows=6).values # read tensile data
    else:
        tensdata = [] # if the tensile data doesn't exist, make empty array
    for array in data: # run through all the data from the original CSV file
        if len(tensdata) > 0: # if tensile data exists
            if int(infopath[0]) == 1: # if Run = 1
                ten = tensdata[1 + i][12] # set temp variables to correct data
                mom = tensdata[1 + i][5]
            elif int(infopath[0]) == 2: # if Run = 2
                ten = tensdata[5 + i][12] # set temp variables to correct data
                mom = tensdata[5 + i][5]
            elif int(infopath[0]) == 3: # if Run = 2
                ten = tensdata[9 + i][12] # set temp variables to correct data
                mom = tensdata[9 + i][5]
            else: # if Run != 1 or 2
                ten = None # set temp variables to null
                mom = None
        else: # if tensile data does not exist
            ten = 'n/d' # set temp variables to "n/d"
            mom = 'n/d'
    if len(infopath) == 8: # if the data has comments
        row = [infopath[6], float(infopath[3]) / 10, float(infopath[4]) / 10, infopath[5], array[0], array[1],
               array[2], mom, ten, infopath[0], i, infopath[7], cleanpath] # set the data as needed+
    else: # if the data does not have comments

```

```

        row = [infopath[6], float(infopath[3]) / 10, float(infopath[4]) / 10, infopath[5], array[0], array[1],
               array[2], mom, ten, infopath[0], i, "", cleanpath] # set the data as needed
    returnme.append(row) # append the data to the return array
    i += 1 # increment the sample number
return returnme # once all data has been read, return the array

#####
### Begin Main Loop
###
#####

if __name__ == "__main__":
    print("Initiated_root_directory_popup") # prints some information to aid with debugging
    rootpath = getrootfilepath() # triggers the filepath popup function
    print("Root_directory_assignment_successful")
    print("The_root_path_is_" + rootpath) # prints root filepath

    big_array = [['Orientation_[deg]', 'Layer_Height_[mm]', 'Nozzle_Diameter_[mm]', 'Print_Speed_[mm/min]',
                  'Avg_Temperature_[C]', 'Avg_Filament_Speed_[mm/s]', 'Avg_Force_[g]', 'Modulus_[MPa]',
                  'Tensile_Stress_at_Tensile_Strength_[MPa]', 'Run', 'Sample', 'Notes', 'Filepath']] # manually sets the first row of the CSV file with the titles
    paths = get_files(rootpath) # triggers the get_files function with the rootpath

    for filename in paths: # prints all extracted info in a readable format
        try:
            info = get_info(filename) # try element to elegantly catch any errors in formatting
        except Exception as e: # get the info from the filename
            print(filename + "_could_not_be_read") # catches said exceptions
            print(e) # prints helpful information
            continue # moves on to next filename
        big_array += info # adds the info to the final file

    lengthBig = len(big_array) - 1 # finds the length of the array without the first row
                                    # (which we added manually)
    if lengthBig <= 0:
        print("Zero_entries_were_found..Ensure_you_are_using_the_correct_folder..If_this_was_an_unzipped_folder,.."
              "you_may_have_to_go_down_another_level_to_get_to_the_correct_folder") # prints an error if no info has been added at this point
    else:
        print(str(lengthBig) + "_sample_entries_found") # prints how many samples were found
    try:
        with open(outputfilename, "w") as my_csv: # write CSV file
            csvWriter = csv.writer(my_csv, lineterminator = '\n')
            csvWriter.writerows(big_array)
            print("File_" + outputfilename + "'_created_successfully")
        launchExcel = 1
    except:
        print("ERROR:_Make_sure_you_closed_the_CSV_file_before_running_the_program._The_file_was_not_written")
        launchExcel = 0

    if launchExcel == 1: # launch excel with the output file open
        os.system("start_EXCEL.EXE_" + outputfilename)
        print(outputfilename + "_opened_in_Excel_successfully")
    except:
        print("Excel_was_unable_to_be_launched")

```