12-2021

Design and Fabrication of a Polymer FDM Printer Capable of Build Parameter Monitoring and In-Sit Geometric Monitoring Via Photogrammetry

Travis Roberts
robert2@clemson.edu

12-2021

Design and Fabrication of a Polymer FDM Printer Capable of Build Parameter Monitoring and In-Sit Geometric Monitoring Via Photogrammetry

Travis Roberts

DESIGN AND FABRICATION OF A POLYMER FDM PRINTER CAPABLE OF BUILD PARAMETER MONITORING AND IN-SITU GEOMETRIC MONITORING VIA PHOTOGRAMMETRY

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

---

by
Travis Roberts
December 2021

Accepted by:
Dr. Cameron Turner, Committee Chair
Dr. Garrett Pataky
Dr. Adam Hoover

ABSTRACT

Additive manufacturing, or 3D printing, is a complex process that creates free-form geometric objects by sequentially placing material in a location to construct an object, usually as a layer-by-layer process. One of the most widespread methods is Fused Deposition Modeling (FDM). FDM is used in many of the consumer-grade polymer 3D printers available today. While consumer grade machines are cheap and plentiful, they lack many of the features desired in a machine used for research purposes and are often closed-source platforms. Commercial-grade models are more expensive and are also usually closed-source platforms that do not offer flexibility for modifications often needed for research. This research focuses on the design and fabrication of a machine to be used as a test bed for research in the field of polymer FDM processes. The goal was to create a platform that tightly controls and/or monitors the FDM build parameters so that experiments can be repeated with a known accuracy. The platform offers closed loop position feedback, control of the hot end and bed temperature, and monitoring of environment temperature and humidity. Additionally, the platform is equipped with cameras and a mechanism for in-situ photogrammetry, creating a geometric record of the print throughout the printing process. Through photogrammetry, backtracking and linking of process parameters to observable geometric defects can be achieved. The controls system and instrumentation are built on an open flexible paradigm enabling customization as necessary for future research.

TABLE OF CONTENTS

Page

CHAPTER

Table of Contents (Continued)

LIST OF TABLES

Table Page

LIST OF FIGURES

List of Figures (Continued)

List of Figures (Continued)

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

CHAPTER ONE

THE PROBLEM

INTRODUCTION AND MOTIVATION

Traditional manufacturing methods, such as machining, molding, and forging, use process qualification techniques to certify part quality because they are significantly faster and cheaper than certifying individual parts. This is only possible because the physics and mechanics of these processes are understood well enough to believe that control of process parameters will produce the expected results. This is currently not the case for additive manufacturing and **no additive manufacturing processes are currently qualified to produce parts for aerospace or defense applications** [1]. This demonstrates a clear knowledge gap that needs to be explored and overcome to progress this technology.

Process feedback for machine tools gives the ability to monitor and adjust machine parameters during the fabrication process and decreases the number of defective parts. This principle has been applied prolifically to subtractive machining, but limited research has been done on its application to additive manufacturing systems. Additionally, process feedback provides the ability to certify a part without slow and expensive after-process evaluations. However, each part currently needs to be qualified using evaluation methods that require additional skilled personnel before it can be put into service in a critical application. Process feedback can change this by certifying a part as it is being made, reducing wasted time and material by mitigating defects. The NIST

Roadmap for Additive Manufacturing lists development of real-time process monitoring techniques and feedback systems as key goals for the advancement of additive manufacturing techniques [2].

While many of the faults, or defects, in subtractive machined parts are results of problems with the machine, such as failing bearings, tool wear, chatter and breakage**, many of the faults in additive machining are based around material errors that can be traced back to printing parameters** [3]–[7], such as build-plate separation, delamination, and dimensional changes due to thermal distortion. These faults can be placed into two categories: tolerable faults and terminal faults. Tolerable faults are deviations in the printed part from the ideal model that do not render the artifact unusable; dimensional tolerances, surface profile, and material density fall into this category. Tolerable faults may need rework in order to be serviceable. Terminal faults are deviations that scrap the printed part such as delamination, positioning problems, and build-plate separation. Terminal faults are a source of inefficiency in the additive manufacturing process that result in loss of machine time and wasted materials. In-situ monitoring of artifacts during the manufacturing process is required to detect these faults in time for corrective action to be taken and turn terminal faults into tolerable ones.

Several of the faults present in FDM printing process, such as delamination, build-plate separation, and some dimensional changes, stem from the material changes that occur during the printing process. FDM printing works by heating a filament of material until it can be extruded into a specific position as the print head moves in space. The print head extrudes layer upon layer of material to build the artifact in the desired shape. During this process, the material goes through a heating and cooling cycle related to the print head depositing new layers of heated filament, as shown by [1]. [2] showed that the cooling profile during this process results in residual thermal stresses and strains, with tensile stresses on the top of the part and compressive stresses on the bottom. [3]–[5] showed that the resulting distortions can be reduced by decreasing layer height, decreasing heat input, preheating the build-plate, and insulating the part to control the cooling profile.

STATE OF THE ART

CONDITION BASED MAINTENANCE IN ADDITIVE MANUFACTURING

In general, attempts to collect information during the printing process can be classified by the method of data collection: either collecting information on the machine or collecting information on the printed artifact. Collecting information on the machine can be useful for determining the health of the machine and catching or predicting faults that stem from machine behavior, such as bearing wear or filament breakage. Condition based maintenance is the concept of using sensor feedback to detect developing faults in a system in order to plan and schedule maintenance only when needed and thereby minimize downtime and increase efficiency of use of machine components. This technique has been used extensively in the realm of subtractive machining to monitor the condition of tool wear, spindle and motor bearing wear, and part surface finish. Since additive manufacturing is a newer technology, CBM techniques have not been as extensively developed in this field. Research areas that have been explored are monitoring of bed and hot end temperatures, filament runout and breakage, and nozzle and extruder states. [8] used acoustic emissions to monitor FDM printers for extruder faults and classified extruder states using support vector machines. [9] also used acoustic emissions to monitor the extruder and classify the health of the extruder using machine learning algorithms. These studies were able to monitor different health conditions such as nozzle state and filament runout and breakage. [10] used two vibration sensors and machine learning algorithms to monitor the extruder state as well as printed artifact defects like warping. Several studies have instituted monitoring of the nozzle state through use of various sensors. [11] and [12] showed a correlation between nozzle state and current consumption of the extruder motor. This was monitoring method was implemented on a 3D printer and validated by experimental means. [13] utilized the force exerted from the extruder to the gantry as a means of monitoring the print head and nozzle condition.

[14] built a platform that is similar in idea to the work being presented here. They modified a low-cost polymer FDM printer to capture data from a selection of different sensor to work on developing a ‘smart’ 3D printer using closed loop feedback of the printer health state. The literature presents a platform that has been modified with the sensor suite but does not yet have closed loop feedback. This platform monitors position information by using rotary encoders on all axes, nozzle state through a thermocouple and rotary encoder on the extruder, and the current print layer through a USB web camera.

IN-SITU FEEDBACK IN 3D PRINTING

Counter to condition-based maintenance techniques, part monitoring is a method of detecting defects in the printed part by using sensors to observe the part instead of the machine. With 3D printing, these methods can be complex due to the free form nature of 3D printed parts. Research in this area has focused on a multitude of different sensor techniques, but especially machine vision applications. [15] used infrared imaging to monitor the surface temperature of the part. This information was used to derive temperature profiles for the current print layer and two layers below it, leading to information critical to interlayer bonding. [16] and [17] both used infrared imaging to capture spatial and temporal part temperatures. [18] used a camera mounted on the print head to classify delamination faults. Pictures from the camera were processed using a neural network to successfully detect these faults. [19] used a camera to gather pictures of the corners of the printed part and then classify the deformation state using a convolutional neural network. By extracting grayscale images of the part corners, this system was able to successfully detect warping in the printed part. [20] used a camera and image processing techniques using OpenCV to detect a number of part defects including material blobs and part detachment from the build plate. [21] used multifractal analysis of images taken with a reflex camera to detect artifact faults in metal parts made by powder bed fusion processes. [22] used ultrasonic inspection techniques to detect delamination in solid FDM parts during manufacture.

Several studies have used vision-based methods and machine learning techniques to monitor surface quality [23]–[27]. [28] and [29] explored using a single and double camera setup to compare the part profile to a profile generated from the part in a CAD program. [30] conducted a similar study but compared the pictures in a layer-wise manner. [31] utilized a multi-camera approach to monitor the entire part. This method used three pairs of cameras to create three reconstructions spaced 120 degrees apart. The reconstructions were compared with STL images of the part and was able to identify several printing errors including dimension errors, nozzle state issues, and incomplete prints.

In addition to image processing with cameras, several studies have also been undertaken using point cloud generation methods. [32] developed a framework for comparing point clouds to the STL file of a part. The STL file is used to generate a reference depth image which is compared to a depth image generated by converting a point cloud of the part. The part in this study was scanned with a laser line scanner mounted on the print head of the 3D printer. [33] proposed the use of the Fiedler number from Spectral Graph Theory as a measure of quantifying the relative quality of 3D printed surfaces. A laser line scanner was used to capture point clouds of the same part printed on two different platforms in two different materials. Use of the Fiedler number to capture differences between two parts was contrasted with statistical feature mining and facet examination techniques. [34] implemented a laser scanner on a consumer-grade polymer 3D printer and proposed a machine learning technique called self-organizing maps as a method of detecting defects using the generated point cloud. [35] and [36] utilized 3D digital image correlation to scan the part and provide feedback in near real time. [37] built a prototype machine for performing two image photogrammetry on a powder bed fusion 3D printer. [38] used photogrammetric techniques with a six-camera array to detect artifact defects after completion of the printing process. [39] also used a similar camera setup to compare printed part profiles to the same profile generated in a CAD program as a means of detecting malicious cyber attacks to a 3D printed part. [40] wrote a review of in-situ monitoring methods for fused filament fabrication.

USE OF PHOTOGRAMMETRY FOR MEASURING SMALL OBJECTS

Photogrammetry is the process of using several two-dimensional photographs to create a three-dimensional model by correlating common points in the photographs and using triangulation to determine the relative position of these points in space. This technique has long been established and has been used successfully and extensively in fields such as terrain mapping, historic object preservation, and metrology. [7], [41]–[43] showed that photogrammetry techniques are currently able to reconstruct features smaller than 1 millimeter. These studies showed that measurement accuracy better than 10 microns is possible with the correct camera setup. [44] used a laser speckle pattern to increase the resolution and reduce error in a photogrammetric model. This was accomplished using additively manufactured parts in several materials. Photogrammetric techniques have even been used to create real time strain measuring systems as demonstrated by [6].

[45] demonstrated the use of a single image photogrammetry technique for estimating the position of a partially completed 3D printed object on the build plate. A single image of the part was compared with known camera pose information to locate the part in the machine coordinates. [46] investigated the use of structured light and single

camera photogrammetry to assess 3D printed part quality after printing. They proposed a series of indicators for quantifying the quality of the point cloud generated by this technique. [47] wrote a review of optical measurement technologies in relation to metrology of additively manufactured parts. This review addressed the challenges and benefits of shear interferometry, time-of-flight sensors, close-range photogrammetry, and several types of profilometry.

KINEMATIC COUPLING OF THE PRINT BED

Kinematic couplings have been around since at least 1876 [48] and work by providing an exactly constrained interface between two parts. In three-dimensional space, every body has six degrees of freedom. To precisely locate two parts relative to each other, there must be exactly one constraint controlling each of those six degrees of freedom. This concept is often used in optical applications to create precision mounts for optical components. One of the common forms of the kinematic coupling are three balls mated to three vee grooves. Each ball has two tangential points of contact with the vee groove, one on each face. Aligning the groves at non-parallel angles to each other creates a condition where the three balls can only occupy one configuration in space if their spacing with respect to each other is held constant (as if they are all part of a single, rigid body). An illustration of this concept can be seen in Figure 1.1. Since the relationship between the two bodies is exactly constrained, the interface can be separated and reassembled in a very repeatable manner. A study by [49] showed a 355 mm (14 in) diameter coupling with a repeatability of ±0.25 micron.

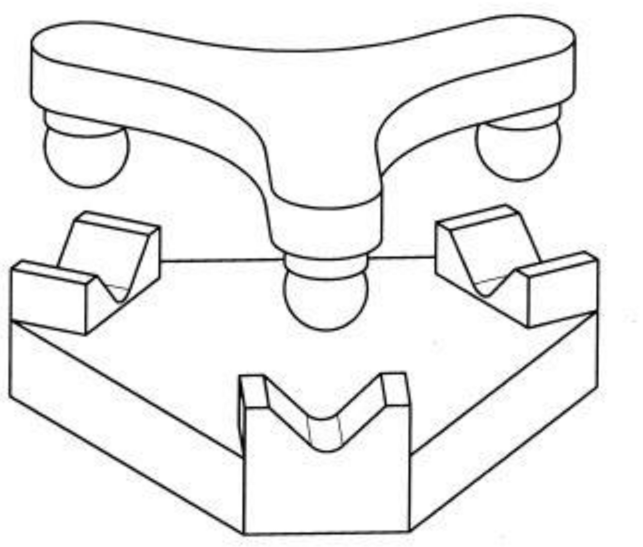


Figure 1.1 Example of kinematic coupling from 3 balls mated with 3 vee blocks. [48]

Markforged holds a patent for using a kinematic coupling to mate the print bed of a 3D printer to a moveable stage to allow the build platform to be removed [Patent US9539762B2]. This feature is built into all current Markforged printers [50]. Kinematic couplings have also been used for the bed of the open-source Jubilee printer [51] to provide an automated three point leveling mechanism, and on the build platform of the Fuselab FL300 printer [52].

PROBLEM DEFINITION

This work will focus on the design and construction of a polymer 3D printer in the fused deposition modeling style. The intended use case for this printer will be as a test bed for future experimentation of both printed objects and printer control and monitoring algorithms where knowledge of the machine's parameters during the printing process are tightly controlled, monitored, and recorded. This will allow qualification of the printing process and provide a method to rule out process anomalies during experimentation. Additionally, the printer has been designed with a mechanism to allow for *in-situ* photogrammetric models to be constructed during the printing process, which will provide a geometric build log of the printing process.

PROBLEM STATEMENT

Design and fabricate a polymer FDM additive manufacturing platform that can print "normal" 3D printed polymers such as ABS and PLA, and will be able to print high temperature polymers like PEEK and Ultem with a minimum amount of modification, is capable of monitoring and controlling relevant build parameters, can capture in-situ geometric data of the printed part through photographs that can be later used for photogrammetric reconstruction, and is flexible and open for modification to meet future research needs.

REQUIREMENTS

The above problem statement gives several guiding principles for this design. First, the product must be a device capable of correctly additively manufacturing an object from polymer materials using FDM methods. Important parameters of the printing process, including print head velocity, position, and temperature, bed temperature, and environment temperature and humidity, should be capable of being monitored, recorded, and controlled. The device should be designed to eliminate as many variables as possible that will not be directly monitored or controlled. As the printer will serve as a test bed for many future experiments, its utility and capabilities should be maximized, and flexibility for future changes should be considered.

The design of a 3D printer requires that the position of the print head be known at all times, and for this machine the desire is that the position be within ±50 microns of the commanded position. Therefore, the machine elements must be designed with this in mind. The rigidity of the machine frame must be sufficient to limit deflection due to load and inertial forces, and the motion elements must be capable of sufficient resolution.

To maximize the utility of the printer, it is desirable to be able to print items in a range of sizes, in different materials, and at a variety of different print speeds. Thus, the printer should be designed to print not only the common 3D printed polymer such as PLA and ABS, but also engineering polymers such as Ultem and PEEK. This will require that the print head be capable of reaching temperatures in excess of 350 C and bed temperatures of at least 120 C. The ability to heat the environment is also needed. A reasonably large print volume is desired, preferably greater than 200 mm (8 in.) in all dimensions. The printer should be able to achieve print speeds of 200 mm/s (8 in/s), and since the key to being able to achieve high print speeds is high acceleration, the printer should be capable of 1000 mm/s^2 (39.4 in/s^2) in the X and Y directions.

ADDITIONAL CONSTRAINTS

There were several real-world limitations for this design. The budget criteria for the project was $2000, and this includes resources for performing photogrammetry. Several parts had been purchased for a separate project but were not used and incorporating those parts into this project to save lab funds was desired. These parts included aluminum extrusion and 5 NEMA 17 stepper motors with a 12VDC 30A power supply. As a final constraint, the funding for this project had a spending deadline so selection and purchase of many of the major components had to be completed within a few months of the start of the project.

THESIS OUTLINE

The remainder of this work will focus on the solution to the problem presented above. In chapter 2, the 3D printer test bed developed for this solution will be introduced. An overview of the platform will be followed by specifics of mechanical design including motion architecture, subsystem specific requirements, component selection and design, and the introduction of a mechanism for allowing single camera photogrammetry to obtain 360-degree coverage of the printed part. Chapter 3 will discuss testing and validation of mechanical subsystem requirements. The electronics and motion control system will be introduced in Chapter 4 and include an outline of the software toolchain supporting the hardware, selection of the sensor suite, and the hardware and software performing the photogrammetry data capture. Chapter 5 will conclude the work with a discussion of future improvements, overall requirement satisfaction, and lessons learned.

CHAPTER TWO

3D PRINTING TEST BED

OVERVIEW

Pictures of the designed system can be seen below in Figure 2.1, including both a render of the system as envisioned in CAD and a picture of the physical system after construction. The printer uses a belt driven CoreXY architecture for movement in the XY plane and a cantilevered bed design motivated by a leadscrew for Z movement. The XY axes are equipped with 12mm linear rails, and the Z axis is equipped with a single 20 mm (0.787 in) linear rail. Magnetic linear position sensors with a resolution of 1.44 microns (0.000057 in) will be used for closed loop position control. The printer volume is enclosed and insulated to provide better control of environmental variables and is equipped with temperature and humidity sensors. The print head will be capable of reaching temperatures of 450 Celsius and the bed is capable of reaching 250 Celsius.

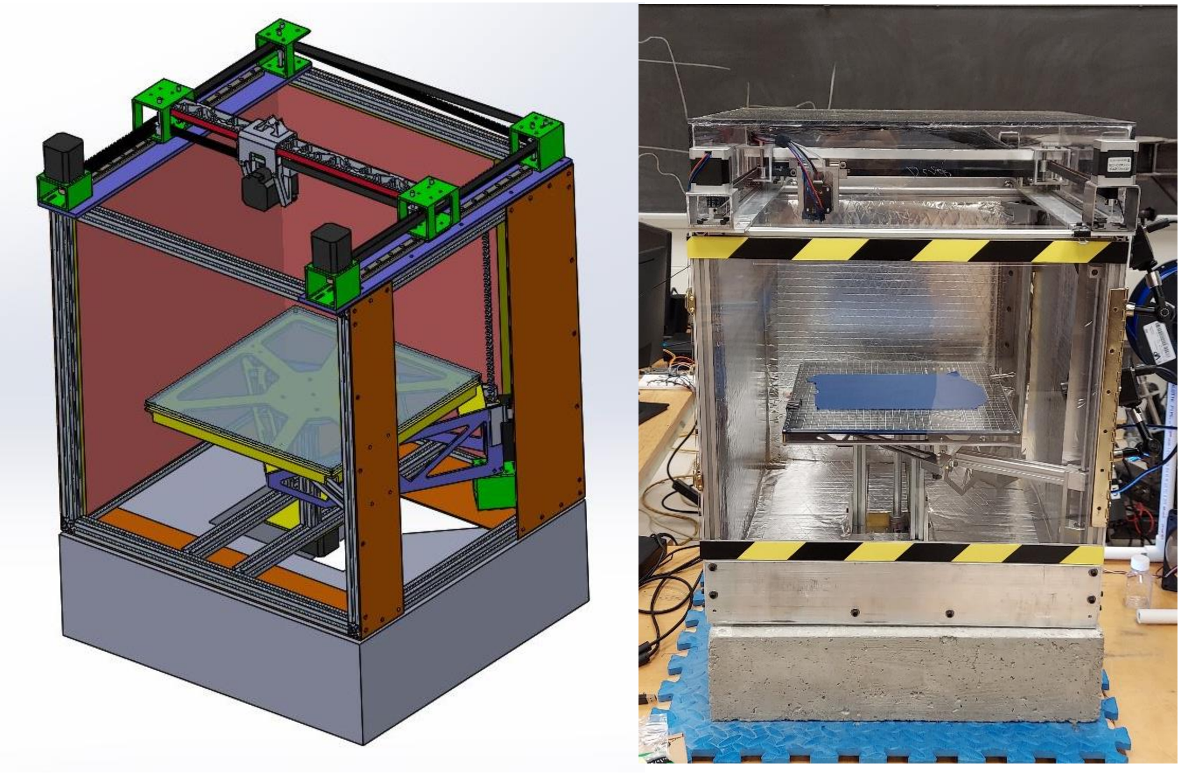


Figure 2.1 (Right) CAD rendering of 3D printer and (Left) physical implementation of that CAD model

The printer is controlled using a Windows PC connected to a Dynomotion KFLOP 8 channel CNC motion controller. Monitoring and recording of process variables is accomplished using a Raspberry Pi 3. The photogrammetry step runs on a Nvidia Jetson TX2 that takes pictures using two 2 MP USB cameras from Arducam.

The platform is capable of monitoring the print using single camera photogrammetry because it has been designed to spin the print bed. Whenever a photogrammetric model is desired, a custom M code is issued to the KFLOP controller, which then lowers the bed to the bottom of the Z axis. A kinematic mount is designed into the cantilevered Z axis and allows the bed to detach from the cantilever frame and be replaced with a very high degree of repeatability. When detached, the bed is rotated using a stepper motor while the Jetson TX2 takes of the entire periphery of the 3D print. Once this process is complete, the Jetson uses the opensource AliceVision framework to produce a dense point cloud of the 3D print.

The end result of this exercise is a machine that can function as a test bed for future experimentation and study in polymer FDM 3D printing. It is able to print all current 3D printable polymers, has a large span of bed and environmental temperatures, and very little undirected airflow. Most build parameters are captured for later study and

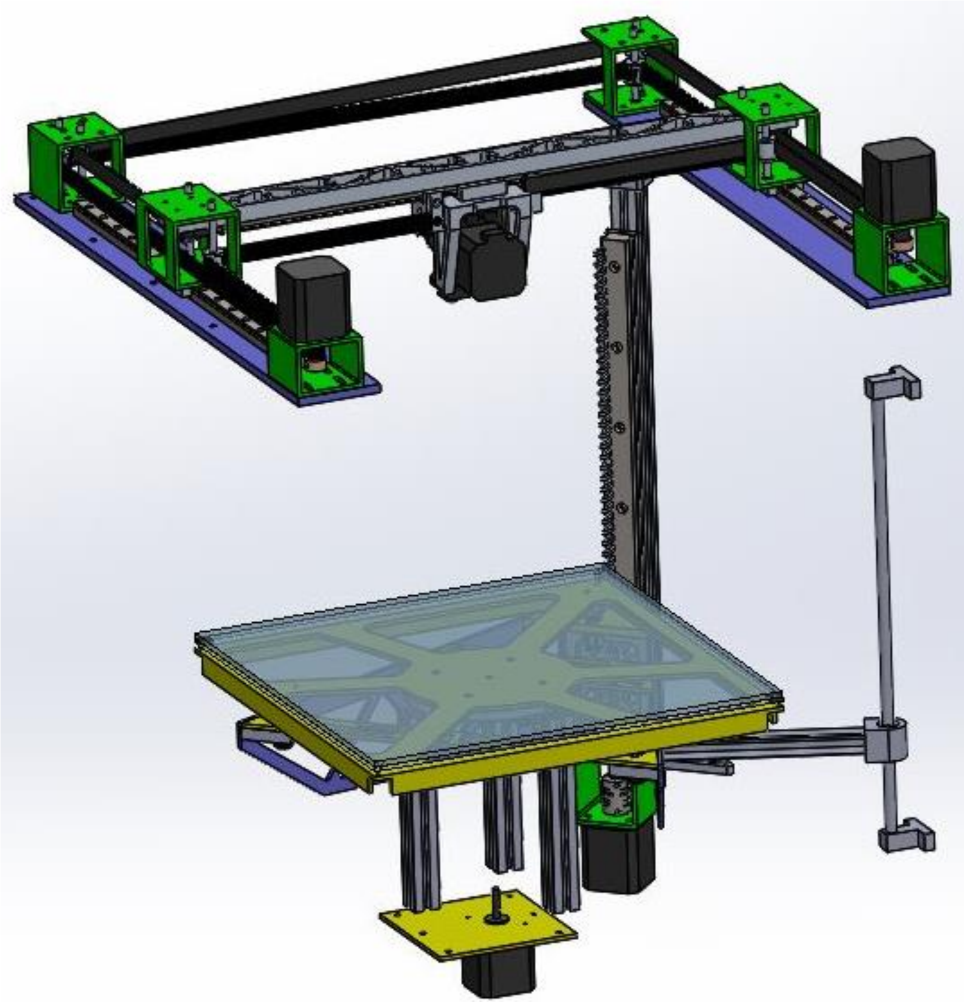


Figure 2.2 CAD rendering of motion components and their relative alignments.

verification of the printing process. Additionally, it is capable of recording a periodic geometric history of the part as it being printed using the photogrammetry sensor and toolchain. A CAD rendering of all the motion components can be seen in Figure 2.2.

The printer was designed as a series of sub-assemblies, each with its own requirements. The rest of this chapter provides a more detailed look at these sub-assemblies and their design choices. Frame and enclosure design will be presented first, followed by the XY motion system, and the Z motion system will conclude the chapter. While requirements for each sub-assembly are presented here, testing was done on the completed system instead of testing the individual sub-assemblies. Therefore, satisfaction of the requirements will be presented in Chapter 3.

FRAME AND ENCLOSURE DESIGN

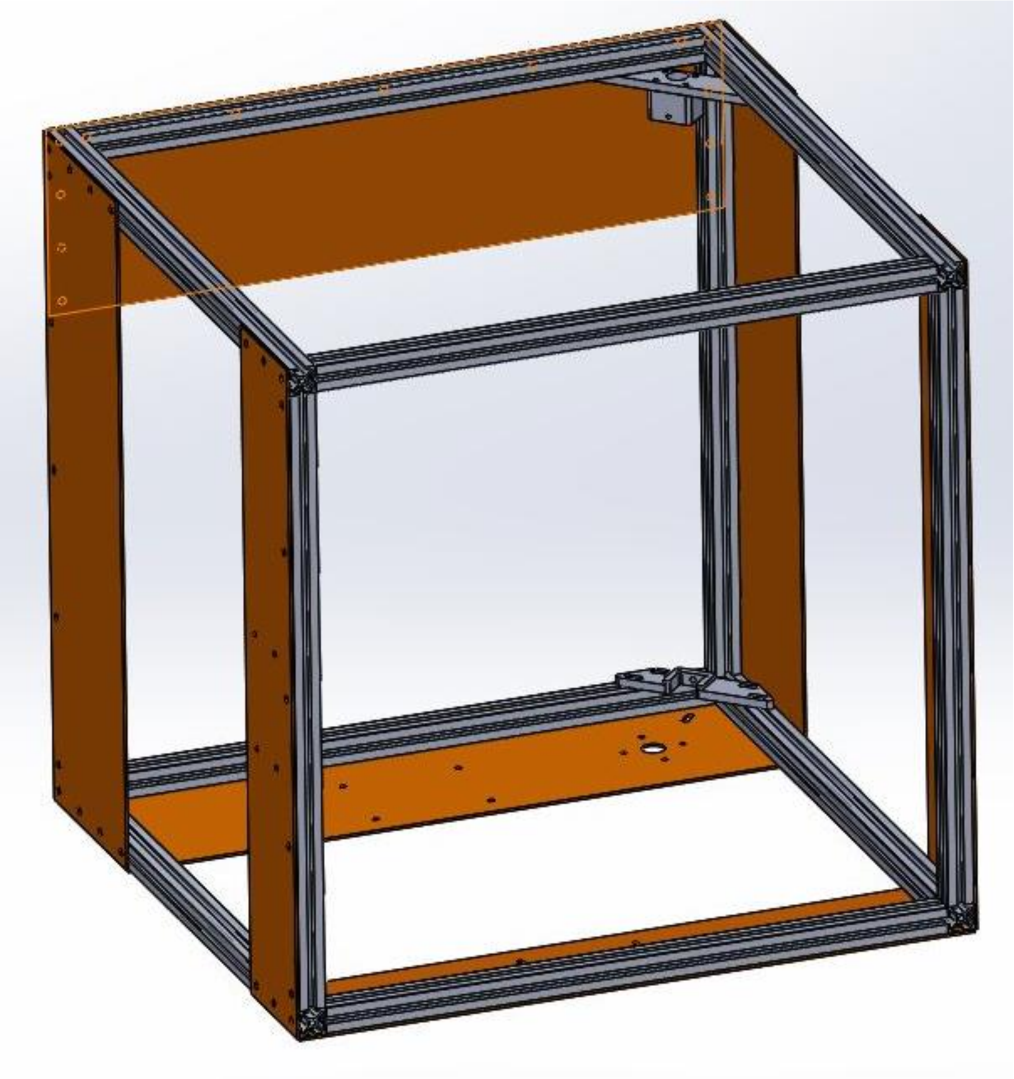


Figure 2.3 3D printer frame consisting of aluminum extrusion and aluminum plates.

RELEVANT REQUIREMENTS

An ideal frame is stiff enough to produce negligible deformation under working loads and has a high enough dynamic frequency to prevent ‘ringing’ artifacts in the printed part. Also, the machine should have enough mass to prevent machine movement during operation. For this project, it was decided that ‘stiff enough’ means that frame deformation should not be distinguishable in normal motion. The desired positional accuracy for this machine is ± 50 microns (0.002 in), so frame stiffness is required to be high enough that the frame deforms less than 20 microns (0.0008 in) under maximum operating loads. This requirement was extended to the gantry bar portion of the XY motion system. Deformation from inertial forces can result from the gantry, the frame, and the extruder mount so taking a tolerance stacking approach is necessary to ensure that the end result is within the system requirement.

Setting a target value for the dynamic stiffness of the frame was difficult as little research has been done to investigate the dynamic characteristics of 3D printers. This is in stark contrast to subtractive machining technologies where significant research has been implemented to determine target values for the dynamic stiffness of machining centers needed to avoid tool chatter and produce desired surface finishes. Therefore, a target value was not assigned to this requirement; it is only required that ringing not be observable in the printed part under the highest operational speed the printer is capable of.

DESIGN ATTRIBUTES

The printer frame was designed as cube constructed of aluminum extrusion of profile dimensions 20 mm by 20 mm. This profile was selected because it was available from previous project. To stiffen the cube, aluminum plates 3.175 mm (0.125 in) thick were used to create a closed construction. Plate placement and orientation can be seen in Figure 2.3. To keep the dynamic response of the frame high, the frame was intentionally kept relatively light. However, to add enough mass to the machine to keep it from moving during high-speed movement of the gantry, a concrete base weighing 105 pounds was constructed and attached to the bottom of the frame. Two aluminum plates were used on the bottom of the cube to provide attachment points for the concrete base, the Z motor, and support for the bed rotation mechanism. Figure 2.4 shows the concrete base and the aluminum extrusion profile. Additionally, the front was left open to provide room for a door to access the interior of the printer during normal use, and the top was left open for the construction of the XY motion assembly. The XY motion assembly was designed to provide additional stiffness by attaching to perpendicular elements of the frame.

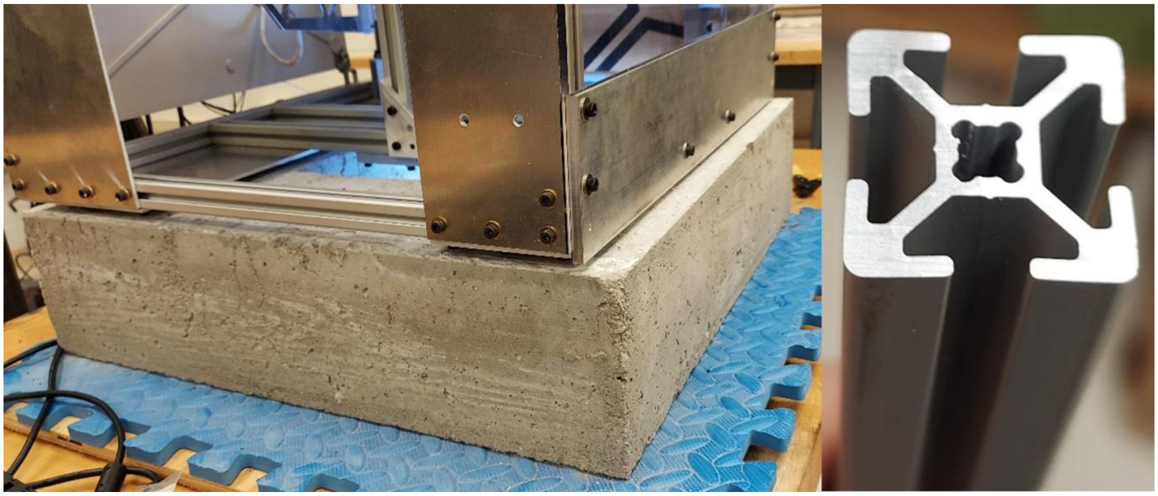


Figure 2.4 (Right) The concrete base used to anchor the 3D printer. This base weighs 105 pounds. (Left) The 20mm x 20mm aluminum extrusion profile used to construct the printer frame.

XY MOTION SYSTEM

RELEVANT REQUIREMENTS

The motion system is responsible for correctly positioning the extruder in space. The normal nominal nozzle diameter is 0.4 mm (0.01575 in). It is desired that the printer has an XY resolution that is less than 5 percent of this distance and for bi-directional repeatability to also be within 5 percent of this distance. This translates to a target value of 20 microns (0.0008 in) for resolution and bi-directional repeatability. These minimum movements also impact the desired stiffness of motion components. Specifically, the gantry must not deflect significantly under operational loads and the target value for this is equal to the repeatability of the motion system, 20 microns (0.0008 in). Stiff, modern printer designs, such as the Voron 2, are capable of printing at high speeds: 250 mm/s (10 in/s). The test bed needs to be able to print at similar speeds so that it can be used for investigations requiring these speeds. The desired target for this printer is 200 mm/s (8 in/s). All parts within the build environment also must be able to operate in the maximum enclosure temperature of 100 Celsius.

In addition to these requirements, there are several design goals for the test bed. It is desired that speed, acceleration, precision, and build area be maximized. This will help make the test bed as useful towards a broad range of future investigations. Deflection or deformation of any component that affects the true position of the print head should be minimized. This is imperative for the accuracy of the test bed. Considering these design goals gives an important heuristic that was used during the build process: minimize the 'flying weight' or the amount of mass that must be moved in the XY axes. In an FDM printer system, inertial loads due to system movement are expected to be largest loads experienced by the system since there are no tool forces or outside resistances to deal with. Reducing moving mass will aid in both maximizing acceleration and minimizing deflection.

COREXY ATTRIBUTES

The CoreXY architecture was chosen for the test bed. This architecture moves the print head in the XY axes and moves the bed in Z. Motivation is implemented by using stationary rotary motors to drive the print head via two belts. An illustration of the CoreXY architecture can be seen in Figure 2.6 The CoreXY implementation on this 3D printing platform.. Note that both belts connect to the print head. When both motors turn at the same rate in the same direction, the print head is driven in the X direction and the gantry does not move. When both motors spin at the same rate but in opposite directions, the gantry is driven in Y and the print head does not move. When just the right motor turns, resulting motion is at a 45-degree angle in the positive Y and negative X directions. When just the left motor turns, resulting motion is at a 45-degree angle in the positive Y and positive X directions. The implementation of the CoreXY architecture on this platform can be seen in Figure 2.6.

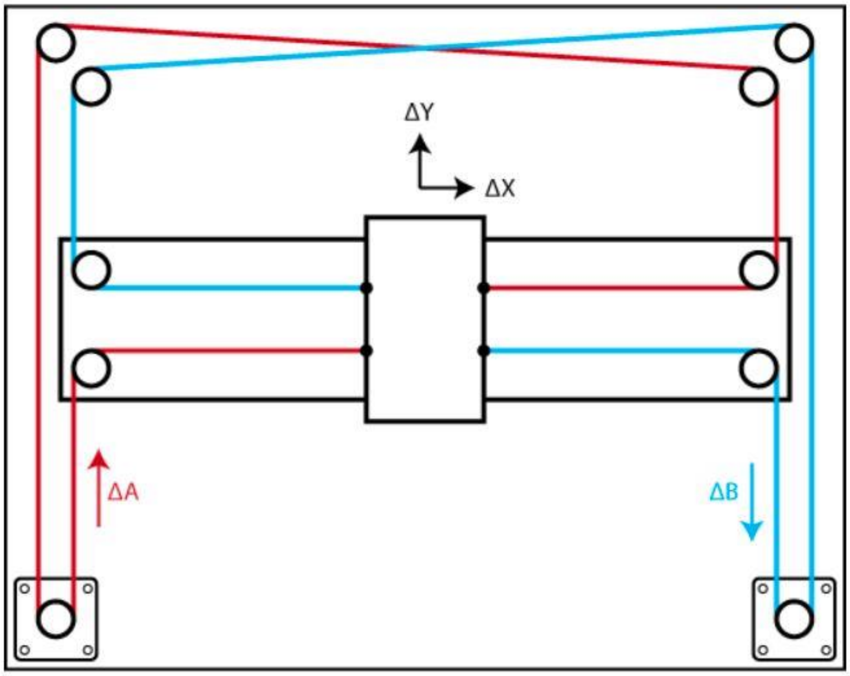


Figure 2.5 An illustration of the CoreXY motion architecture. [53]

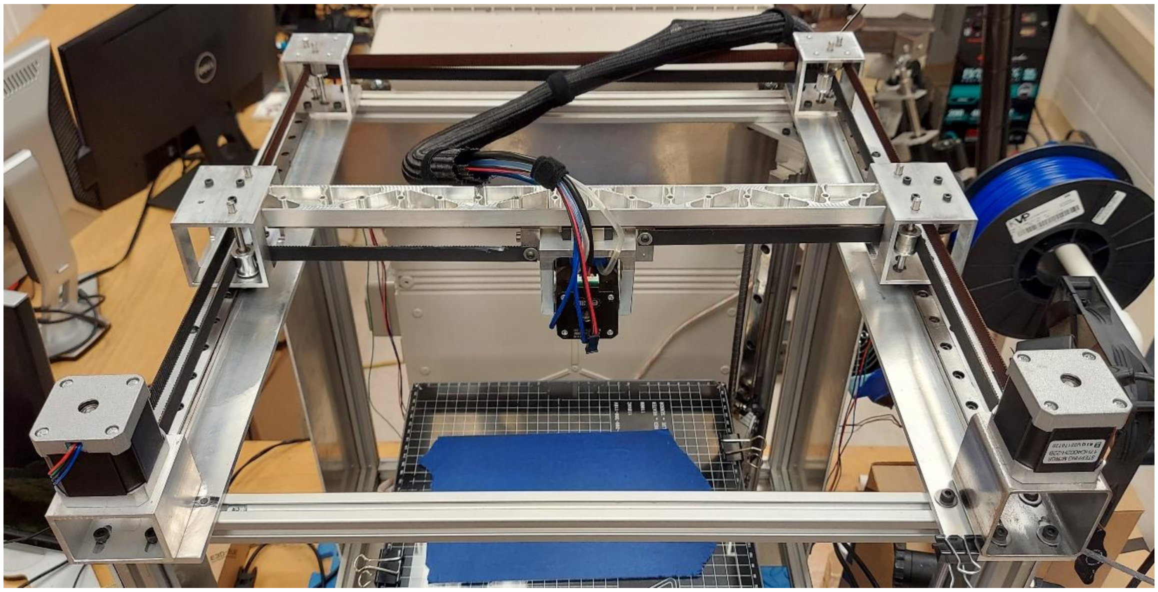


Figure 2.6 The CoreXY implementation on this 3D printing platform.

The governing equations for equating motor rotation to XY movement are shown below [53]. A and B are the left and right motors used for XY motion. These labels can be seen in Figure 2.5.

$$\Delta X = \frac{1}{2}(\Delta A + \Delta B) \tag{1}$$

$$\Delta Y = \frac{1}{2}(\Delta A - \Delta B) \tag{2}$$

$$\Delta A = \Delta X + \Delta Y \tag{3}$$

$$\Delta B = \Delta X - \Delta Y \tag{4}$$

CoreXY has several design characteristics that are beneficial for our design goals. First, by using stationary motors the ‘flying weight’ or the amount of mass that the motors have to motivate is reduced since the motors do not move themselves. This means that for a given size of motor, the system can be accelerated more quickly. Since the motor size for this project was predetermined (see Chapter 1), this maximizes the acceleration of our system with respect to the motor constraint. Additionally, this architecture allows the motors to be located outside of the build chamber, meaning that they will not need to survive prolonged operation at elevated temperatures. This can be seen in Figure 2.7.

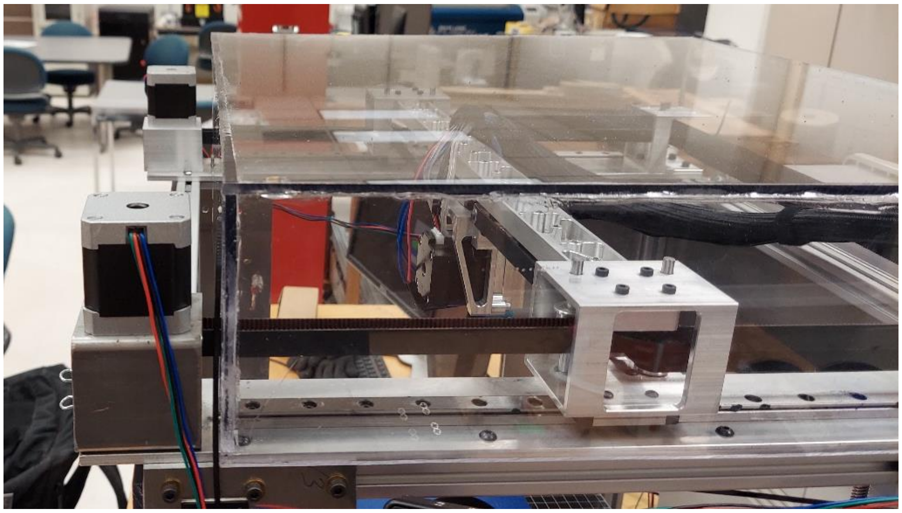


Figure 2.7 The CoreXY architecture allows the XY stepper motors to be excluded from the build environment.

MECHANICAL COMPONENT DECISIONS

The major design decisions for the XY motion system components were the choice of linear movement guide, print head, and belts, and the design of the gantry and print head mount. Recall that the motors were previously selected. The print head was selected first because several of its properties such as mass, position, and mating interface would strongly influence other design decisions. The desire to remove errant airflow from around the part and the requirement for the heated build chamber eliminate the majority of print heads on the market because the cold end is often air cooled by a fan that is attached to the print head. The airflow from this fan can reach and affect the part even though its purpose is not to cool the part. Another source of airflow is usually provided specifically for cooling the part and is controlled depending on the desired part cooling. It is imperative to remove the possibility of the uncontrolled variable this represents. Additionally, trying to cool the cold end with hot air from the printer environment is not only ineffective, but now makes the temperature profile within the print head a function of build environment temperature. Again, separation of controllable variables is imperative to the research aspect of this platform. The print head chosen for the test bed is the E3D Titan Aqua, a water-cooled, direct-drive print head. This system uses water cooling to keep both the cold end and the extruder motor cool, allowing sustained use in a heated environment. The print head is connected to a pump, a reservoir, and a radiator equipped with a cooling fan by silicone tubing. However, this extruder is only rated for sustained environmental temperatures of 80 Celsius which does not meet the temperature requirement for components within the build environment, but it was chosen because it is the best off the shelf option. In order to provide an extruder capable of operating in the 100 Celsius environment, an analysis of the extruder was performed. This revealed that most components are metal except for the large gear used to drive the hob and the filament guide. While this has not been completed yet, these parts can be replaced with metal versions much more conveniently than designing and constructing a custom extruder. Information on both parts is available from E3D as technical drawings and models are available on GrabCAD. If other factors crop up, such as having the extruder motor overheat, an additional water-cooled heat sink can be developed for the backside of the motor. Upgrading the commercial product will require less effort than designing a custom component. The extruder and mount can be seen in Figure 2.8.

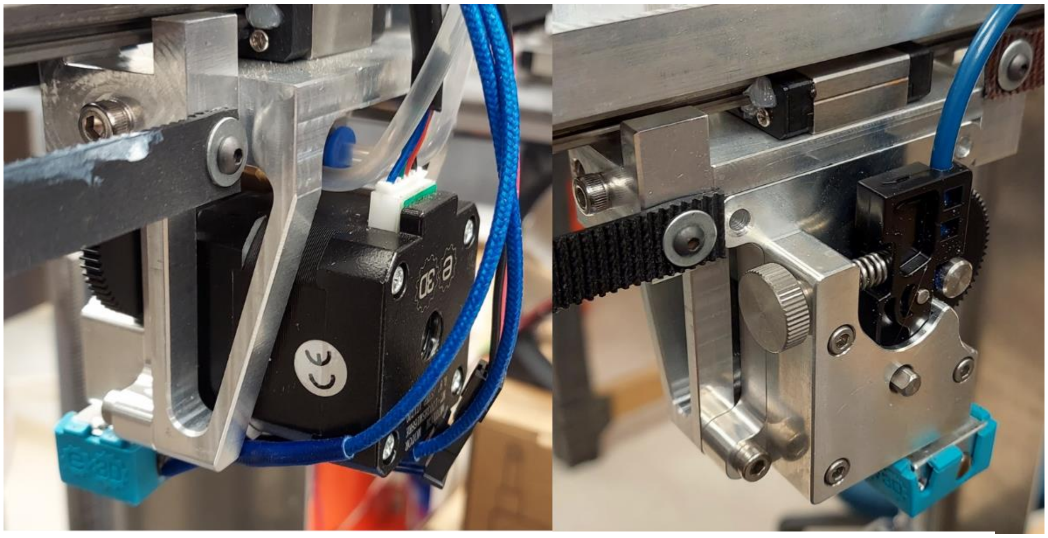


Figure 2.8 Pictures detailing the E3D Titan Aqua print head and custom mount.

Linear rails were chosen as the guides for linear motion. While linear rods were also considered, linear rails were judged as the superior option for several reasons. First, linear rails can be supported along their entire length, while linear rods cannot. This means that the stiffness of the rail sections can be tied to another structure, i.e. whatever they are bolted to, which makes is far easier to tailor the deformation characteristics of the motion system while minimizing weight. Additionally, many 3D printing systems attempt to save cost on linear motion by using sliding elements on linear rod instead of rolling elements. This reduces the precision of the system as there is now a greater element of static friction to overcome before movement takes place. Since the desired resolution and repeatability for this system are in the micron range, any design choice that reduces system precision should be avoided.

Belts are the normal choice of force transfer element in the CoreXY architecture for 3D printers. The use of toothed timing belt allows for force transfer without slip, ensuring that the position of the print head is not indeterminate. While many 3D printers use 6 mm (0.236 in) wide 2GT profile belts, 12 mm (0.472 in) wide high temperature 2GT belts were chosen for the test bed. The wider belt provides better system stiffness compared to the smaller belts, and the high temperature version is rated for up to 135 Celsius versus 85 Celsius for the normal version.

GANTRY DESIGN

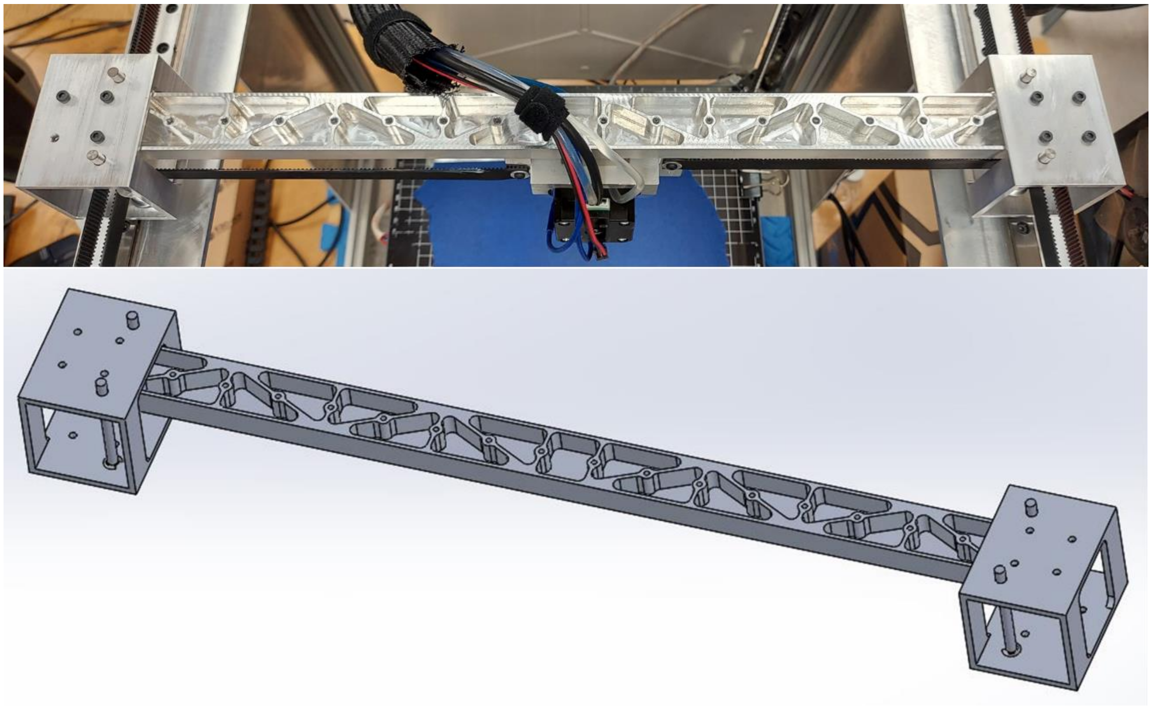


Figure 2.9 (Top) Gantry as manufactured and (Bottom) CAD model of gantry assembly.

A CAD rendering and a picture of the built gantry system can be seen in Figure 2.9. Once the decision to use linear rails and the Titan Aqua extruder was made, designing a gantry to meet the specified requirements could begin. The typical arrangement for the CoreXY system is to have two rails outside of the build area with a gantry that spans between them. With a preloaded 12 mm linear rail, the stiffness of a single rail and carriage setup is sufficient to keep the deflection of the nozzle relative to the gantry within the specified limits if the rail does not deflect. FEA was performed in ANSYS to determine if the unsupported rail would be sufficient to meet the deflection requirements or if support was needed. The analysis showed that a single rail would not meet the deflection requirement and a bar support was designed and optimized in ANSYS. The final design is shown in Figure 2.9 (Top) Gantry as manufactured and (Bottom) CAD model of gantry assembly. The FEA for the final part can be found in Chapter 3. It should be noted that this particular part could likely be optimized further. A minimum wall thickness was placed on the optimization problem to make manufacturing easier and to avoid any problems with milling thin wall elements. The wall and floor thickness can be reduced beyond what was produced here, as can the corner radii. A 6.35 mm (0.25 in) end mill (the smallest available at the time) was used to make the interior pockets of this part and this dimension limited the size of interior corner radii that could be manufactured. Figure 2.10 shows a picture of the gantry bar being manufactured on a small CNC mill.

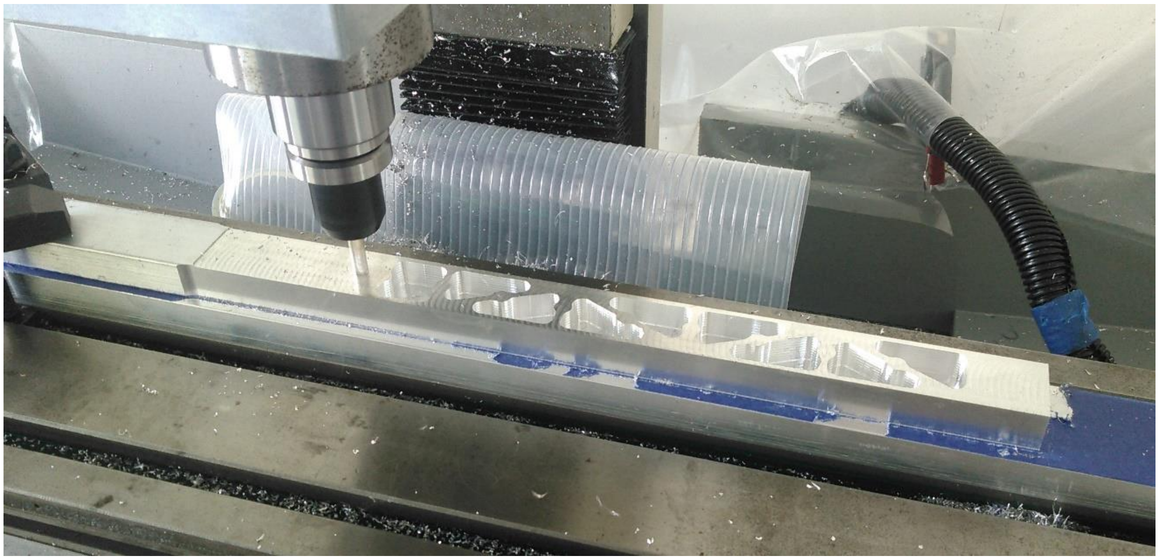


Figure 2.10 Gantry bar being milled on a small CNC using a 0.25 inch carbide endmill.

To reduce deflection, the gantry was mounted on two riser housings situated at the end of the gantry. These housings connected the Y axis carriage blocks to the gantry and provided a mounting point for the belt pulleys that redirect the belt along the X axis. Doing this placed the center of gravity for the gantry and extruder assembly in between the Y axis rails, minimizing the moment about the X axis and allowing for the use of a single rail carriage on each end of the gantry instead of two carriages.

One important aspect of any motion system is orthogonality of the motion axes. Ease of tramming, or the process of aligning the motion axes to ensure orthogonality, must be considered during the build stage. For the CoreXY architecture, it is imperative that the belt paths that are routed to the pulleys on the gantry are parallel to the linear rail. Failure to keep these exactly align will result in movement error. To accomplish this, the Y stage elements, including motor mounts, linear rail, and end pulley mount, were split into two groups, with each group being a mirror image of the other. Each set was mounted to an aluminum plate, and this plate was later mounted to the frame. This allowed for the linear rails to be mounted and aligned with their respective motor mounts and fixed pulley mounts without considering parallelism to the other Y axis group. Then one group was mounted and aligned to the frame. The second group was mounted and aligned parallel to the first linear rail using a dial test gauge. Then the gantry assembly was mounted to both Y axis rails and squared up using a dial test gauge and a machinist square. Once the X and Y axes were aligned, the Z sub-assembly could easily be trammed. Figure 2.11 shows the assembly used to keep the relative alignment of the belt paths to the linear rails, and how those plates are aligned on the frame. Note that the belt paths that go to the gantry pulleys are parallel to the linear rails.

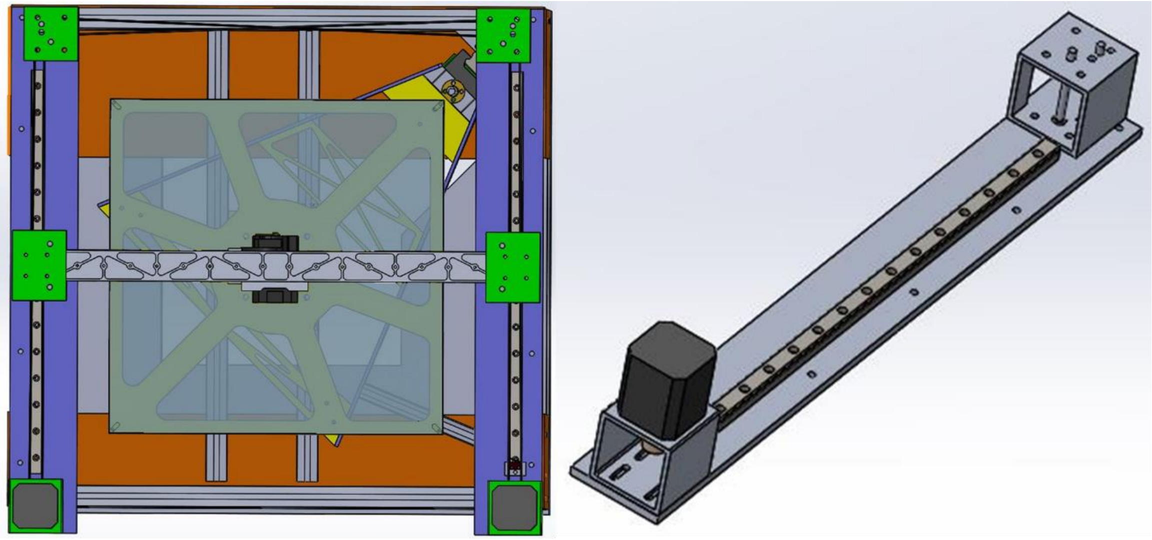


Figure 2.11 (Left) Alignment of both top plate assemblies on the printer frame and (Right) an assembly consisting of motor mount, pulley mount, and linear rail. This assembly keeps the relative alignment of the belt path constant during machine setup.

Z MOTION SYSTEM

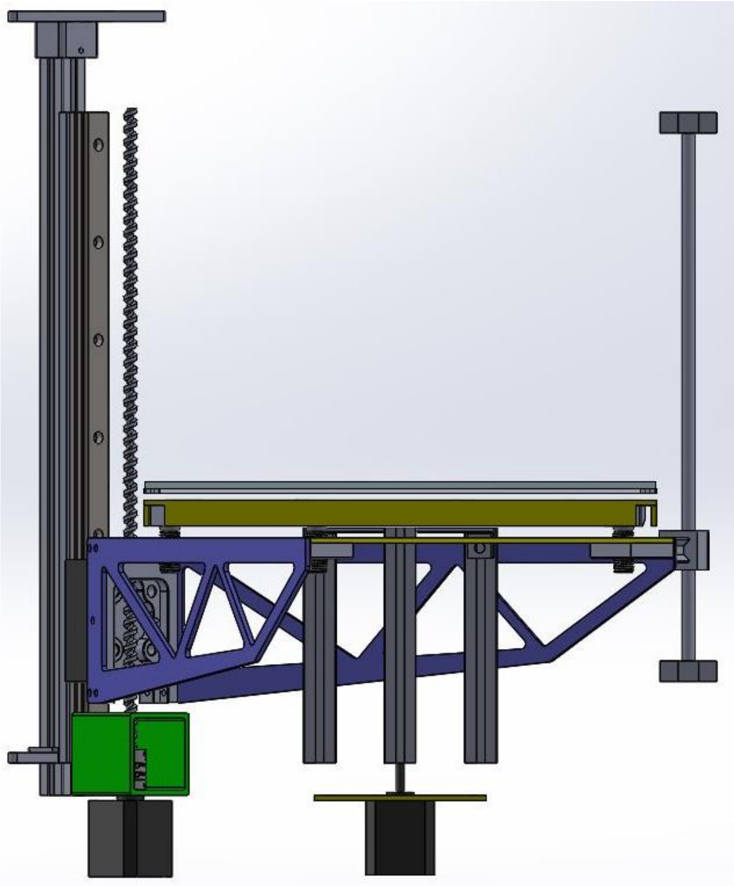


Figure 2.12 Elements of the Z axis including motion elements, build platform, and bed rotation mechanism.

RELEVANT REQUIREMENTS

The most important aspect of the Z motion system is its minimum repeatability. This dictates how small the layers can be and how fine a print the system can produce. The minimum desired layer height for this system is 25% of the nozzle diameter, or 0.1mm (0.004 inches). Thus, the minimum repeatability of this axis should be less than 10% of this value but this repeatability only needs to be in the direction of the print bed moving away toward the print head. This is the critical dimension for FDM Z motion. This may seem counterintuitive, but since gravity is pulling the Z carriage downwards, moving upwards always means the Z carriage is loaded against the actuator screw. The slicer chosen for this project has a setting to ensure that every Z motion ends with an upward motion, and this is discussed in Chapter 4. Z motion needs to be linear and able to be adjusted to be orthogonal to the XY plane defined by print head movement. The print bed needs to be heated, and able to reach a temperature that will support printing with high temperature polymers like PEEK and Ultem. The print bed also needs to be able to be leveled to be parallel to the XY plane defined by print head movement. For safety, it is desired that the print bed will remain at its current position if power is removed from the Z actuator.

The Z axis has a few unique requirements to support the desire for the printer to capture in-situ geometric data about the printed artifact. Photogrammetry is the chosen method for capturing this geometric data and uses several 2D photographs of an object taken from various angles to reconstruct a 3D representation of the object. To produce the best reconstruction, each camera used for this purpose needs to be calibrated for any distortion present in the image sensor, lens structure, etc. This calibration needs to happen every time the environmental parameters affecting the camera or any object that light passes through on its way to the camera changes. Since the camera will have to be inside the build chamber, or light will have to pass through a window in the build chamber to reach the camera if the camera is outside, this means that calibration is required for every print. Thus, minimizing the number of cameras is desirable because it reduces the amount of setup work that must be done to use the machine.

This presents an interesting dilemma. Pictures need to be taken of the printed artifact during the build process from all angles to perform photogrammetry, but it is best to use the fewest number of cameras possible to accomplish this task. The two easiest ways to accomplish this would be to either move the camera around the stationary object, or to keep the camera stationary and spin the object. Both options were considered, and the following requirements were developed: the method of capturing pictures cannot adversely affect the printing process and must allow for pictures to be taken from any relative rotation of camera and object about the Z direction.

MECHANICAL COMPONENT DECISIONS

The print bed is built in typical polymer FDM style: a borosilicate build surface supported by an aluminum plate with bonded resistive heater pad. The bed is 330 mm x 330 mm of which 300 mm x 300 mm (11.8 in x 11.8 in) is directly over the heating coils in the heating pad. The heating pad is a 1200 W heater that runs off 120 VAC and can reach peak temperatures of 250 Celsius and sustained temperatures of 200 Celsius. The heated bed and its borosilicate build surface are made by E3D.

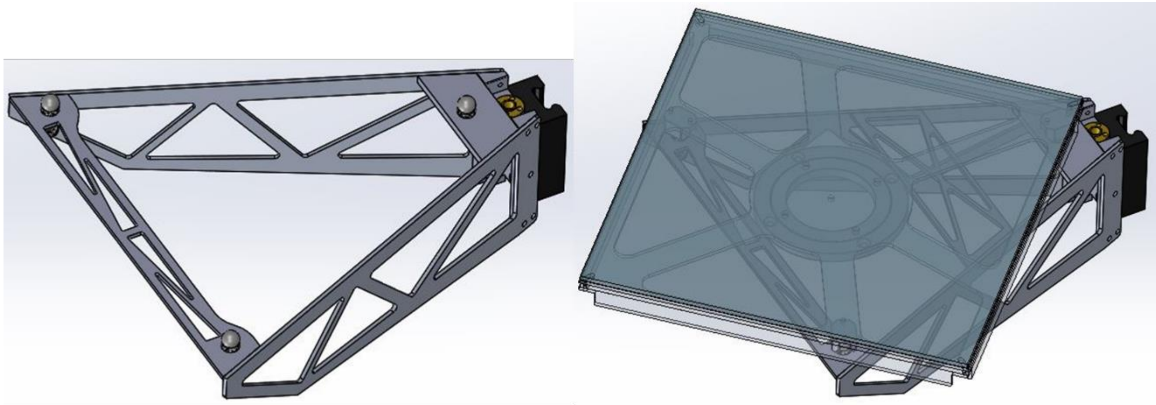


Figure 2.13 (Left) Bed support structure showing the leveling screws with mount for the kinematic coupling. (Right) Build plate mounted on the Z support structure.

The bed is supported by a kinematic coupling consisting of three ball bearings mated to three v-blocks. This coupling is discussed further in the next section. The ball bearings are mounted on threaded shafts to provide a bed leveling mechanism, and these shafts are connected to the support frame of the Z carriage. Figure 2.13 shows the print bed and Z carriage in detail. The carriage rides on a 20 mm (0.787 in) linear rail chosen for its ability to support the moment generated by the cantilevered print bed design. The entire carriage is motivated by an 8 mm (0.315 in) pitch acme leadscrew. The leadscrew is not preloaded, but the weight of the Z axis ensures that there is no backlash during normal movement of the printer. Normally positional accuracy is only required of the Z axis when the axis is stationary, not during movement. If the printer were to be operated in a manner that necessitated positional accuracy while the Z axis was moving, this design choice would need to be re-visited. The leadscrew is powered by the same model of NEMA 17 stepper motor that powers the X and Y axes.

BED ROTATION MECHANISM

The v-blocks are mounted to the underside of the bed structure and are oriented so that the axis for each block parallel to the vee is pointed towards the middle of the print bed. This setup provides two things: thermal growth of the build plate does not change the position of the center of the build plate, and a mating interface that allows for the bed to be removed from its support structure and then replaced with extreme repeatability. Thermal growth is an important consideration since the bed is designed to heat up over 180 Celsius from room temperature and to be able to cycle through this temperature span repeatedly. If the bed was rigidly constrained, thermal growth would cause the bed to warp or buckle and would not present a flat build surface, causing defects in first layer adhesion and build plate separation. If the bed were constrained on one edge and designed to allow for thermal growth, the position of the bed would shift relative to the constrained point. This means that any print algorithm that varied the temperature of the bed would move the actual position of the bed from the assumed position of the bed in the system control, making the 3D printer unsuitable as a test platform for any experimentation with fluctuating bed temperatures. A sphere in a V-block can only move parallel to the axis of the vee surface, so by orienting the axis of each V-block to intersect with a point directly under the center of the print bed, the bed is designed to make the

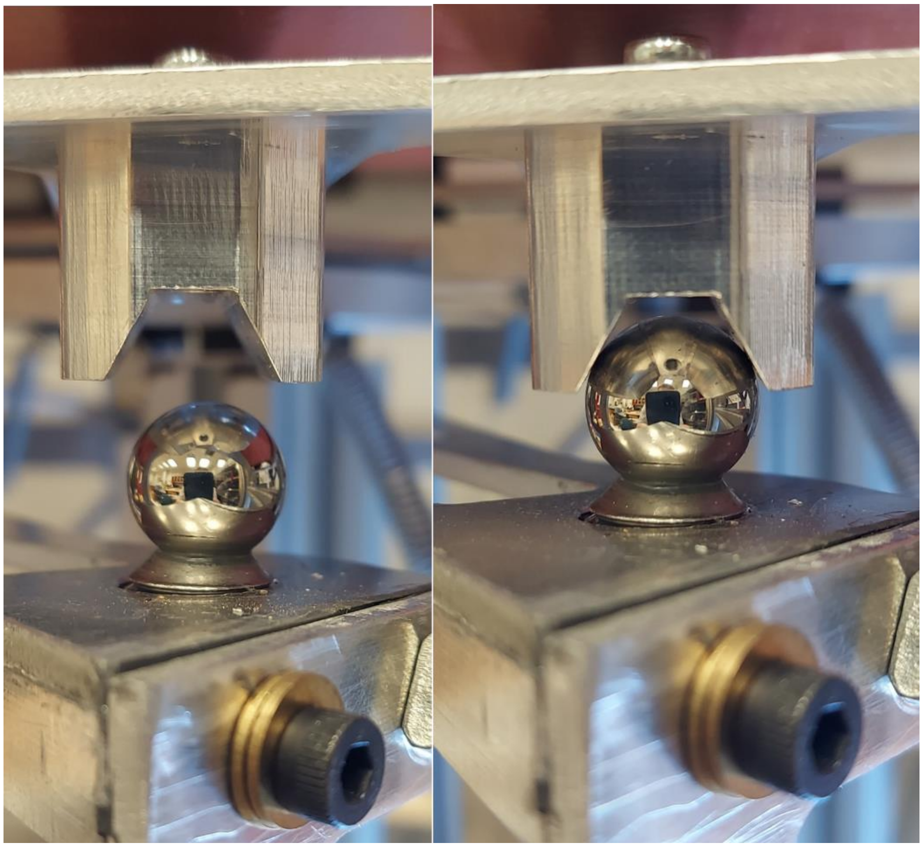


Figure 2.14 Detailed view showing the mated (right) and separated (left) states of one of the three bearing ball and vee block pairs that make up the kinematic coupling.

position of the center of the bed invariant with respect to bed temperature and to remain planar and stress free during thermal growth. Figure 2.14 shows a detail view of the mating between the ball and vee block. Figure 2.15 shows the orientation of the vee blocks. Note that all of the vee blocks point towards the center of the bed which coincides with the center of the bearing shown in the picture.

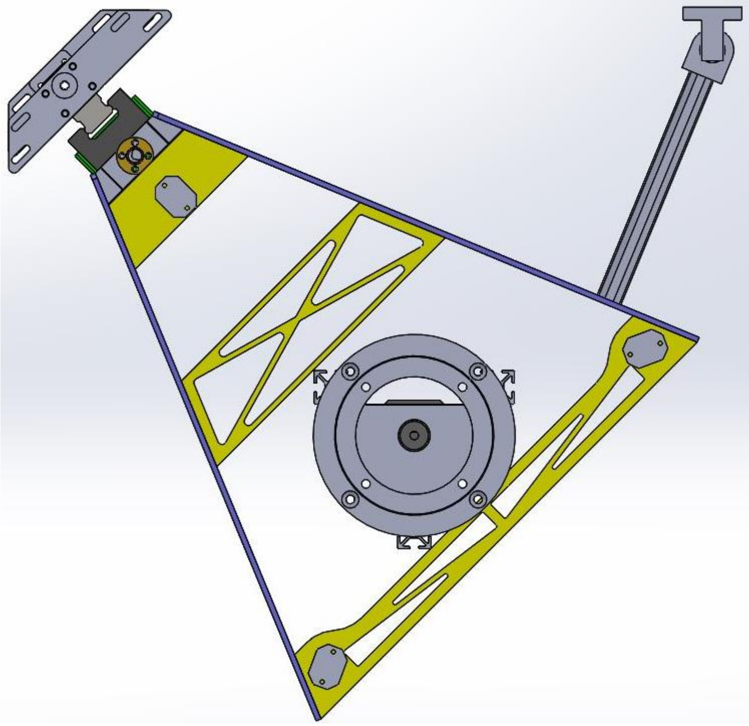


Figure 2.15 Top view of bed rotation mechanism. Note that the vee blocks point towards the center of the bearing, which is mounted in the center of the underside of the build plate.

This kinematic coupling is exactly constrained, which is to say that each degree of freedom has only a single constraining force. This means that the position of the bed is deterministic, and if the connections between the components of the coupling are broken, the bed can be replaced again in almost the exact same spot. This provides a mechanism by which the bed can be separated from the support structure of the Z motion mechanism, rotated to take pictures for photogrammetry, and then returned without disturbing the printing process. Additionally, by placing the bearing spheres on threaded shafts, the coupling also provides a method for leveling the build plate to the XY plane of the print head. A pair of springs provide a nesting force for the kinematic coupling as well as stabilizing the bed during the photogrammetry data capture cycle. These springs can be seen in the right picture of Figure 2.16.

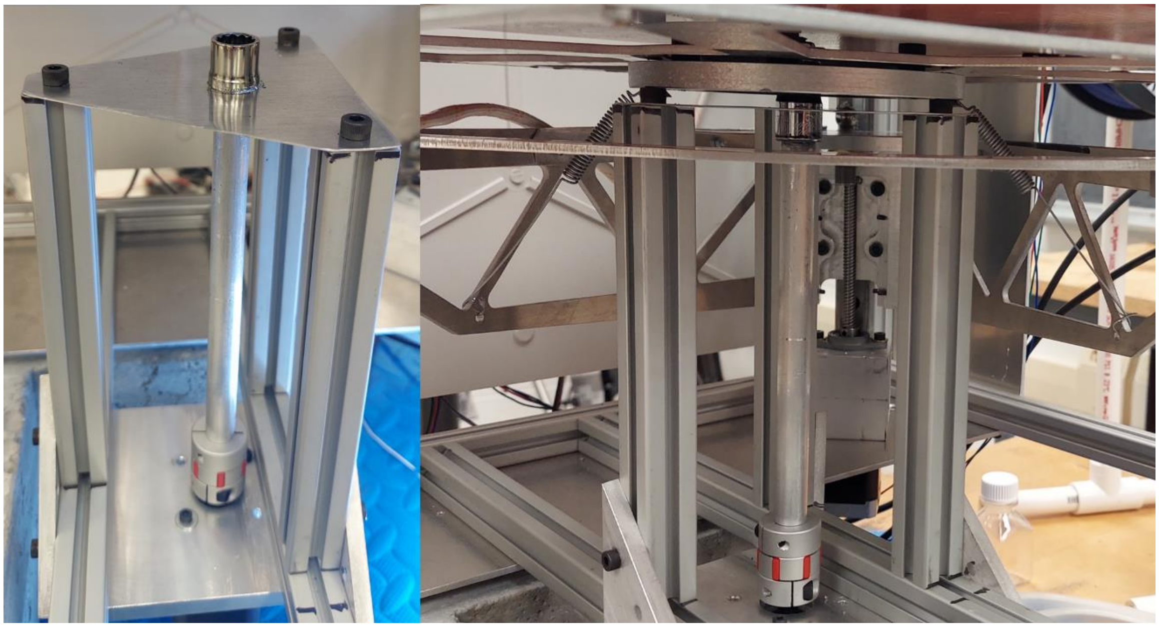


Figure 2.16 (Left) Support structure for the bed rotation mechanism. The three black bolt heads make contact with the outer race of the bearing on the bottom of the bed as shown on the (Right).

To rotate the bed, the inner race of a 120 mm OD diameter bearing is affixed to the bottom of the print bed. The Z axis lowers until the outer race of this bearing comes in contact with three pillars of extruded aluminum that are connected to the bottom of the printer. These pillars then bear the weight of the print bed, and the Z carriage continues to lower until the elements of the kinematic coupling separate. Additionally, there is a hexagonal bolt head attached to the bottom center of the print bed, and when the print bed lowers onto the pillars, this bolt head is engaged by a socket that is connected by a shaft to a NEMA 17 stepper motor. Figure 2.16 shows the support structure, drive shaft, and socket. Since the bed is supported through the bearing, the stepper motor can now be used to rotate the bed. This mechanism is used to position the printed artifact for the camera so that photogrammetry can be performed. Once the capture of pictures is complete, the bed is rotated back into its original position, and the Z carriage is raised until the elements of the kinematic coupling re-engage, and the Z carriage returns the print bed to the appropriate height to resume printing. Figure 2.17 shows the mechanism engaged to allow bed rotation for photogrammetry data capture against the state where the kinematic coupling is engaged for printing.

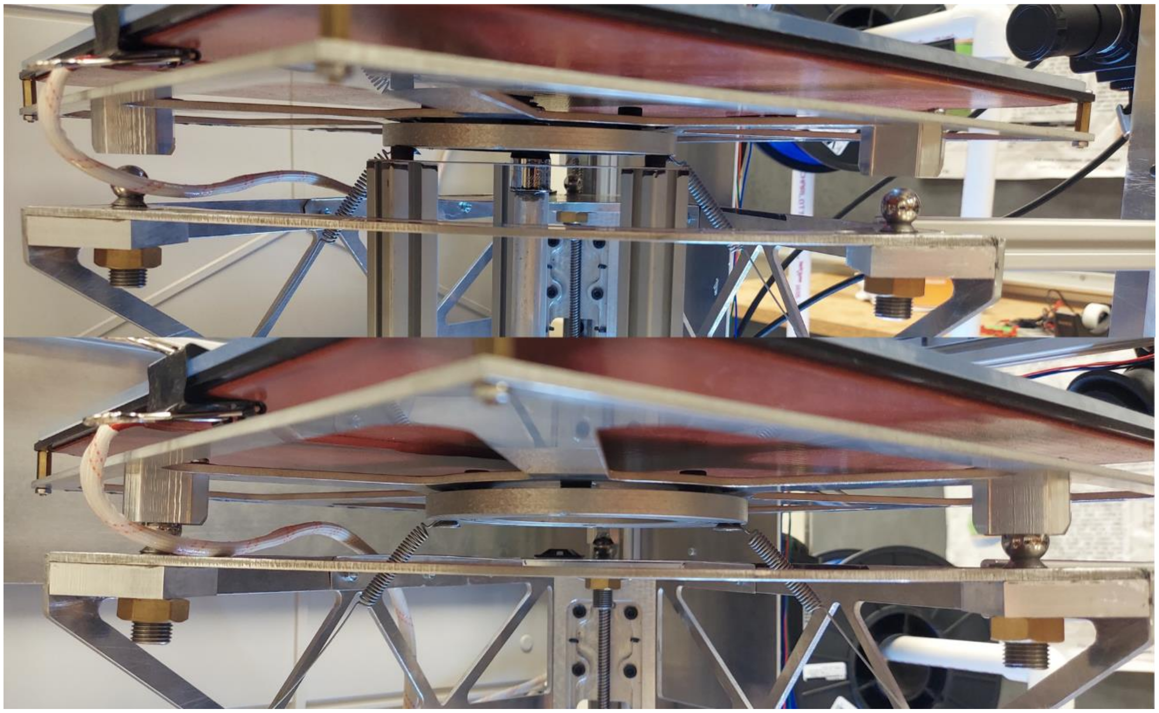


Figure 2.17 (Top) Build plate separated from the Z carriage for photogrammetry data capture and (Bottom) Build plate supported by Z carriage for printing.

A design error was made during the design of the Z axis support system. The Z axis was designed using a single linear rail to aid in ease of tramming the Z axis to the XY motion plane. However, this provided a single tie point for the entire axis and the systems resistance to motion about the Z axis depends entirely upon the stiffness of the connection made by the linear rail and carriage. Even though the carriage is designed with significant preload, this is a bad practice, and the stiffness will degrade with component wear. To compound this problem, the support frame acts as a large flexure, and allows for significant motion at the outer edges of the build plate. This provides unacceptable movement about the Z axis during printer operation. A linear rod was attached to the front left corner of the printer so that it could constrain the undesired motion of the print bed.

The Z axis should have been designed with two linear rails, and the entire sub-assembly should have been constructed as a part that could have been adjusted relative to the XY plane for tramming purposes. The two arms that support the bed should be turned into a closed form using a top and bottom plate that would keep the assembly from acting as a flexure. These two design changes would prevent the motion that this design had and provide an easy method of tramming the Z axis. While the addition of the linear rod in the current design eliminated the unwanted motion, the entire assembly is very difficult to tram. This design approach is discussed more in Chapter 5.

CHAPTER THREE

SATISFACTION OF MECHANICAL REQUIREMENTS

OVERVIEW

Testing was carried out on a whole-system basis after assembling all off the major components. The system verification is separated into physical testing and FEA simulation. FEA simulation was used as a tool for verification of quantities that would be difficult to directly measure such as deflection of the frame and gantry under load. To conclude the chapter, some speed tests were conducted to assess the high-speed printing capabilities of the platform.

TESTING-BASED VERIFICATION

SPEED AND ACCELERATION

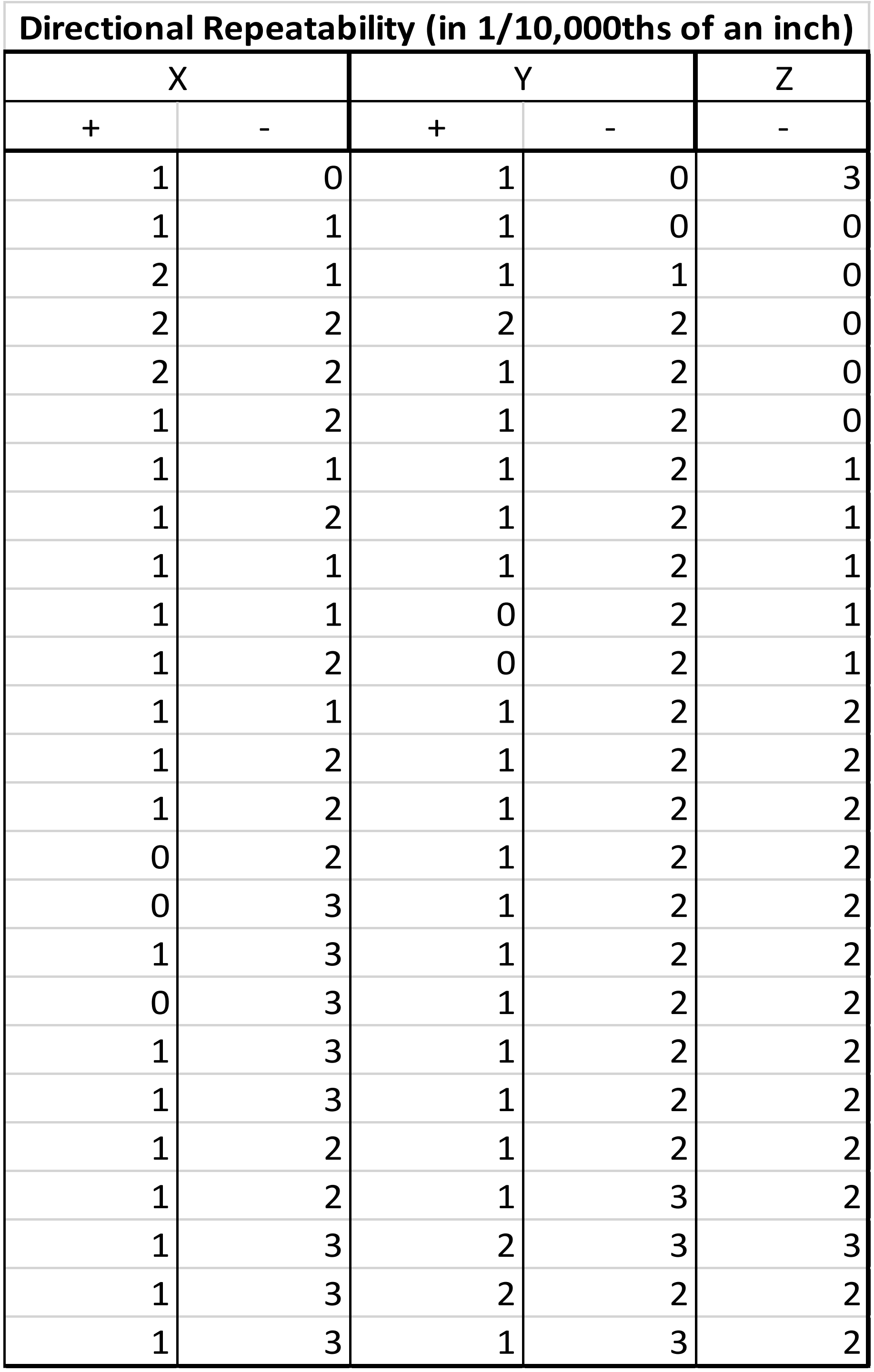
Testing for speed and acceleration was accomplished by tuning the settings in the printer's software and then running a G-code program to move the print head at maximum speed in a repeated X pattern across the build volume. The speed and acceleration settings were incremented upwards until failure points were found. There is a trade-off between these two quantities. Higher acceleration demands more torque from the motors, and torque tends to drop off as speed increases. Therefore, the printer could have several points of operation along this Pareto frontier depending on what kind of experimentation is being investigated. A stable point was determined for the minimum required speed: velocity was set to 200 mm/s (8 in/s) and a stable acceleration found at 9000 mm/s^2 (354.3 in/s^2). However, to attain a reasonable print quality the acceleration was reduced until more tuning can take place. The current acceleration setting is 5000 mm/s^2 (196.8 in/s^2). This speed and acceleration meet the requirements set out for the XY axes, but other combinations can be tested in the future if the need arises.

MOVEMENT REPEATABILITY

The XY resolution and bi-directional repeatability are both desired to be less than 20 microns (0.0008 in). Bi-directional repeatability was tested using a Mitutoyo model 2803S-10 dial test indicator with a resolution of 0.0001 inches, accuracy of ±0.0001 inches, and measurement range of 0.025 inches. The indicator was placed against the side of the print head and a short G-code program was executed to move the print head away from the indicator by 25.4 mm (1 in) and then return to the original position. This was completed 25 times and then the indicator was moved to the other side of the print head and the test was performed from the other direction. The table below shows deviation from starting position in increments of 0.0001 inches. The XY axes have a bi-directional repeatability of ±0.0003 inches. This is 7.6 microns, which is under the desired target of 20 microns.

This test was also performed on the Z axis, but only with the carriage moving in an upward direction. Table 3.1 Results of repeatability testing. These measurements are presented in imperial units because of the native resolution of the testing indicator.Table 3.1Table 3.2 also shows the data for this axis. The Z axis has a single-direction repeatability of 0.0003 inches (7.6 microns).

Table 3.1 Results of repeatability testing. These measurements are presented in imperial units because of the native resolution of the testing indicator.



Directional Repeatability (in 1/10,000ths of an inch)

X		Y		Z
+	-	+	-	-
1	0	1	0	3
1	1	1	0	0
2	1	1	1	0
2	2	2	2	0
2	2	1	2	0
1	2	1	2	0
1	1	1	2	1
1	2	1	2	1
1	1	1	2	1
1	1	0	2	1
1	2	0	2	1
1	1	1	2	2
1	2	1	2	2
1	2	1	2	2
0	2	1	2	2
0	3	1	2	2
1	3	1	2	2
0	3	1	2	2
1	3	1	2	2
1	3	1	2	2
1	2	1	2	2
1	2	1	3	2
1	3	2	3	3
1	3	2	2	2
1	3	1	3	2

MOVEMENT RESOLUTION

The theoretical resolution of the system can be calculated using the belt pitch, number of teeth on the drive pulleys, motor step size, and micro-stepping capability of the motor drivers and for this machine the resolution in the X and Y directions is 12.5 microns (0.00049 in). However, many things play into the actual movement the machine is capable of. In theory, one commanded step from the controller should move the print head 12.5 microns, but due to static friction in the linear rails and pulley bearings, stretch in the belts, and non-linearity of micro-stepped stepper motors this level of motion is not realized. This is to be expected, and this machine does not need incremental motion matching its resolution since the expected distances of motion segments is on the order of one half of the nozzle width or 0.2 mm (0.008 in). Testing showed that an incremental single step from the controller produced motion that ranged from 0.0001 inches to 0.001

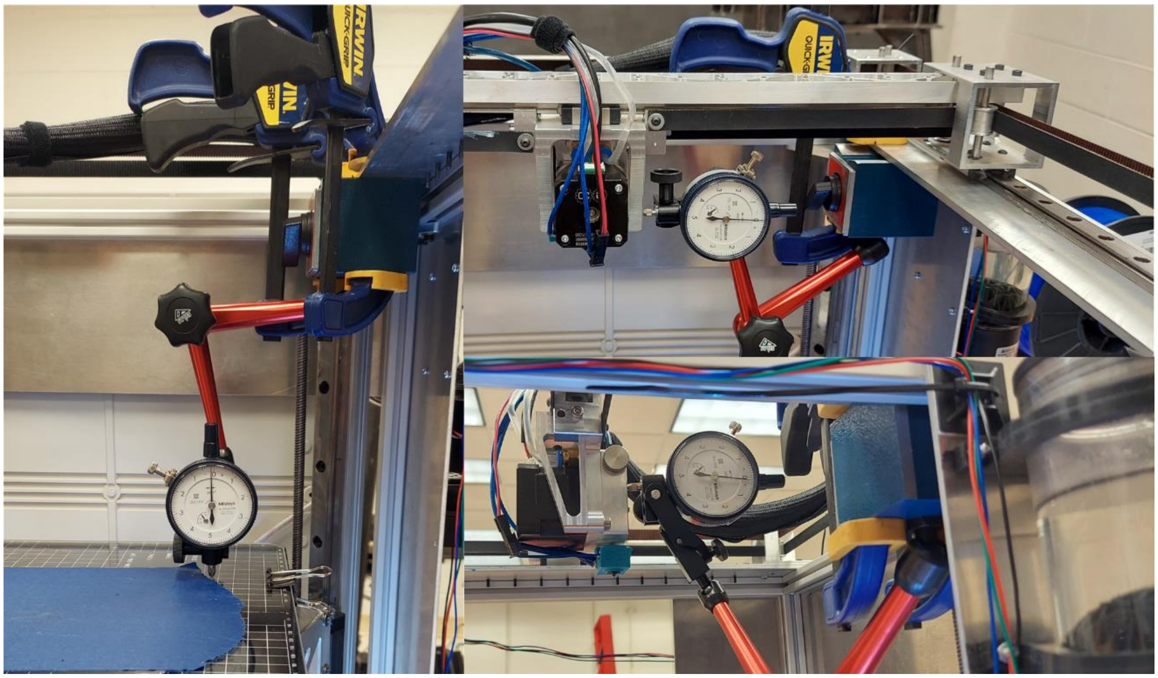


Figure 3.1 Setups for testing resolution and repeatability in (Left) Z direction, (Top Right) X direction, and (Bottom Right) Y Direction

inches (2.54 microns – 25.4 microns). However, normal 3D printing toolpaths will not require an incremental move this small. Contrast this with a coordinate measurement machine, where any incremental movement commanded by the machine controller must be represented by a corresponding movement from the measuring probe head. So instead of characterizing resolution by testing incremental motion, a method similar to that used for repeatability was used. The dial test indicator was setup as shown in Figure 2.3 for each respective axis. The print head was moved away from the indicator and then commanded back to the starting position using a simple G-code program. This was performed 15 times. Then the G-code program was altered to return the print head to a position 12.5 microns (0.00049 in) closer to the dial test indicator. This was also repeated 15 times. This process was repeated until 5 steps had been completed: a total distance of 50 microns (.002 in) from the starting position. The repeated trials help to account for the repeatability of the axes. Table 3.2 shows the data from these tests for the X, Y, and Z directions.

Table 3.2 Resolution test for X, Y, and Z axes

Axis	**X**					**Y**					**Z**		
Commanded Position (microns)	0	13	25	38	50	0	13	25	38	50	0	2.5	5
Average Position (microns)	0.0	0.0	8.3	21.0	33.4	0.0	20.5	25.1	33.7	42.8	0	1	0.7
Actual Step Size		0.0	8.3	12.7	12.4		20.5	4.6	8.6	9.1		1	-0
	X	**Y**	**Z**										
Average Step Size	8.3	10.7	0.3										

This testing indicates one of the problems with using only a calculated theoretical resolution to characterize a machine's minimum movement capabilities. The step sizes for all axes are very inconsistent. The Y axis has the closest average step size, with 10.7 microns (.00042 in) vs the theoretical value of 12.5 microns (0.00049 in), but examining the individual increments shows the first increment to be a step of 20.5 microns (0.0008 in), and the second step to only be a quarter of this value. After the linear encoders are implemented on the printer, a C program can be written to automate this testing and a more rigorous evaluation can be performed.

POSITION ACCURACY

Currently, we are unable to test the position accuracy of the printer. The linear position sensors for implementing closed loop control were also to be used for testing the accuracy of the printer. Without them the lab does not have tools capable of testing this quantity. Instead, two 1-inch cubes were printed and measured. These parts were printed using ABS filament at 100 mm/s (4 in/s) and an acceleration of 2000 mm/s^2 (78.7 in/s^2) and 8890 mm/s^2 (350 in/s^2) respectively. Figure 3.2 shows these parts and Table 3.3

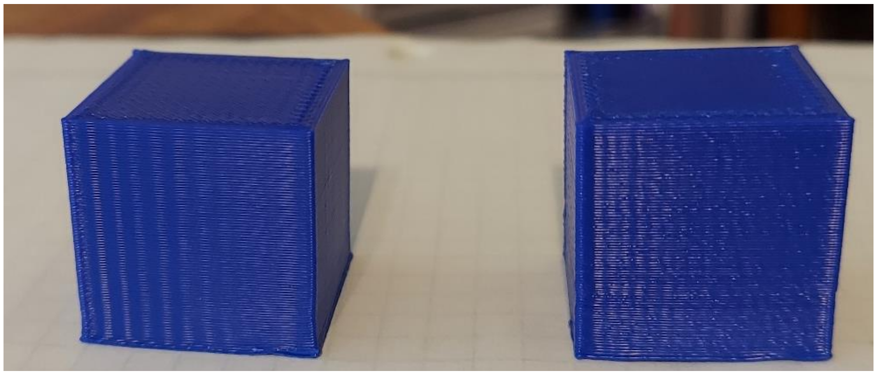


Figure 3.2 25.4 mm (1 in) cubes printed to test accuracy of the machine. Both cubes were printed at 100 mm/s (4 in/s), but the cube on the left was printed at 2000 mm/s^2 (80 in/s^2) acceleration and the cube on the right was printed with 8890 mm/s^2 (350 in/s^2) acceleration.

shows the theoretical and measured dimensions. Out-of-plane ringing, or “ghosting”, is evident in these parts on all XY faces indicating that the belt tension needs to be adjusted.

Table 3.3 Dimensions and print times for 25.4 mm cubes printed in ABS

	X Axis [mm]	*Y Axis [mm]*	*Z Axis [mm]*	*Print Time*
Cube 1- 2000 mm/s^2	*25.146*	*25.146*	*25.070*	*16:20*
Cube 2- 8890 mm/s^2	*25.044*	*25.121*	*25.578*	*11:33*

KINEMATIC COUPLING REPEATABILITY

The kinematic coupling implemented on the build plate make it possible to rotate the plate and capture pictures of the print from all sides to perform photogrammetry. The other requirement for this mechanism is that it cannot adversely affect the print. The biggest failure point of this mechanism is the position repeatability every time the coupling is disconnected and reseated during the photogrammetry data capture cycle. Any horizontal variation in the bed position after rotating the bed will affect how well successive layers line up, and any vertical variation will induce a non-uniform thickness to each layer after a data collection cycle. To test this, a laser pointer was connected to the bed and the laser was projected on a wall 9450 mm (31 ft) away from the center of the print bed. A G-code program was executed to move the Z carriage downwards until the bed disconnected from the support frame and then bed was rotated 15 degrees. The bed was then rotated back into place and the carriage raised until the bed was reseated on the kinematic coupling. The location of the laser dot on the wall was marked and this was repeated 25 times. However, this testing method was not able to capture any variation in the laser spot position. This testing indicates that the repeatability of the kinematic coupling will not adversely affect the printing process. Figure 3.3 shows the setup for this test.

SIMULATION-BASED VERIFICATION

FRAME DEFLECTION UNDER LOAD

FEA simulation was performed to test the requirements for frame and gantry deflection. Both of these sub-assemblies were not allowed to deflect more than 20 microns (0.0008 in) during load. These analyses were performed twice, once before manufacture with nominal accelerations and once upon testing the actual acceleration capability of the system. The first analysis provides design justification and the second

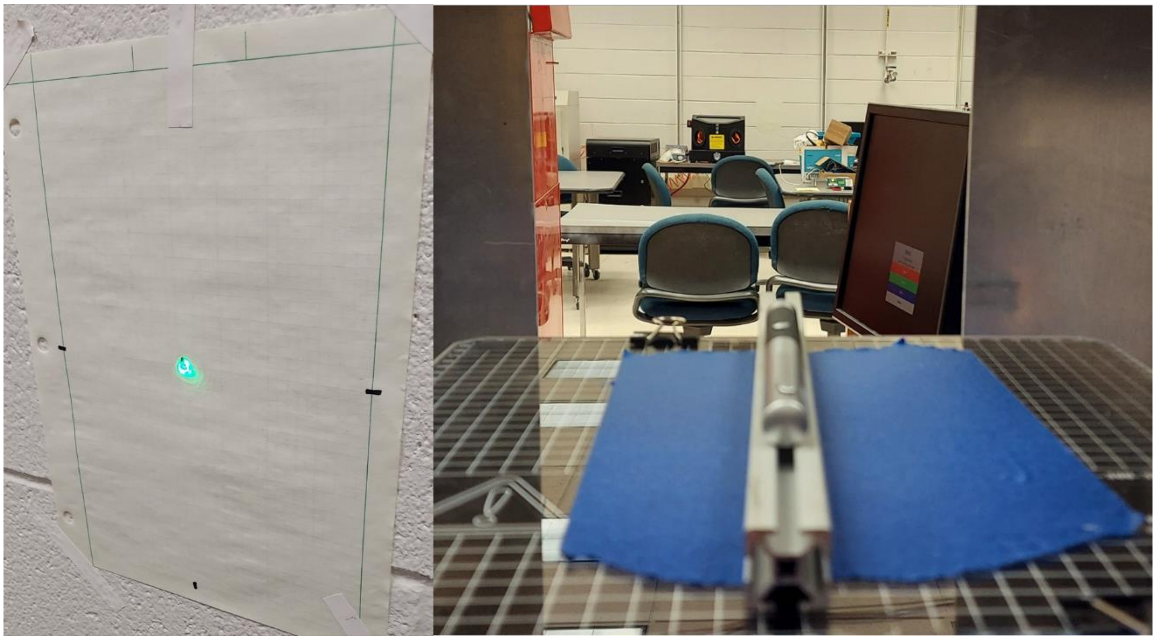


Figure 3.3 (Left) Laser pointer spot on wall 31 feet from the center of print bed and (Right) view overlooking the laser pointer towards the target wall.

provides verification that the desired accuracy can be attained using the full acceleration of the machine. Initial testing was completed using an acceleration of 3000 mm/s^2 (9.84 ft/s^2) and secondary testing with a value of 9000 mm/s^2 (29.53 ft/s^2). The frame is fixed to the concrete base by the two bottom plates, so these plates were anchored in the simulation. The faces of the bolt holes for attaching the frame to the base were used as fixed supports. A remote force equal to the force generated by accelerating the entire gantry assembly and print head at the respective rates mentioned above was placed on the top surface of the left and right upper extrusion members of the frame. The maximum distortion experienced by the frame in the X direction is 0.61 microns (0.00002 in) for the initial simulation and 1.84 microns (0.00007 in) for the final simulation. These are within the 20 micron (0.0008 in) requirement for this quantity.

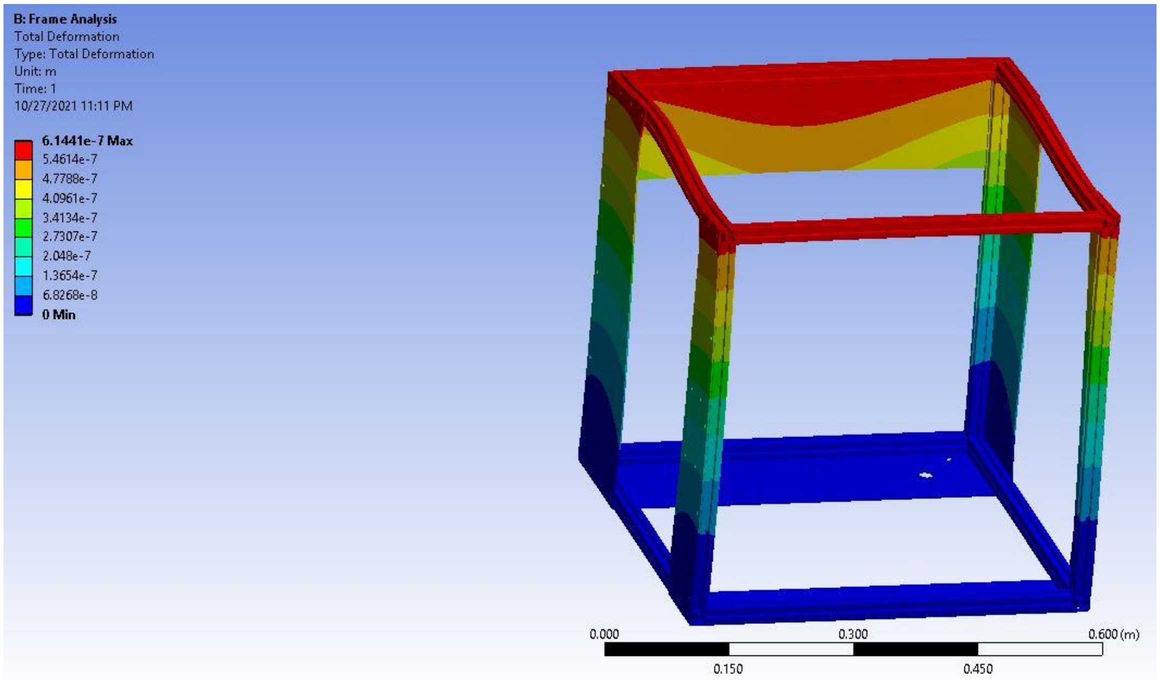


Figure 3.4 FEA simulation of printing forces on the frame under accelerations of 3000 mm/s^2

Deflection in the Y direction was simulated similarly, except that the mass of the

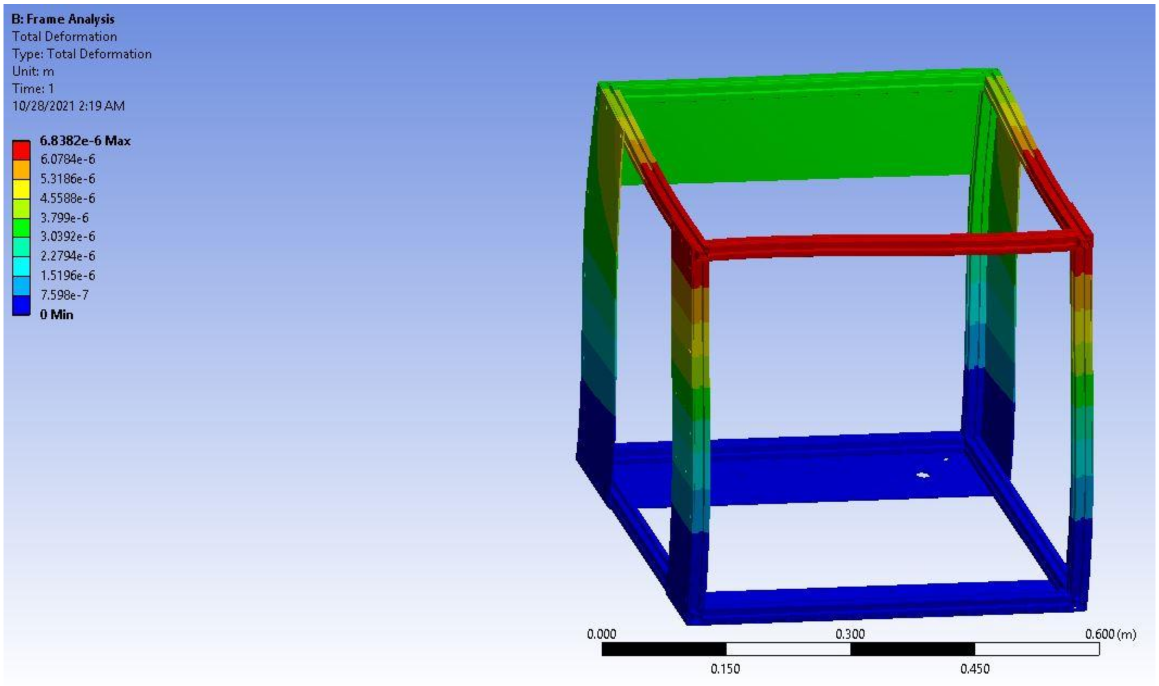

B: Frame Analysis
Total Deformation
Type: Total Deformation
Unit: m
Time: 1
10/28/2021 2:19 AM

Figure 3.5 FEA Simulation showing forces in the Y direction under acceleration of 3000 mm/s^2

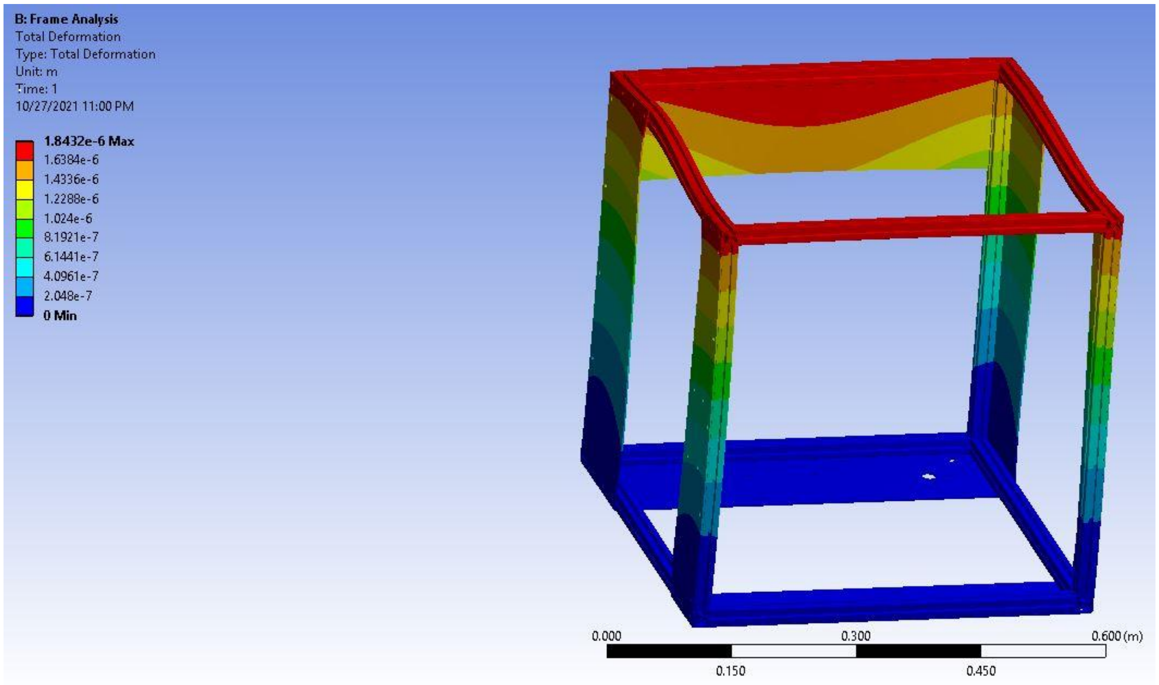

B: Frame Analysis
Total Deformation
Type: Total Deformation
Unit: m
Time: 1
10/27/2021 11:00 PM

Figure 3.6 FEA simulation of printing forces on the frame under accelerations of 9000 mm/s^2

gantry sub-assembly was not used and only the mass of the print head contributed to the remote force. Maximum distortion in the frame is 6.8 microns (0.00027 in) for the initial simulation and 13.7 microns (0.00054 in) for the final simulation. The deflections are also within the 20 micron (0.0008 in) requirement.

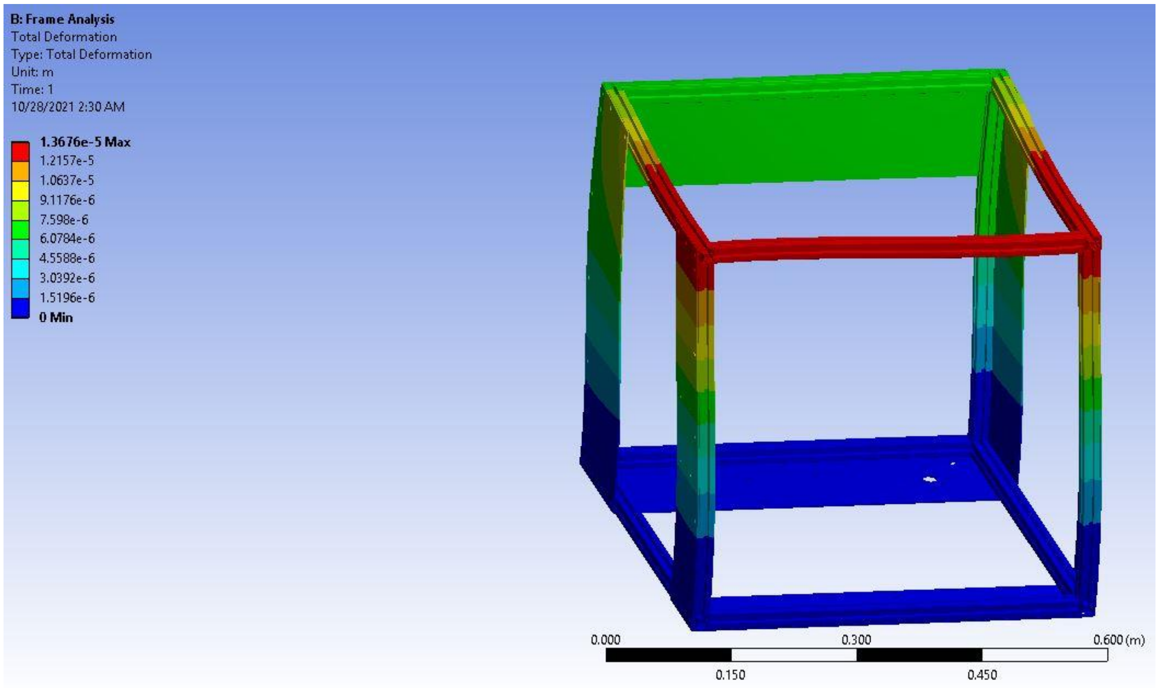


Figure 3.7 FEA simulation showing forces in the Y direction under acceleration of 9000 mm/s^2

It should be noted that this is a conservative estimate of frame deflection. The sub-assembly for mounting the linear rails for the Y direction add stiffness to the frame assembly but are not modeled here. This should lend confidence to the conclusion that this requirement has been satisfied.

GANTRY DEFLECTION UNDER LOAD

The simulation for the gantry was performed only in the Y direction, since the X direction will be much stronger than the Y. If the Y direction meets the deflection requirement, so will the X direction. The faces of the bolt holes securing the gantry to the linear rail carriages were used as fixed supports and a remote force was applied to the linear rail. The force was located in the center of the gantry, 40.6 mm (1.6 in) below the bottom surface of the linear rail in the same location as the center of gravity for the extruder assembly. This force was calculated using the mass of the print head and accelerations of 3000 mm/s^2 (118.1 in/s^2) and 9000 mm/s^2 (354.3 in/s^2) for the initial and final simulations respectively. Maximum deformation for the initial simulation was 1 micron (0.000034 in) and for the final simulation was 3.3 microns (0.00013 in). These

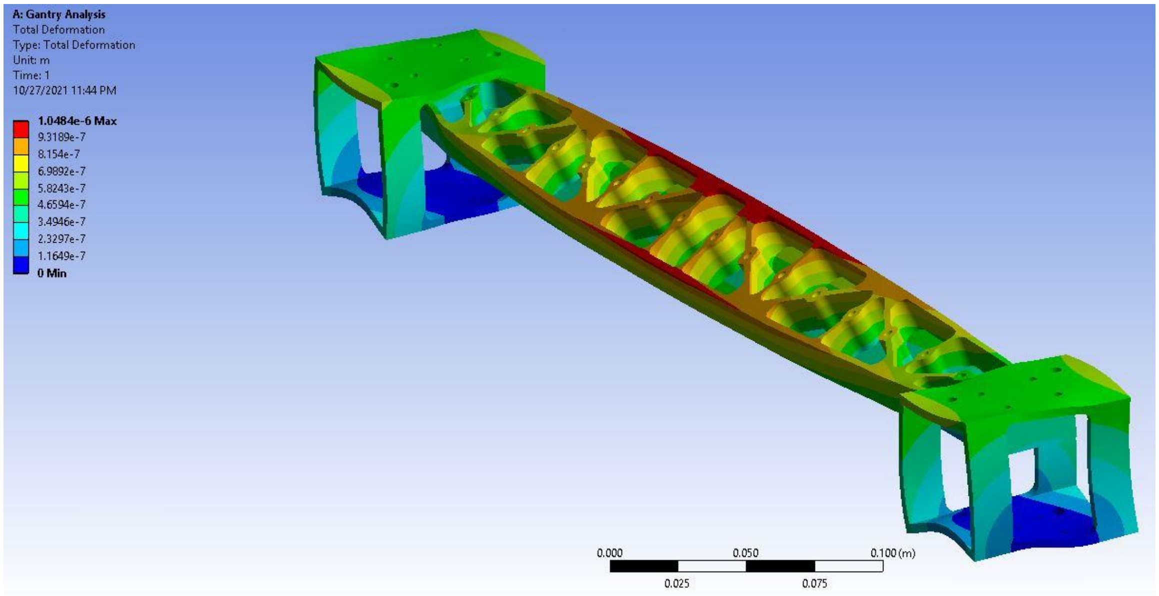


Figure 3.8 FEA simulation of printing forces on the gantry under accelerations of 3000 mm/s^2

values are below the 20 micron (0.0008in) requirement. Figure 3.8 and Figure 3.9 show total deformation plots for both of these simulations.

Z SUPPORT OPTIMAZTION

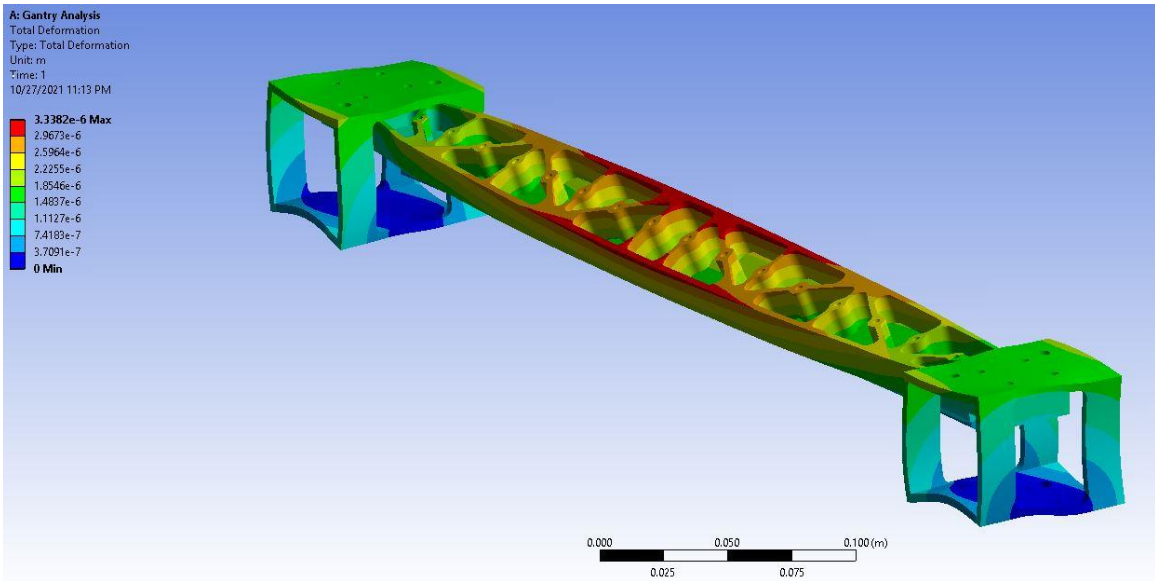


Figure 3.9 FEA simulation of printing forces on the gantry under accelerations of 9000 mm/s^2

The shape of the Z supports was formulated as an optimization problem to ensure that the added weight of printed parts would not cause a change in build plate orientation or positioning. Most filament spools are sold in 1 kilogram quantities but can also be purchased as 5 kilogram spools. To ensure that there is plenty of overhead to ensure that the bed does not move, these simulations were performed simulating the 1 kg weight of the bed, and 9 kgs of printed plastic. A force of 100 Newtons was applied as a remote force on the faces of the bolt holes that hold the leveling screws. The faces of the bolt holes to connect the assembly to the Z rail carriage were used as fixed supports. The supports were modeled as a truss shape and the length and orientation of the top of the support were fixed. The length and angle of the bottom and connecting members were allowed to vary as was the width of all members. The final optimal shape can be seen in Figure 3.11. Figure 3.10 shows the total deformation map of the assembly using the final truss shape. Maximum deformation at the bolt holes is less than 0.2 millimeters (0.0079 in). There is a distance of 292 mm (11.5 in) between these bolt holes and the bolt hole near the Z carriage. Using a small angle approximation, this translates to an angle change

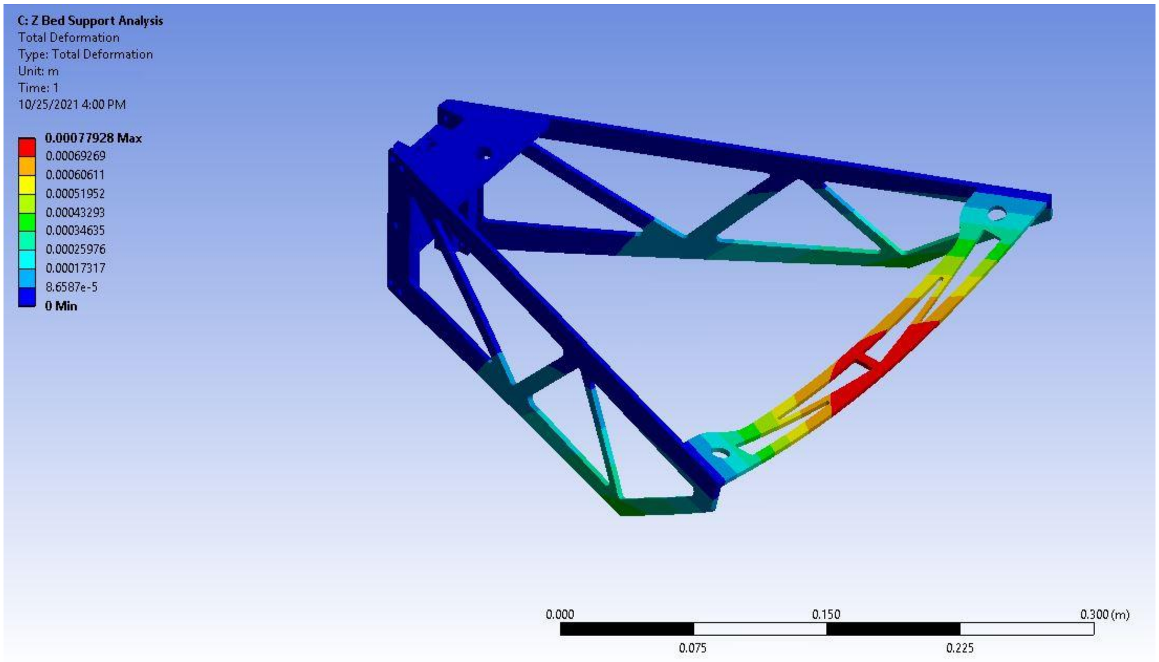


Figure 3.10 FEA simulation of 100N force being transferred to Z support structure through the leveling screws.

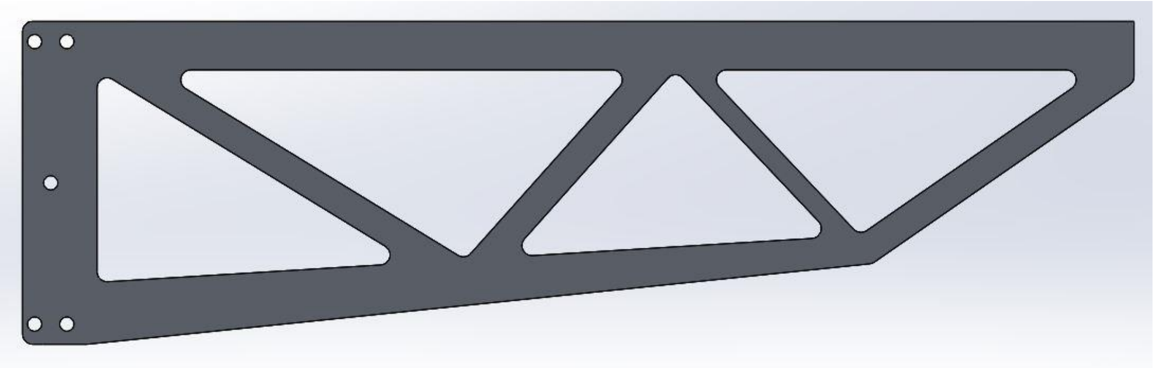


Figure 3.11 Final optimized shape of the Z support for the build plate

of less than 0.04 degrees. While the deformation map shows that the side supports are sufficient for their purpose, the crossbar connecting them should be the subject of a future re-design.

CHAPTER FOUR

ELECTRONICS AND SYSTEM CONTROL

ELECTRONIC SYSTEMS OVERVIEW

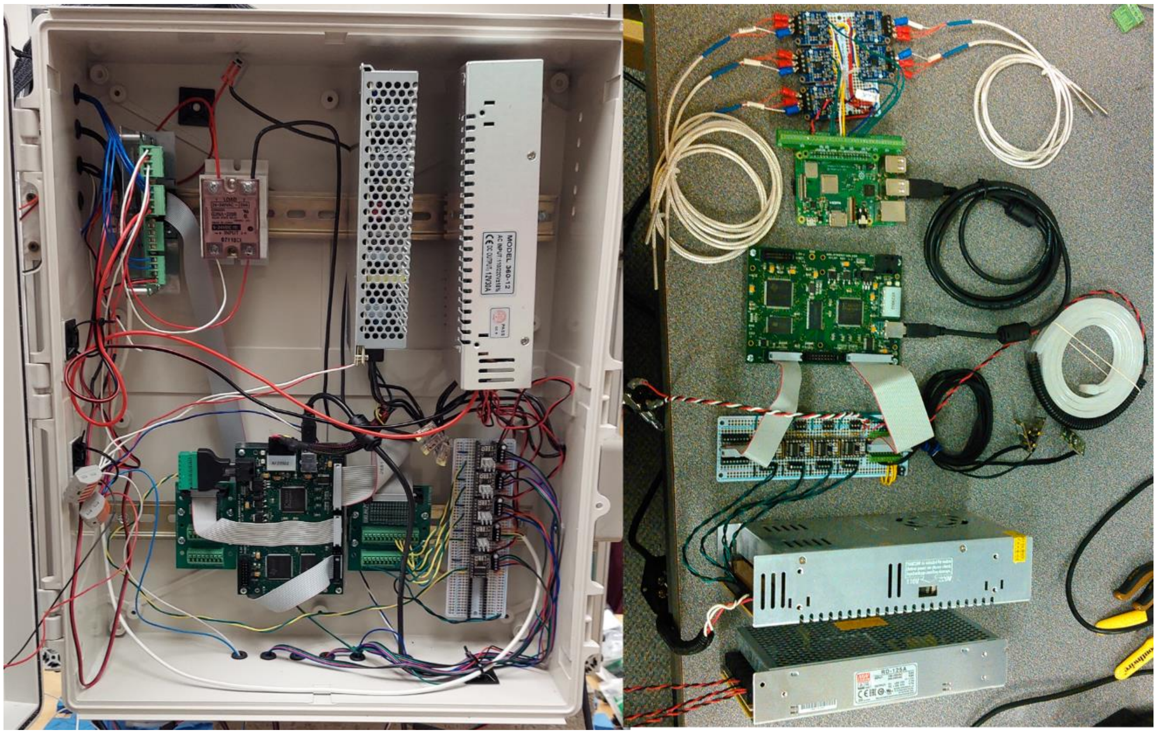


Figure 4.1 The electrical system for the 3D printer (Right) during assembly and (Left) inside the electronics cabinet.

The electronic system that drives the printer is made up of three different controllers. A motion controller board is tasked with dealing with the motion of the printer and controlling the temperatures of the hot end and heated bed. The motion controller coordinates and synchronizes the functions of the other controllers as well as recording the temperature of the heated bed and hot end and the position of bed and print head during a print. The KFLOP motion controller from Dynomotion was chosen for this project and is discussed more in following section on Motion Control. A microcontroller is used to sample and record data from a variety of sensors within the build environment. The Raspberry Pi 3 was chosen for this task and is discussed in detail later in this chapter in the section on Sensors and Data Recording. Figure 4.1 shows the electronics system. A separate controller board capable of interacting with one or more cameras and performing the intensive calculations for photogrammetry was also needed. A Jetson TX2 from Nvidia was chosen for this task and is discussed further in this chapter in the section on the Photogrammetry Software and Toolchain.

Three different voltage levels are required for different components in this system: 5VDC, 12VDC, and 120VAC. 120VAC wall power is connected to a switch that is used to turn the entire printer on and off. Power from the switch is run to a solid-state relay that is used to control the power to the heated bed, and two separate power supplies. One power supply is dual voltage that supplies 5VDC and 12VDC for running the KFLOP, Raspberry Pi, and various sensors and accessories. The other supply is a 12VDC 30A supply that is powers the stepper motors used to motivate the printer. The KFLOP control board is connected to a series of DRV8825 stepper motor drives from Pololu that are used to drive the stepper motors. Each DRV8825 board take inputs of step and direction signals from the KFLOP and 12VDC power from the second power supply and outputs power to a stepper motor. 12VDC from the first power supply is passed through the KFLOP board and is used to power an interior LED light strip and the fan and pump for the water-cooling system. This power is also connected to the KNozz board and used to power the hot end and control the solid-state relay for the heated bed.

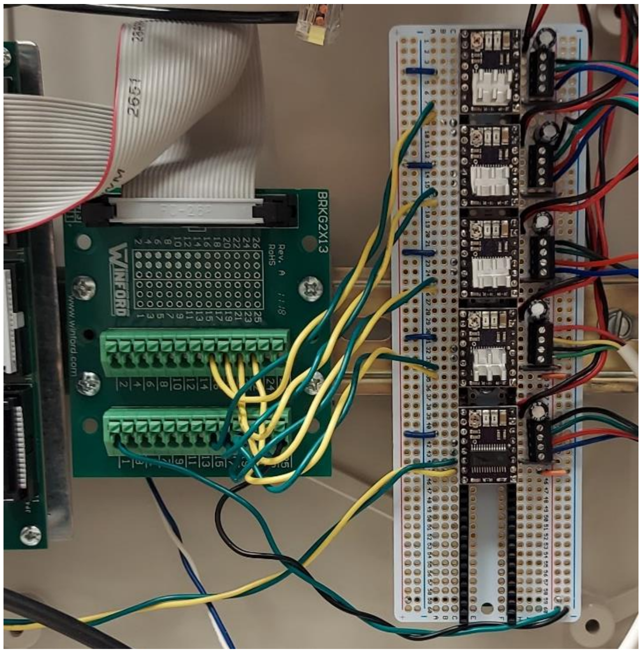


Figure 4.2 The custom board for interfacing with the DRV8825 stepper motor drivers.

5VDC is also fed to the Raspberry Pi 3 and used to power its network of sensors. The Raspberry Pi is connected to five PT1000 RTD sensors and their respective breakout boards from Adafruit, and a humidity sensor that is also from Adafruit. The Raspberry Pi receives start, stop, and periodic synchronization commands from the KFLOP board to record sensor readings during a print.

The Jetson TX2 receives 19VDC from its own power supply (included with the board) and is connected to two USB webcams via a USB hub. These webcams are from Arducam and feature the 2MP AR0230 CMOS sensor. The cameras are placed outside the build environment and are used to capture photographs of the printed artifact for photogrammetric reconstruction. At the moment, the TX2 takes signals from the KFLOP board to tell it when to take a picture and when to compile a batch of photographs into a new point cloud but is not otherwise integrated into the control system. In the future, it is desired that the TX2 be more connected and perhaps even take over the function of the Raspberry Pi but the software development for this has not yet taken place. Figure 4.3 shows the Jetson TX2 and a picture of the two web cameras used to collect data for the photogrammetry process.

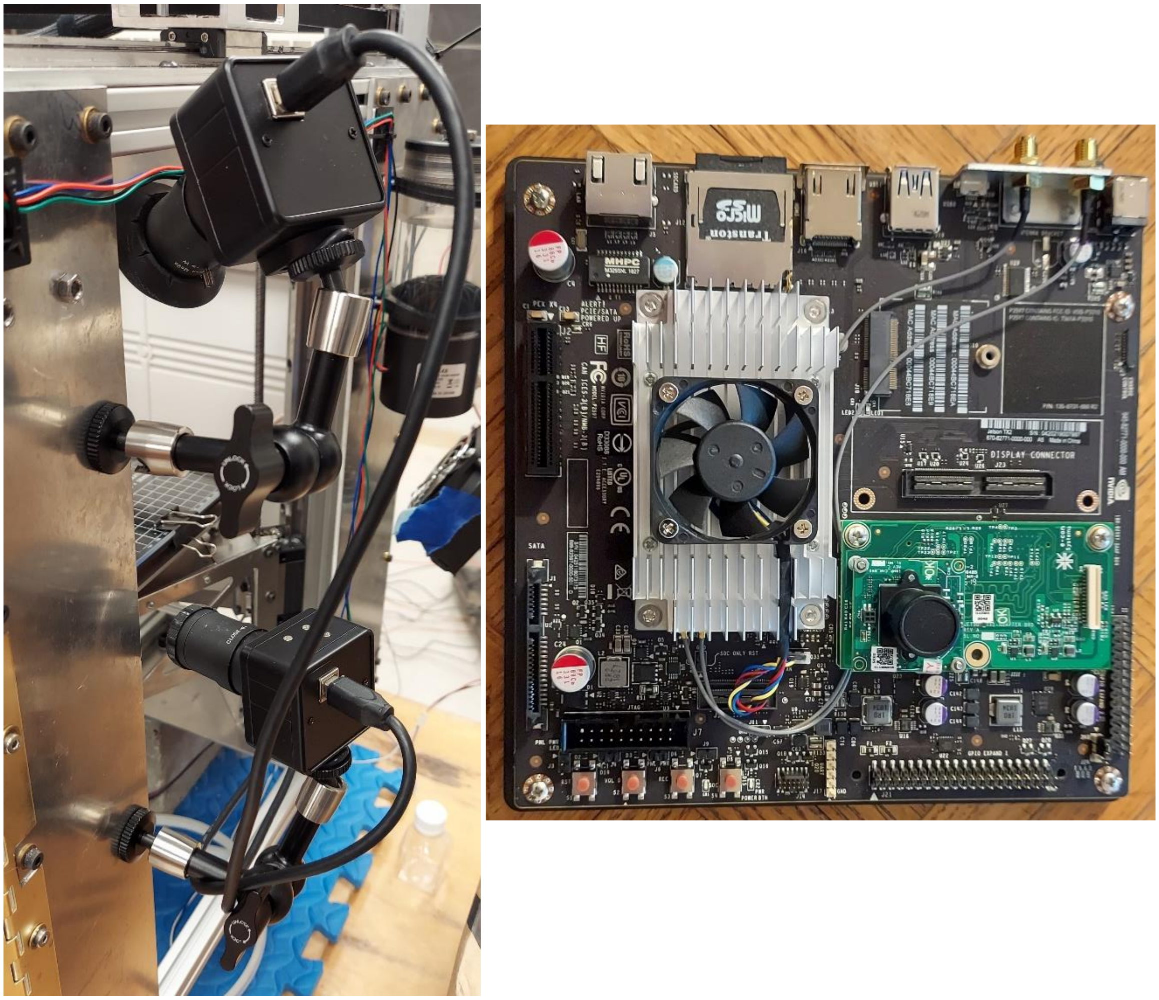


Figure 4.3 (Left) Two-megapixel USB web cameras from Arducam mounted to the right side of the printer frame and (Right) a NVIDIA Jetson TX2

MOTION CONTROL

RELEVANT REQUIREMENTS

Requirements for the motion control system are mostly derived from the general movement requirements for the printer. The control system must be able to drive the chosen motors at the desired rates and desired accuracy and repeatability. Additionally, the motion control system must have the ability to interface and control peripheral components such as the heated bed and hot end. The controller must also be able to support the type of motion architecture chosen for the printer. Since the printer is supposed to be a testbed, it should also be able to cope with currently unforeseen control needs and should have extra input and output channels for doing so. The controller should also have an interface that allows for customization to help meet future needs. It is desired that the printer operate using closed loop control for axis movement, and the controller must be capable of accepting encoder feedback for this purpose.

CONTROLLER CHOICE AND SPECIFICATIONS

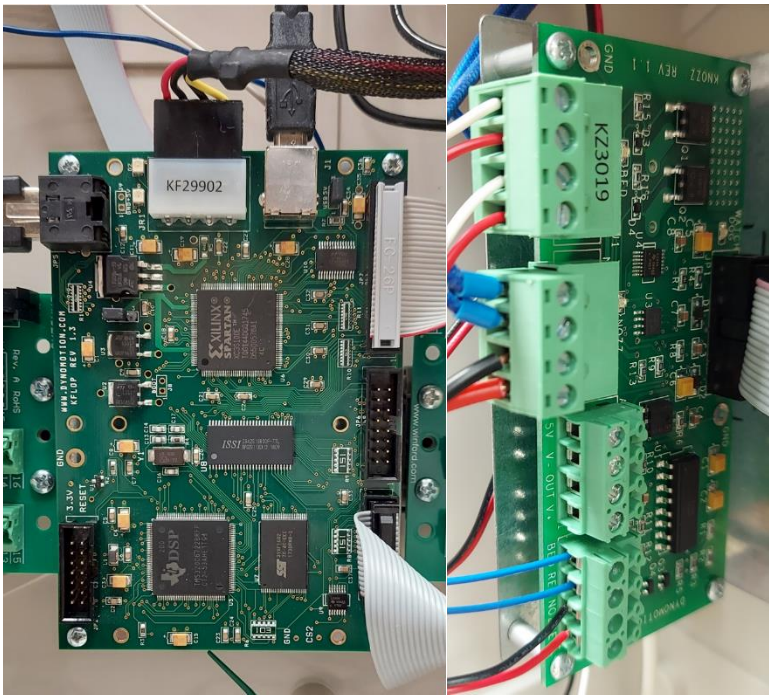


Figure 4.4 (Left) The KFLOP motion controller from Dynomotion and (Right) the Knozz daughter board that KFLOP uses to monitor and control bed and print head temperatures.

The KFLOP motion controller from Dynomotion. This controller has the ability to coordinate eight channels of motion through a number of control schemes including step and direction signals and can operate at a speed of 2.5 MHz. The controller accepts 8 channels of encoder feedback and has built in PID routines for motor control. Additionally, it has 46 digital I/O ports that operate on 3.3V logic but are 5V tolerant and 8 channels that are capable of PWM signals.

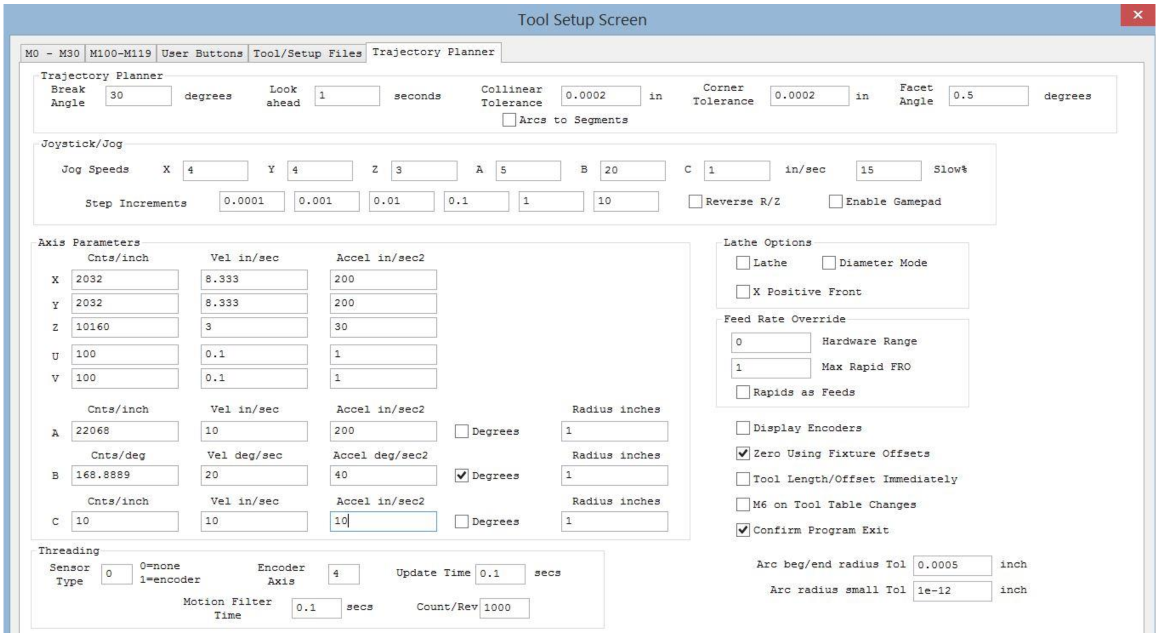
The controller is accompanied by an open-sourced software for manually commanding the controller as well as loading and running G-code files. This software can run C programs and can link custom M codes to C programs which provides a very simple and expedient method of creating custom control routines that can be invoked in a normal G-code program. Additionally, the software is accompanied by an entire library of control subroutines that can be easily accessed when writing custom C programs. This C functionality can be accessed through APIs for C#, VisualC++, and any .Net compatible language. An important aspect of this open software is the ability to write and implement custom kinematic profiles since the printer uses the CoreXY architecture instead of normal cartesian motion.

To aid in the goal of supporting future needs, the KFLOP controller has an ecosystem of other boards that it can be linked to in order to expand its capabilities. One board in particular, the Kanalog board, adds eight ±10VDC analog-to-digital channels, ±10VDC digital-to-analog channels, and eight opto-isolated inputs that are tolerant of up to 24VDC, and will be very useful for adapting to future needs. It was not purchased for this project but having this board available if needed in the future was part of the equation when choosing this controller.

A daughter board that was purchased for this project was the KNozz board. This board provides two 12-bit ADC thermistor channels linked to two 48VDC channels operated with FET switches that can handle up to 20 amps and a watchdog function to keep track of both heaters. This gives the KFLOP controller the ability to control both the heated bed and hot end while monitoring the temperature of both. The watchdog utility monitors the thermistor channels and shuts down the linked heater channel if there is no measurement signal within a certain amount of time. This adds an element of safety and protects against damaged or missing sensors. In the future, the resistors linked to the input channels for measurement of the thermistors can be bypassed to link a RTD temperature sensor instead of a thermistor.

MOTION AND SLICER SOFTWARE

The KFLOP controller board has its own software, KMotionCNC, that is available for use with the controller board. Other options exist for control software such as Mach3, or even custom written software. However, the KMotionCNC software satisfies the needs of this project, is free, and requires no additional work on the part of the author. KMotionCNC can run G code programs, provides an open display GUI with a customizing utility to modify the display as needed, and makes machine setup and customization very easy. The software also provides utilities to tune servo or stepper motors, monitor IO, measure step responses, and write and download C programs to the control board. Unlike most 3D printers where machine parameters such as velocity limits, acceleration and jerk settings must be accessed through the firmware on the machine, KMotionCNC has a Trajectory Planner page that keeps all of these setup numbers easily accessible. While the setup file for the machine records the absolute maximum values for the machine, the Trajectory Planner makes it easy to change settings depending on experiment requirements without having to change the setup file. For instance, initial machine testing may indicate that 5000 mm/s^2 (200 in/s^2) is the maximum acceleration the machine can withstand without overloading the stepper motors. This number would be recorded in the setup file and will not change during machine operation. However, if you wanted to limit the acceleration to 1200mm/s^2 (50 in/s^2) for a certain experiment it can be done by changing the value in the Trajectory Planner instead of having to modify the setup file. The Trajectory Planner values cannot override or exceed the values in the setup file. This is an incredibly useful tool for a research platform since it will allow for easy replication of other testing environments, such as trying to copy or confirm an experiment performed on another printing platform or replicate a specific build environment.



Trajectory Planner

Break Angle 30 degrees
Look ahead 1 seconds
Collinear Tolerance 0.0002 in
Corner Tolerance 0.0002 in
Facet Angle 0.5 degrees
□ Arcs to Segments

Joystick/Jog

Jog Speeds X 4 Y 4 Z 3 A 5 B 20 C 1 in/sec 15 Slow%
Step Increments 0.0001 0.001 0.01 0.1 1 10 □ Reverse R/Z □ Enable Gamepad

Axis Parameters

	Cnts/inch	Vel in/sec	Accel in/sec2		
X	2032	8.333	200		
Y	2032	8.333	200		
Z	10160	3	30		
U	100	0.1	1		
V	100	0.1	1		
	Cnts/inch	Vel in/sec	Accel in/sec2		Radius inches
A	22068	10	200	□ Degrees	1
	Cnts/deg	Vel deg/sec	Accel deg/sec2		Radius inches
B	168.8889	20	40	☑ Degrees	1
	Cnts/inch	Vel in/sec	Accel in/sec2		Radius inches
C	10	10	10	□ Degrees	1

Threading

Sensor Type 0 0=none 1=encoder
Encoder Axis 4
Update Time 0.1 secs
Motion Filter Time 0.1 secs
Count/Rev 1000

Lathe Options

□ Lathe □ Diameter Mode
□ X Positive Front

Feed Rate Override

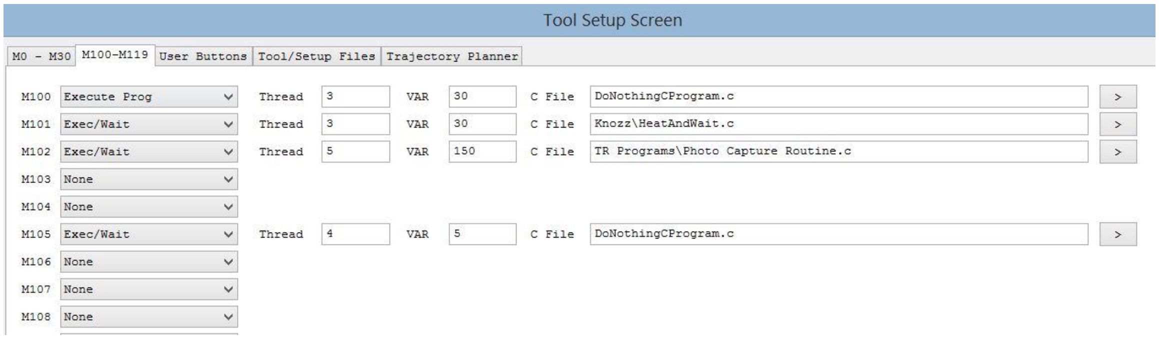
0 Hardware Range
1 Max Rapid FRO
□ Rapids as Feeds

□ Display Encoders
☑ Zero Using Fixture Offsets
□ Tool Length/Offset Immediately
□ M6 on Tool Table Changes
☑ Confirm Program Exit

Arc beg/end radius Tol 0.0005 inch
Arc radius small Tol 1e-12 inch

Figure 4.5 KMotionCNC Trajectory Planner screen.

Additionally, the KMotionCNC makes it very simple to define M words in G code. Upon encountering the M words M0-M9, and M100 – M119, KMotionCNC can take several predefined actions such as setting, monitoring, or flipping an IO bit, setting the internal digital-to-analog converter, or executing a C program on the control board or the host PC. The KFLOP board can run up to seven C programs concurrently and can pass variable values through to these C programs using P, Q, and R parameters in conjunction with M words. This function makes the KFLOP and KMotionCNC an open platform that is easy to use and program for specialty functions. This feature is used in this project to implement functions such as homing, setting and controlling bed and hot end temperature, and performing photogrammetry data capture. The setup screen for defining new M codes can be seen in Figure 4.6



M100	Execute Prog	Thread	3	VAR	30	C File	DoNothingCProgram.c
M101	Exec/Wait	Thread	3	VAR	30	C File	Knozz\HeatAndWait.c
M102	Exec/Wait	Thread	5	VAR	150	C File	TR Programs\Photo Capture Routine.c
M103	None						
M104	None						
M105	Exec/Wait	Thread	4	VAR	5	C File	DoNothingCProgram.c
M106	None						
M107	None						
M108	None						

Figure 4.6 The KMotionCNC Tool Setup screen.

One of the quirks of the way the KFLOP is programmed is that each axis is given a channel number, 0-7, and each of these channel numbers will be associated with a specific lettered word in G code. For instance, G0 is the G word for rapid movement and the command G0 X5 Y5 Z1 tells the controller to send the machine to the coordinates of [X, Y, Z] = [5, 5, 1], which is easy enough to understand. However, the controller has specific letters to choose from when defining the axis association. So, while most 3D printers use an E word for the extruder by default, the E word is not available in the KFLOP controller, making it incompatible with many of the slicer programs available. This incompatibility could be dealt with by using a post processing program to change any E words in a G code program to one of the letters used by the KFLOP controller. However, KISSlicer has the ability to set the letter for extruder operations without a need for a post processor program, so it is directly compatible with the KMotionCNC software. Additionally, KISSlicer has the option of inserting custom G code segments every N layers. This particular ability is necessary to the photogrammetry process, where a geometric record is desired at intervals throughout the build process. Again, this task could be accomplished through a post processor, but that added step is not necessary with KISSlicer. KISSlicer also opens an incredible number of options and settings to the user, allowing full and exacting control over the slicing operation as is appropriate for a research platform. For these reasons, KISSlicer was chosen as the slicer for this project.

SATISFACTION OF REQUIREMENTS

The KFLOP plus KNozz combination satisfies the requirements for controlling the printer. The easiest to justify are the general requirements supporting future flexibility and adaptability of the controller. The ecosystem of additional add-on boards provides expanded input and output capabilities, and the open software architecture provides an easy method to develop and implement new control routines. These capabilities give confidence that the controller will be able to cope with new demands in the future. The open software also gives the controller the ability to handle the abnormal motion architecture.

The KNozz board provides the means of satisfying the requirement for controlling the heated bed and hot end. The hot end on the E3D Titan Aqua is 12VDC 30W heating element which means that it can sink up to 2.5 amps. This is far below the capabilities of the heating channel, which can handle 20 amps. The heated bed runs on 120VAC at 1100W, so cannot be directly controlled by the KNozz board. However, by connecting the heating channel to the control inputs of a solid-state relay, the heated bed can be controlled using the FET switched heater channel.

The KFLOP board has the speed required to drive the stepper motors to achieve the desired top speed as well as provide the minimum movement increments. The stepper motors have 200 full steps per revolution and are being used with DRV8825 drivers which are configured to provide 1/16 micro-stepping. This means the controller must provide 6400 step signals per motor revolution. The drive pulleys have 20 teeth and the belt pitch on the 2GT belts is 2mm which results in linear motion of 40mm per motor revolution. Thus, the minimum movement the system is theoretically capable of is 40mm / 3200 steps or 12.5 microns per step (0.0005 in). The KFLOP controller can generate step and direction signals at a rate of up to 2.5MHz, so the top speed that the KFLOP can support using this drive pulley and micro-step configuration is 2.5e6 steps per second times 12.5 microns per step or 31,250 mm/s (102.5 ft/s). This exceeds the desired top speed of 200 mm/s (8 in/s).

SENSORS AND DATA RECORDING

RELEVANT REQUIREMENTS

The main point of this platform is to be able to monitor and record the build parameters of the printing process. Ideally, all of these parameters could also be controlled but that is not required at this point in the design. Instead, control is required for print head and bed position and print head and bed temperatures. Unlike most hobby 3D printers, closed loop position control is desired. Monitoring is required of the environment temperature and humidity. The system should be flexible and allow for the addition of other sensors at a later time. Additionally, the system should be able to monitor and record all of this information at a rate of at least 2 times per second and be able to communicate with other portions of the control system. All sensors within the build environment must be able to withstand the enclosure temperature of 100 Celsius.

SENSOR CHOICE AND SPECIFICATIONS

The KFLOP motion controller and its daughter board, the KNozz, have the ability to monitor and control the bed and print head temperature and position as discussed in Chapter 4. Thus, a controller is only needed to take care of monitoring environmental build parameters. The Raspberry Pi 3+ was chosen for this task as it has a 40 pin GPIO header and can handle many different sensors. The Raspberry Pi also has multiple USB ports, a CSI camera port, ethernet and wireless capabilities. While these attributes are not currently needed, they make the Raspberry Pi a flexible platform for coping with future needs.

Environmental temperature monitoring is accomplished through the use of five three-wire PT1000 RTD sensors. These sensors can measure temperatures up to 550 Celsius. Each sensor is attached to a breakout board from Adafruit that uses the MAX31865 RTD sensor amplifier. These boards read the resistance of the PT1000 sensor and output a digital signal in the SPI protocol. The SPI protocol makes it very easy to integrate a large number of these boards to the Raspberry Pi while only using 3 of the GPIO pins. The other environmental sensor is a Sensirion SHT31-D temperature and humidity sensor on a breakout board from Adafruit. The sensor is capable of accuracy of ±2% relative humidity and rated for a temperature range of -40 to 125 Celsius.

Position feedback is accomplished through the use of magnetic linear position sensors. The magnetic tape used for the distance track is elastomer bonded ferrite on a stainless steel carrier substrate with a single pole length of 1mm and accuracy of ± 10 microns per meter. This is used in conjunction with an AMS5311 magnetic position sensor. The AMS5311 from AMS (formerly Austria Microsystems) uses a linear hall effect array and an on-board 12-bit ADC to interpolate distance across a 2mm magnetic pole pair. The chip has both an absolute and incremental interface. The absolute interface can be used with an increment/decrement signal to achieve position resolution of 0.488 microns but would require an external microcontroller to work in this mode for this application. Instead, the incremental mode gives resolution of 1.95 microns and can be read directly by the KFLOP without a need for additional electronics. This is a terrific boon for the project budget, as a demo board is available that breaks out the connections for the IC with the necessary resistors to make the chip function at a price of 17 dollars per chip. 330 mm of magnetic strip and one sensor board per axis for all three axes is a total cost of 264 dollars; far cheaper than purchasing a three-axis, off-the-shelf magnetic scale feedback system. Additionally, if extra resolution is needed in the future, a microprocessor can be added to the system to read the absolute position interface and keep track of the increment/decrement signal and convert these signals to be compatible with the KFLOP. The maximum travelling speed for this sensor is 650 mm/s (25.6 in/s) which is far in excess of the desired travelling speed of 200 mm/s (8 in/s). The chip has an operating temperature range of -55 to 125 Celsius.

Future control of environmental parameters is desired for this platform but has not yet been implemented. Concerning humidity, this platform will be operated in a lab environment in an air-conditioned building so humidity control is not likely to be an issue. If future investigation finds the humidity in the lab environment to fluctuate too much, a solution will be devised. To meet the requirement of mitigating unwanted airflow within the build chamber, a system of heat sinks can be used to control the chamber temperature by pumping heated or cooled water through the heat sinks. This idea has not been prototyped or tested but is a possibility for future implementation.

SATISFACTION OF REQUIREMENTS

The printer has the capabilities asked for in terms of sensors with the exception of the closed loop positioning. The controller has the capability for closed loop control, but the magnetic strips used as part of the linear position sensors are still in the process of being purchased. This is unfortunate as these are necessary sensors for measuring and capturing the position and velocity of the bed and print head. Also, they were the planned sensor for testing positional accuracy of the system. When these parts arrive, the system will be upgraded and tested.

Otherwise, the system has sensor to monitor the build environment's temperature and humidity, the print head temperature, and bed temperature. The print head and bed temperatures are controlled through the Knozz board. These variables can be recorded several times per second using the Raspberry Pi microcontroller. The microcontroller has plenty of GPIO pins for the addition of additional sensors in the future to expand the system capabilities.

PHOTOGRAMMETRY SOFTWARE AND TOOLCHAIN

RELEVANT REQUIREMENTS

A novel aspect of this platform is the desire for a method of geometric feedback. This feedback has a few purposes: it provides a method of recording the geometric output of the printing process, it provides a medium for comparison between samples, and can provide a means of detecting geometric errors in-situ. Error detection and correction through this geometric feedback is a research direction of interest to this lab and this printer will provide a platform for conducting this research. For this feedback to be relevant, it is paramount that it does not disturb the printing process. Also, the process needs to capture all angles of the printed artifact not obscured by the build plate. The controller for this process needs to be able to integrate with the motion controller to facilitate data collection and must be able to control and record from a camera or vision sensor.

Photogrammetry is a complex process with many calculations that benefit from parallelism in computation. The software tool chosen to complete this process ideally takes advantage of this to reduce processing time. While the eventual goal of research into in-situ error detection using this process is the ability to detect errors with ample to time to correct those errors during the build process, at this stage of the research no requirements or limitations are placed on the timeliness of computation. Among other reasons, the author did not have enough knowledge of this space to develop realistic expectations. Thus, the only requirement of the toolchain is that it is able to collect picture data of the specimen and process it into a dense point cloud, but faster computation times are preferred. As with the rest of this project, adaptability to future needs is desired, so open-source tools and equipment are preferable to proprietary options.

TOOLCHAIN AND CONTROLLER SPECIFICATIONS

This process starts with the selection of the software used to perform the photogrammetry. There are several excellent programs for this application, but since open-source programs are preferred, we considered MicMac and AliceVision's Meshroom. MicMac is a command line program developed by the French National Geographic Institute and the French National School for Geographic Sciences. MicMac uses only a CPU for calculations during the photogrammetry process which means that a GPU is not needed, opening up a large choice of controllers. However, the biggest problem with MicMac is that it is not optimized to take advantage of parallelism in the photogrammetry computations and only uses a single CPU core. Due to this, dense point cloud computations can take a long time.

Meshroom is a GUI built in Python to utilize the AliceVision framework for photogrammetry. Meshroom is a multithreaded process that uses a GPU to perform the photogrammetry calculations that benefit from parallelism. However, the downside to this is that a CUDA enable Nvidia GPU is required for computation. This drastically reduces the number of controllers that can be used for this system. It does significantly speed up the computation time and for this reason Meshroom was chosen as the software of choice. Ultimately, as will be explained shortly, the Meshroom GUI was not used and the AliceVision framework was utilized with a Python batch script instead.

The controller chosen for this project was a Nvidia Jetson TX2. This board is an edge computing platform that contains a dual core Denver CPU, a quad core ARM A57 CPU, a 256 core GPU, and 8Mb of RAM. While this is an amazing amount of computing power in a small package, it is not without its drawbacks. One of the biggest problems with using unusual architectures is that precompiled programs are not often available and must instead be compiled from the source code. The AliceVision framework is dependent on several other toolchains such as OpenCV, OpenImageIO, Boost, Geogram, and others which made this compilation process rather painful. Once all of these dependencies and AliceVision were built on the TX2, the 32 GB of on-board emmc memory was nearly full. Compiling Meshroom required compiling QT as a dependency, which the author was unable to accomplish. However, a batch script was written in Python to access the relevant functions of the AliceVision framework to perform photogrammetry and process a set of photographs into a dense point cloud. This method is preferable for automation since it does not need the overhead of a GUI and is a better long-term solution for this platform.

The Jetson TX2 controls two USB web cameras to capture data for the photogrammetry process. These cameras are from Arducam and feature the 2MP AR0230 CMOS sensor. These sensors have a pixel size of 3.3 microns x 3.3 microns and a dynamic range of 105 dB. One camera is placed in the top front corner of the build chamber and the other camera placed below it near the lower limit of the print bed travel. When the bed rotation mechanism discussed in Chapter 2 is engaged, these cameras are used to capture photographs of the printed artifact. These photographs are then processed by the Jetson TX2 into a dense point cloud which is saved for future analysis.

Satisfaction of Requirements

The scanning process was implemented in the KMotionCNC by writing a C program and linking it to the M102 word. A P parameter is passed with the M word to determine the number of positions from which pictures should be taken. For example, M102 P6 will perform the scanning process and take two pictures (one from each USB camera) at 6 intervals of 60 degrees so that the entire perimeter of the part is captured, resulting in 12 pictures with which to complete a photogrammetry reconstruction. The C program for this process is included in Appendix A. To test the procedure, a model of a Benchy was sliced in KISSlicer with the command M102 P6 inserted every 100 layers and after the end of the printing process. Figure 4.7 and Figure 4.9 below show the 12 pictures captured from the scan at layer 100 and Figure 4.8 and Figure 4.10 show the pictures captured from the scan at the end of the print.

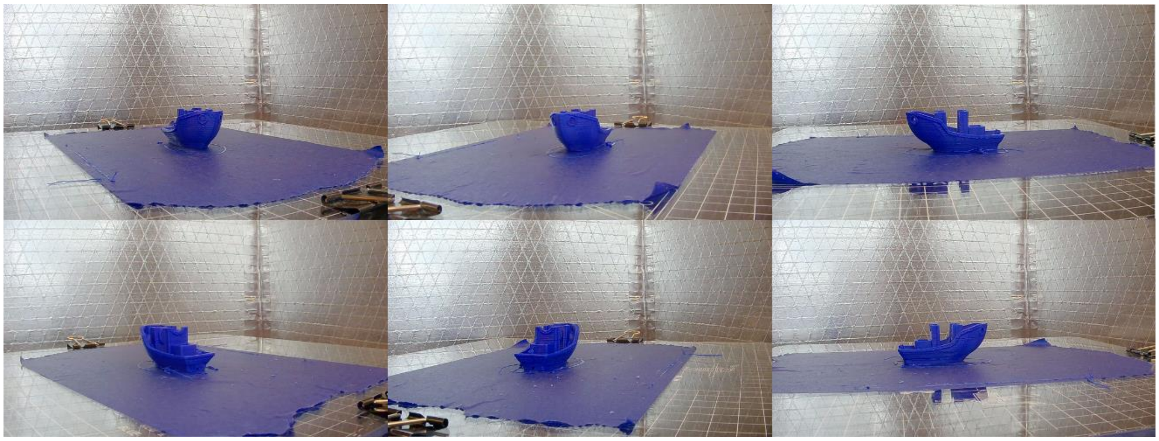


Figure 4.7 Six pictures of a Benchy captured from the lower USB camera during a scan at layer 100 of the printing process.

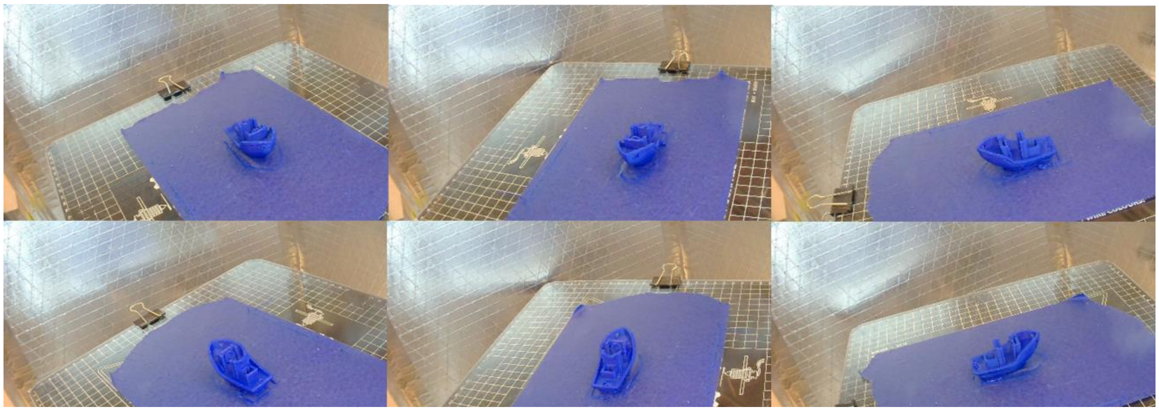


Figure 4.9 Six pictures of a Benchy captured from the upper USB camera during a scan at layer 100 of the printing process.

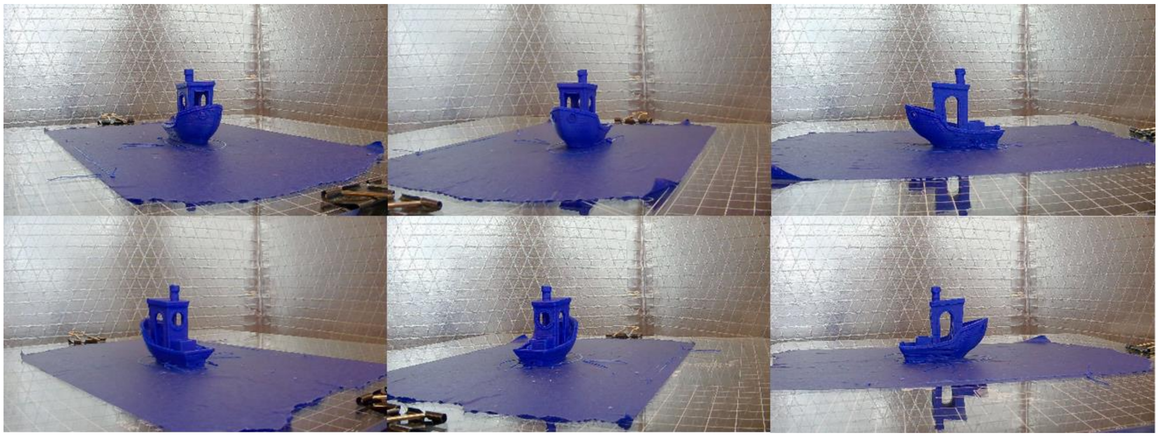


Figure 4.8 Six pictures of a Benchy captured from the lower USB camera during a scan after completion of the printing process.

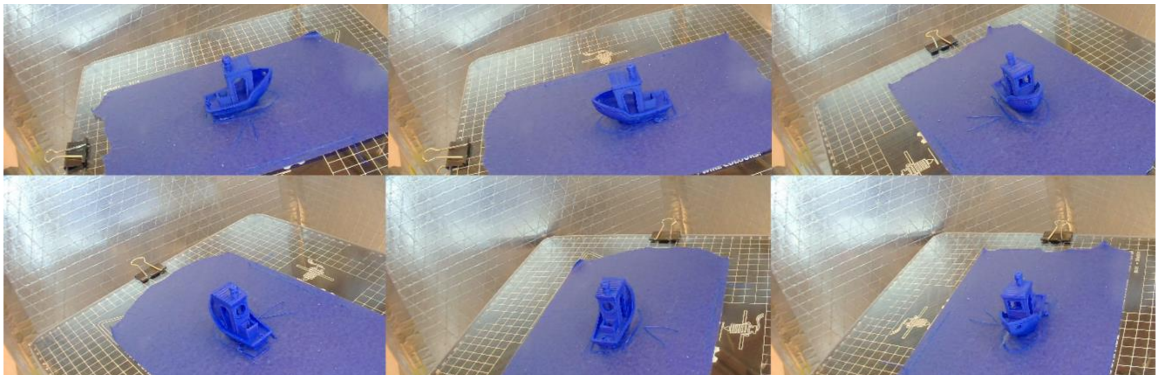


Figure 4.10 Six pictures of a Benchy captured from the upper USB camera during a scan after completion of the printing process.

CUSTOM SOFTWARE AND SOFTWARE MODIFICATIONS

One of the most difficult parts of mechatronic system design is integrating different pieces of hardware to work together harmoniously. For this project, extensive software setup was required to make the various systems communicate and function as desired. The control software was setup to reflect the physical dependencies between the control signals and the physical movement and configuration of the hardware and modified to allow for the chosen movement architecture. M-code words were programmed in C to perform specific functions like homing, bed and nozzle temperature control, data capture for photogrammetry, and data synchronization and capture across the different sensor platforms. Python programs were written to monitor and record the temperature and humidity sensors and perform the photogrammetry process. The python and C code discussed in this section can be found in Appendix A.

In order for the KFLOP controller to be able to send the correct command signals for XY movement, a new kinematic class had to be added to the KMotionCNC software. The class is written in C and allows the software to calculate the inverse kinematics representation of the equations found in Chapter 2. A few small changes had to be made to other functions in the KMotionCNC software to support this new class. All code changes are highlighted in Appendix A.

A significant consideration when choosing the KFLOP controller and its accompanying software, KMotionCNC, was ease of creating custom commands and control algorithms. KMotionCNC makes this incredibly easy by providing a setup screen to assign actions to various M-code words such as executing a C program. Since the motion and control C libraries for the board can be used in any C program by including the single line ‘#include “KMotionDef.h”’, just about anything can be programmed into G code with a simple M word.

The M100 and M101 words are used to control the temperature of the heated bed and the hot end. These words must be passed with P and Q parameters that correspond to bed and hot end temperatures respectively. M100 is used to begin a temperature change and proceed without waiting for the change to take place, and the M101 word is used to initiate a temperature change and wait on it to take effect before continuing. For example, M100 P100 Q200 will set the bed temperature to 100 Celsius and the hot end temperature to 200 Celsius and will continue to execute G code. The command M101 P100 Q200 will set the temperatures as with the M100 command but will monitor them and wait until both temperatures are reached before continuing to execute G code.

M105 will be used to home the machine and set soft limits for the range of motion so that no crashes can occur, however it cannot be implemented until the linear position sensors and limit switches are installed. This code will start by disabling the soft limits instituted in the control software, and then moving each axis in a specified direction until a limit switch is tripped. The machine will be equipped with linear positions sensors and the movement of each axis continues until the machine is directly over the next magnetic pole on the position sensor. Then each axis reverses direction and moves 4 mm. This position is set as the minimum soft limit. Each axis moves another 2mm and this is set as the machine origin. The axes continue until they hit the limit switches at the other end of travel. They continue until they are directly over the next magnetic pole and then back up

4 mm (2 magnetic poles). This position is then recorded as the upper soft limit. This homing sequence should be very repeatable because it will use the encoders to set the home positions and not mechanical switches.

The M102 word is used to perform the photogrammetry data capture process. As mentioned previously, the slicer program for this tool chain is KISSlicer. KISSlicer allows an incredible amount of control over the slicing parameters and G code generation. One very important feature is the ability to insert custom G code every N layers. For this project, this means that every N layers the M102 word can be easily inserted into the G code to perform the photogrammetry data capture and take a record of the geometry of the printed object. The M word is passed with a P parameter that tells the printer how many positions to take pictures from during one full rotation of the bed. For instance, M102 P20 will cause the printer to take two pictures, one from each USB web camera, from 20 equally spaced positions around the object. The code for this causes the Z axis to go to its bottom position where the three vee blocks on the print bed have disengaged from the ball bearings on the motion stage. At this point, the bed is supported through the outer race of its large bearing by the three pillars mounted in the bottom of the build space and the stepper motor for controlling bed rotation is engaged with the nut on the bottom of the bed through its custom shaft and socket. The picture taking for this process is currently done manually but will be implemented to take place automatically in the future. When the KFLOP is ready to take a picture, a pop-up window is displayed on the control computer indicating that the bed is in position. The user captures a photograph from both USB cameras and then hits the “OK” button on the pop-up window and

KFLOP will move the bed to the next position. Once a full rotation has been completed in this manner, the KFLOP reverses the direction of motion and returns the bed to its original position. The Z stage moves upwards, securing the print bed and then continuing to the previous Z position to reinitiate the printing process.

This concludes the discussion of the printer's design and capabilities. The following chapter will first talk about the overall satisfaction of system requirements as well as how the system can be modified to meet any unmet requirements in the future. A discussion of possible future design improvements with possible solutions to some of the problems encountered during the current design process is also presented. The chapter finishes with lessons learned from the design of this test bed.

CHAPTER FIVE

CONCLUSION

UNSATISFIED REQUIREMENTS

While the majority of the design constraints for this project were met, there were a few areas where the test bed does not meet the original design requirements. The only requirement that was truly not met is the specification for the maximum temperature of the build environment. The requirement was a minimum of 100 Celsius. However, the magnetic strips used for the linear position sensors and the extruder are only capable of 80 Celsius. While this is unfortunate, the magnetic strip is a very specialized part with few suppliers, and we were unable to find a part that would meet our design requirements. Glass scales were also looked at, but most of the products on the market are only rated for 45 Celsius, have considerably more bulk than magnetic scale systems which would be harder to fit in the machine, and are several times more expensive than the magnetic scale system chosen. Currently, closed loop feedback has not been implemented on the printer because the magnetic scales have taken a very long time to procure. They will be implemented in the near future but are not on the machine at the time of this report.

DESIGN FOR FUTURE IMPROVEMENTS

The test bed is a complex system with many design features and subsystems that all require time to design, fabricate, and integrate into the overall system. Therefore, there are some improvements and additional subsystems that are part of the long-term goals of the project that were unable to be accomplished in the timeframe allotted for this thesis. First, we would like to achieve the ability to print high temperature polymers as mentioned in the design statement. Specifically, the only modification needed to do this is an upgrade of the hot end. The current configuration is only capable of 285 Celsius but temperatures greater than 350 Celsius are needed. Earlier in the section on the XY Motion System we stated that the Titan Aqua extruder is capable of 450 Celsius which is part of why it was chosen. However, in its stock configuration, the Titan Aqua has a brass nozzle, aluminum heater block, and a thermistor temperature sensor, none of which are adequate for temperatures above 285 Celsius. The aluminum heater block begins to soften before the desired temperature, the thermistor will likely burn, and the brass nozzle experiences significant wear as it also softens before the desired temperature. Additionally, there is a PTFE tube in the aluminum heater block underneath the heat break to help constrain the filament path that must be removed to print above 285 Celsius. These parts can be replaced with off the shelf parts from E3D that will allow the Titan Aqua to print at temperatures up to 500 Celsius. These parts are a titanium heat break, copper heater block, high temperature resistive heater, a PT100 RTD sensor, and a hardened tool steel nozzle. The 12VDC power supply was sized correctly for the high temperature heater cartridge, which is 70W versus the stock cartridge of 30W, so the additional power requirements should not be an issue. The PT100 will be the hardest part to integrate because it will require bypassing or replacing some resistors on the KNozz board. These resistors are being used with the thermistor that is currently being used to monitor the print head temperature but are unnecessary for the RTD sensor. This integration needs to be figured out and confirmed with Dynomotion before moving forward with purchasing the high temperature components.

Next, we would like to address the unmet requirement of being able to sustain operations with an environment temperature of 100 Celsius. The Titan Aqua has two plastic pieces that likely need to be replaced to operate at this temperature. If the plastic extruder parts can be replaced and the temperature limitations of the magnetic strips can be circumvented, then the next limiting component has a working temperature of 125 Celsius which is well in excess of the desired temperature for the build environment. One solution would be to cool the strips directly, by placing them on a liquid-cooled substrate. However, this subsystem would need temperature sensing and control of its own to ensure the strips do not exceed their maximum temperature. We think this subsystem would be easily integrated with the Raspberry Pi. Adding this capability would also make the machine more accurate, as the measurement system would be operated at a constant temperature and therefore not subjected to thermal growth.

Currently, the environmental temperature is monitored but not controlled. This needs to change in order for this platform to truly be a controlled test bed. To meet the requirement of reducing errant airflow, the control system needs to use a principle other than air exchange to manage the environment temperature. One solution is to use the extra space in the bottom and corners of the build enclosure to implement a series of heat sinks with internal fluid channels. The internal temperature of the environment can then be controlled by controlling the temperature of the fluid being pumped through the heat sinks. This method would need to be prototyped and tested before integration with the system. Similar to the cooling system for the magnetic strips, this system could be integrated with the Raspberry Pi as the controller.

Additionally, the platform does not currently have a method for direct cooling of the printed parts. Since the desire is to mitigate as much airflow as possible, a fan that pulls air from the environment and blows it onto the part will promote unacceptable air movement. Instead, a compressed air line will be run to a duct around the nozzle. The compressed air will be controlled by an air regulator and solenoid. This will allow for predictable, repeatable control of the airflow volume. Since this will introduce room temperature air to the build volume, it is best that this is not experimented with until a method of controlling the environmental temperature has been implemented.

There are a few additional changes and optimizations to be made that involve the coding for the platform. First, the temperature control for the bed and print head is not as uniform as desired. It currently fluctuates 3-4 degrees during operation. The KFLOP controller has methods of applying different control routines such as PID to the KNozz signal for these components, but they have yet to be implemented and tested on this platform.

The photogrammetry parameters need to be optimized. There are a large number of settings within the AliceVision framework that can be manipulated to produce better models from the process, and many of them drastically affect the amount of processing time it takes to turn a batch of photos into a dense point cloud. If the point cloud is to be used to provide feedback during the printing process, the photo processing time needs to be reduced to a reasonable amount of time. When the Jetson TX2 was chosen for this project, we felt that it represented a good amount of processing power for the task and was cheaper than other options, such as a desktop computer with graphics card. However, during testing we realized that the amount of memory available on the system is a bottle neck for processing because the system usually runs out of memory when parallelizing some of the computation steps before it runs out of processing power. In this respect, the 8GB of memory available on the Jetson TX2 is a hinderance to processing speed. As an embedded platform, this memory is not upgradeable which presents a problem for trying to mitigate this issue in the future. Additionally, the Jetson TX2 is an edge computing platform and figuring out how to compile the AliceVision framework and working through all of the versioning problems with its dependent software packages was very time consuming. Having completed this process, we think that it would be better to use the computer that runs the KFLOP controller to do the photogrammetry processing. There is a Linux port of the KMotionCNC program and one of our hopes at the outset of this project was to use this port to be able to run the KFLOP from the Jetson TX2 without the need for an external computer. In retrospect, it would have been easier but slightly more expensive to specify a decent desktop computer with a graphics card and plenty of memory instead of trying to use the Jetson TX2. We are currently evaluating if the best route moving forward is to keep the Jetson TX2 or transition the photogrammetry operations to a desktop computer. This would require an investment in new computer hardware but may open up more freedom in manipulating and using the photogrammetry process for future experiments.

The platform currently meets the requirements set out for motion capabilities in terms of speed and acceleration. However, during testing it was discovered that the printer cannot go above its current speed setting without losing steps and this is due to the 12VDC power supply that is used to power the stepper motors. The motor drivers can handle up to 45VDC. Increasing the power supply from 12VDC to a higher voltage would improve the torque characteristics at higher motor speeds, which is what is currently limiting the platform. This could enable experimenting with the capabilities of very high speed FDM printing.

LESSONS LEARNED

Several excellent lessons and many skills were learned during the design and fabrication of this test bed. One of the biggest practical lessons was to not allow design decisions that had already been made to restrict the options for current design decisions. Instead, options that are incompatible with current design decisions have an addition cost to implement them where the previous design decision must be changed. This usually incurs both time and monetary costs but may have a larger benefit than the options that are compatible with previous design decisions. For example, the Jetson TX2 seemed like an excellent choice of hardware for its intended task at the time the decision was made to purchase it. However, after evaluation, trouble with finding suitable cameras, and getting AliceVision to compile, it appears that sticking with the Jetson TX2 moving forward may limit the capabilities of the system in terms of performance and ease of use.

Another excellent lesson deals with the design of subassemblies. This printer had numerous elements that needed precise alignment to work properly. Building each of these assemblies so that they can be assembled separately and then aligned as an entire subassembly on the machine dramatically reduces the tediousness and the amount of effort needed for precise construction of the system. For instance, the XY assembly was created in three different parts: a left and right rail section for the Y, and the gantry rail for the X. In order to have precise motion from the CoreXY architecture, it is imperative that the belt paths that move the Y carriages and the print head are parallel to the direction of motion. See the figure below for an illustration of the critical paths. This means that the motor mounts and the pulley mounts must be aligned with the Y linear rails, but the Y linear rails also need to be aligned to each other. These Y rails were split into two subassemblies, each one consisting of a plate upon which a linear rail was mounted. The motor and pulley mounts were placed on the plate and aligned to the linear rail. This way one plate was mounted to the printer frame as the reference and the second plate was then mounted. The entire plate was moved to align the rails to each other, keeping the relative alignment of rail and belt paths fixed during the alignment process. The gantry was then mounted between the two carriages and easily aligned with the help of a machinist square. While the XY subassemblies were done well, the Z was designed very poorly in this respect. Note that the Z motion components are not tied together; that is to say that there is no way to adjust a single item and adjust the entire assembly. Instead, the linear rail had to be aligned to be perpendicular to the XY motion plane, and then the lead screw had to be offset and aligned parallel to the Z linear rail. This process was difficult and tedious as a result. Whenever adjustments need to be made, both rail and lead screw will have to be moved separately. Because the XY was so easy to align, the Y rails are parallel to within 12.7 microns (0.0005 in) and they are square to the X rail within 12.7 microns (0.0005 in) over their respective travel distances. However, even after significant time spent aligning the Z axis, it is only square to the XY plane to within 50 microns (0.002 in) over the travel distance.

Overall, the project meets the design goals set out at the beginning of this process. A flexible 3D printer was created that can be used as a test bed for research into polymer FDM processes and algorithms. The test bed will soon have the capability to print all current 3D printable polymers including the demanding high temperature polymers like PEEK and Ultem. The platform has sensors for capturing and recording relevant build parameters like environmental conditions, bed and hot end temperature, and print head position. Additionally, the platform includes a novel mechanism for in-situ capturing of geometric part data using a photogrammetry process. The flexibility and adaptability of the platform should make it an excellent research tool.

APPENDICES

Appendix A

Computer Programs and Code Modifications

KMotionCNC Initialization Program

```
#include "KMotionDef.h"

// A KMotionCNC MCode can be configured to set the temperature setpoints
// using P and Q parameters. ie.
// M100 P100 Q50 (Set Nozzle and Bed temperature setpoints)

#define AUX 0 // 0=Aux0 1=Aux1  defines which Aux connector to use.

// define pair of persist variables (as floats) as heater setpoints
#define NOZ_VAR 30
#define BED_VAR (NOZ_VAR+1)
#define NOZ_CURRENT (NOZ_VAR+2)
#define BED_CURRENT (NOZ_VAR+3)

//SPI and Heater IO definitions
#define CS (23 + AUX*10)
#define DATAIN (24 + AUX*10)  // with respect to KFLOP
#define CLK (25 + AUX*10)
#define DOUT (22 + AUX*10)  // with respect to KFLOP
#define NOZ_HEAT (21 + AUX*10)
#define BED_HEAT (20 + AUX*10)

int SPI_IN(int send_data); // function to read serial SPI dual ADC
float TempToADC(float T); // function to convert Temp C to ADC counts
float ADCtoTemp(float At);// Solve inverse function numerically using guesses and
linear interpolate
void ServiceKNozz(void); // Service KNozz Temperature controls

float *NozSetPoint = (float *)&persist.UserData[NOZ_VAR]; // define convienient
pointers to Persist floats
float *BedSetPoint = (float *)&persist.UserData[BED_VAR];
float *NozTemp = (float *)&persist.UserData[NOZ_CURRENT];
float *BedTemp = (float *)&persist.UserData[BED_CURRENT];

int main()
{
```

```
ch0->InputMode=NO_INPUT_MODE;
ch0->OutputMode=STEP_DIR_MODE;
ch0->Vel=16932;
ch0->Accel=711200;
ch0->Jerk=10000000;
ch0->P=1;
ch0->I=0;
ch0->D=0;
ch0->FFAccel=0;
ch0->FFVel=0;
ch0->MaxI=200;
ch0->MaxErr=1000;
ch0->MaxOutput=200;
ch0->DeadBandGain=1;
ch0->DeadBandRange=0;
ch0->InputChan0=0;
ch0->InputChan1=0;
ch0->OutputChan0=8;
ch0->OutputChan1=1;
ch0->MasterAxis=-1;
ch0->LimitSwitchOptions=0x110;
ch0->LimitSwitchNegBit=19;
ch0->LimitSwitchPosBit=20;
ch0->SoftLimitPos=1e+30;
ch0->SoftLimitNeg=-1e+30;
ch0->InputGain0=1;
ch0->InputGain1=1;
ch0->InputOffset0=0;
ch0->InputOffset1=0;
ch0->OutputGain=1;
ch0->OutputOffset=0;
ch0->SlaveGain=1;
ch0->BacklashMode=BACKLASH_OFF;
ch0->BacklashAmount=0;
ch0->BacklashRate=0;
ch0->invDistPerCycle=1;
ch0->Lead=0;
ch0->MaxFollowingError=1000;
ch0->StepperAmplitude=20;

ch0->iir[0].B0=1;
ch0->iir[0].B1=0;
ch0->iir[0].B2=0;
ch0->iir[0].A1=0;
```

```
ch0->iir[0].A2=0;

ch0->iir[1].B0=1;
ch0->iir[1].B1=0;
ch0->iir[1].B2=0;
ch0->iir[1].A1=0;
ch0->iir[1].A2=0;

ch0->iir[2].B0=1;
ch0->iir[2].B1=0;
ch0->iir[2].B2=0;
ch0->iir[2].A1=0;
ch0->iir[2].A2=0;


ch1->InputMode=NO_INPUT_MODE;
ch1->OutputMode=STEP_DIR_MODE;
ch1->Vel=16932;
ch1->Accel=711200;
ch1->Jerk=10000000;
ch1->P=1;
ch1->I=0;
ch1->D=0;
ch1->FFAccel=0;
ch1->FFVel=0;
ch1->MaxI=200;
ch1->MaxErr=1e+06;
ch1->MaxOutput=200;
ch1->DeadBandGain=1;
ch1->DeadBandRange=0;
ch1->InputChan0=1;
ch1->InputChan1=1;
ch1->OutputChan0=9;
ch1->OutputChan1=3;
ch1->MasterAxis=-1;
ch1->LimitSwitchOptions=0x110;
ch1->LimitSwitchNegBit=0;
ch1->LimitSwitchPosBit=0;
ch1->SoftLimitPos=1e+30;
ch1->SoftLimitNeg=-1e+30;
ch1->InputGain0=1;
ch1->InputGain1=1;
ch1->InputOffset0=0;
ch1->InputOffset1=0;
```

```
ch1->OutputGain=1;
ch1->OutputOffset=0;
ch1->SlaveGain=1;
ch1->BacklashMode=BACKLASH_OFF;
ch1->BacklashAmount=7;
ch1->BacklashRate=4000;
ch1->invDistPerCycle=4;
ch1->Lead=0;
ch1->MaxFollowingError=1000;
ch1->StepperAmplitude=20;

ch1->iir[0].B0=1;
ch1->iir[0].B1=0;
ch1->iir[0].B2=0;
ch1->iir[0].A1=0;
ch1->iir[0].A2=0;

ch1->iir[1].B0=1;
ch1->iir[1].B1=0;
ch1->iir[1].B2=0;
ch1->iir[1].A1=0;
ch1->iir[1].A2=0;

ch1->iir[2].B0=1;
ch1->iir[2].B1=0;
ch1->iir[2].B2=0;
ch1->iir[2].A1=0;
ch1->iir[2].A2=0;


ch2->InputMode=NO_INPUT_MODE;
ch2->OutputMode=STEP_DIR_MODE;
ch2->Vel=20320;
ch2->Accel=304800;
ch2->Jerk=3048000;
ch2->P=1;
ch2->I=0;
ch2->D=0;
ch2->FFAccel=0;
ch2->FFVel=0;
ch2->MaxI=200;
ch2->MaxErr=1e+06;
ch2->MaxOutput=200;
ch2->DeadBandGain=1;
```

```
ch2->DeadBandRange=0;
ch2->InputChan0=2;
ch2->InputChan1=2;
ch2->OutputChan0=10;
ch2->OutputChan1=5;
ch2->MasterAxis=-1;
ch2->LimitSwitchOptions=0x110;
ch2->LimitSwitchNegBit=0;
ch2->LimitSwitchPosBit=0;
ch2->SoftLimitPos=1e+30;
ch2->SoftLimitNeg=-1e+30;
ch2->InputGain0=1;
ch2->InputGain1=1;
ch2->InputOffset0=0;
ch2->InputOffset1=0;
ch2->OutputGain=1;
ch2->OutputOffset=0;
ch2->SlaveGain=1;
ch2->BacklashMode=BACKLASH_OFF;
ch2->BacklashAmount=0;
ch2->BacklashRate=8200;
ch2->invDistPerCycle=1;
ch2->Lead=0;
ch2->MaxFollowingError=1000;
ch2->StepperAmplitude=20;

ch2->iir[0].B0=1;
ch2->iir[0].B1=0;
ch2->iir[0].B2=0;
ch2->iir[0].A1=0;
ch2->iir[0].A2=0;

ch2->iir[1].B0=1;
ch2->iir[1].B1=0;
ch2->iir[1].B2=0;
ch2->iir[1].A1=0;
ch2->iir[1].A2=0;

ch2->iir[2].B0=1;
ch2->iir[2].B1=0;
ch2->iir[2].B2=0;
ch2->iir[2].A1=0;
ch2->iir[2].A2=0;
```

```
ch3->InputMode=NO_INPUT_MODE;
ch3->OutputMode=STEP_DIR_MODE;
ch3->Vel=40000;
ch3->Accel=400000;
ch3->Jerk=4e+06;
ch3->P=1;
ch3->I=0;
ch3->D=0;
ch3->FFAccel=0;
ch3->FFVel=0;
ch3->MaxI=200;
ch3->MaxErr=1e+06;
ch3->MaxOutput=200;
ch3->DeadBandGain=1;
ch3->DeadBandRange=0;
ch3->InputChan0=3;
ch3->InputChan1=0;
ch3->OutputChan0=11;
ch3->OutputChan1=0;
ch3->MasterAxis=-1;
ch3->LimitSwitchOptions=0x110;
ch3->LimitSwitchNegBit=0;
ch3->LimitSwitchPosBit=0;
ch3->SoftLimitPos=1e+30;
ch3->SoftLimitNeg=-1e+30;
ch3->InputGain0=1;
ch3->InputGain1=1;
ch3->InputOffset0=0;
ch3->InputOffset1=0;
ch3->OutputGain=-1;
ch3->OutputOffset=0;
ch3->SlaveGain=1;
ch3->BacklashMode=BACKLASH_OFF;
ch3->BacklashAmount=0;
ch3->BacklashRate=0;
ch3->invDistPerCycle=1;
ch3->Lead=0;
ch3->MaxFollowingError=10000;
ch3->StepperAmplitude=20;

ch3->iir[0].B0=1;
ch3->iir[0].B1=0;
ch3->iir[0].B2=0;
```

```
ch3->iir[0].A1=0;
ch3->iir[0].A2=0;

ch3->iir[1].B0=1;
ch3->iir[1].B1=0;
ch3->iir[1].B2=0;
ch3->iir[1].A1=0;
ch3->iir[1].A2=0;

ch3->iir[2].B0=1;
ch3->iir[2].B1=0;
ch3->iir[2].B2=0;
ch3->iir[2].A1=0;
ch3->iir[2].A2=0;

ch4->InputMode=NO_INPUT_MODE;
ch4->OutputMode=STEP_DIR_MODE;
ch4->Vel=5420;;
ch4->Accel=400000;
ch4->Jerk=4000000;
ch4->P=1;
ch4->I=0;
ch4->D=0;
ch4->FFAccel=0;
ch4->FFVel=0;
ch4->MaxI=200;
ch4->MaxErr=200;
ch4->MaxOutput=200;
ch4->DeadBandGain=1;
ch4->DeadBandRange=0;
ch4->InputChan0=4;
ch4->InputChan1=0;
ch4->OutputChan0=12;
ch4->OutputChan1=0;
ch4->MasterAxis=-1;
ch4->LimitSwitchOptions=0x100;
ch4->LimitSwitchNegBit=0;
ch4->LimitSwitchPosBit=0;
ch4->SoftLimitPos=1e+09;
ch4->SoftLimitNeg=-1e+09;
ch4->InputGain0=1;
ch4->InputGain1=1;
ch4->InputOffset0=0;
ch4->InputOffset1=0;
```

```
 ch4->OutputGain=1;
 ch4->OutputOffset=0;
 ch4->SlaveGain=1;
 ch4->BacklashMode=BACKLASH_OFF;
 ch4->BacklashAmount=0;
 ch4->BacklashRate=0;
 ch4->invDistPerCycle=1;
 ch4->Lead=0;
 ch4->MaxFollowingError=10000000;
 ch4->StepperAmplitude=250;

 ch4->iir[0].B0=1;
 ch4->iir[0].B1=0;
 ch4->iir[0].B2=0;
 ch4->iir[0].A1=0;
 ch4->iir[0].A2=0;

 ch4->iir[1].B0=1;
 ch4->iir[1].B1=0;
 ch4->iir[1].B2=0;
 ch4->iir[1].A1=0;
 ch4->iir[1].A2=0;

 ch4->iir[2].B0=1;
 ch4->iir[2].B1=0;
 ch4->iir[2].B2=0;
 ch4->iir[2].A1=0;
 ch4->iir[2].A2=0;



 EnableAxis(0);
 EnableAxis(1);
 EnableAxis(2);
 EnableAxis(3);
 EnableAxis(4);
 DefineCoordSystem6(1,0,2,3,4,-1);
 //DefineCoordSystem(1,0,2,-1);

 SetBitDirection(CS, 1);
SetBitDirection(CLK, 1);
SetBitDirection(NOZ_HEAT, 1);
SetBitDirection(DOUT, 1);
SetBitDirection(BED_HEAT, 1);
```

```
   *NozSetPoint = 0;  // start with heaters off
   *BedSetPoint = 0;
   *NozTemp = 0;
   *BedTemp = 0;

   for (;;)
   {
      Delay_sec(0.001);  // loop ~every millisecond
      ServiceKNozz();
   }

   return 0;
}


// Service KNozz Temperature controls
void ServiceKNozz(void)
{
   static int JobWasActive = FALSE;
   static int i = 0;

   int raw_counts_bed = SPI_IN(0xf000);  // Read ADCs
   int raw_counts_noz = SPI_IN(0xd000);

     float NTempFloat = ADCtoTemp(raw_counts_noz);
     float BTempFloat = ADCtoTemp(raw_counts_bed);

     persist.UserData[NOZ_CURRENT] = *(int *) & NTempFloat;
     persist.UserData[BED_CURRENT] = *(int *) & BTempFloat;

   if (i++ >= 5000) // diagnostic printout ~ every 1 sec
   {
      i = 0;
      printf("Nozz:Setpt %6.1fC %6.1fcnts Actual %6.1fC %4dcnts   Bed:Setpt %6.1fC
%6.1fcnts Actual %6.1fC %4dcnts\n",
         *NozSetPoint, TempToADC(*NozSetPoint), ADCtoTemp(raw_counts_noz),
raw_counts_noz,
         *BedSetPoint, TempToADC(*BedSetPoint), ADCtoTemp(raw_counts_bed),
raw_counts_bed);
   }

   if (raw_counts_noz < TempToADC(*NozSetPoint))
      SetBit(NOZ_HEAT);
   else
      ClearBit(NOZ_HEAT);

   if (raw_counts_bed < TempToADC(*BedSetPoint))
      SetBit(BED_HEAT);
   else
      ClearBit(BED_HEAT);


   if (JobWasActive && !JOB_ACTIVE)  // Job Stopped?
   {
      //                *NozSetPoint=0;  // yes, turn off heater?
      //                *BedSetPoint=0;
   }

   JobWasActive = JOB_ACTIVE;
}

void Dly(void)
{
   Delay_sec(5e-6);
}

int SPI_IN(int send_data)
{
   int i;
   int dataIn = 0;

   SetBit(CS); //CS high
   Dly();
   ClearBit(CLK); //CLK low
   Dly();
   ClearBit(CS); //CS low
   SetStateBit(DOUT, (send_data >> 15) & 1);
   Dly();
   for (i = 0; i < 16; i++)
   {
      SetBit(CLK); //CLK high
      Dly();
      dataIn = (dataIn << 1) | ReadBit(DATAIN);  // read the bit
      ClearBit(CLK); //CLK low
      SetStateBit(DOUT, (send_data >> (14 - i)) & 1);
      Dly();
   }
   SetBit(CS); //CS high
   Dly();

   return dataIn;
}


// function to convert Temp C to ADC counts
float TempToADC(float T)
{

      return (((0.0008855153784 - 0.000000824412405 * T) * T - 0.3469344589) * T +
59.42378307)*T - 1767.00037;  // 3rd order polynomial
   //return T;

}


// Solve inverse function numerically using guesses and linear interpolate
float ADCtoTemp(float At)
{
   int i;
   float A, T, T0 = 40.0, T1 = 100.0;  // initial guess 0 snd 1

   float A0 = TempToADC(T0);  // see how well they did
   float A1 = TempToADC(T1);


   for (i = 0; i < 10; i++)
   {
      // linearly interpolate
      T = T0 + (T1 - T0) * (At - A0) / (A1 - A0);

      A = TempToADC(T); // check how well it works
//          printf("Desired ADC %f guess Temp %f ADC %f\n",At,T,A);

      if (fast_fabs(A - At) < 0.1f) break;  // good result exit

      // replace furthest away guess with new result
      if (fast_fabs(A - A0) > fast_fabs(A - A1))
      {
         T0 = T;// replace guess #0
         A0 = A;
      }
```

```
        else
        {
           T1 = T;// replace guess #1
           A1 = A;
        }
     }
     return T;
  }
```

C program for Photograph Capture; M102

```
#include "KMotionDef.h"
#define TMP 10
#include "KflopToKMotionCNCFunctions.c"

#define Zaxis 2
#define BedAxis 4
#define NEXT 1
#define LAST 2
#define BedCountsPerDegree 168.88889
#define ZCountsPerInch 10160
#define PICS_VAR 50

// Define number of pictures, counts per rotation move

float *NumPictures = (float *)&persist.UserData[PICS_VAR];

int CommunicateTX2(int flagLen);

main()
{
    int numPics = (int) *NumPictures;
    if(numPics<1)
    {
        numPics = 1;
    }
    if(numPics>120)
    {
        MsgBox("Cannot take more than 120 pictures",MB_OK);
        numPics = 1;
    }
    //Calculate the number of counts per move between pictures
    int cntsMove = (int)(360 / numPics * BedCountsPerDegree);
    printf("\nCountsPerMove: %i\tNumPictures: %i\n", cntsMove, numPics);

    // Remember original position
    double OrigX, OrigY, OrigZ;

    OrigX = ch1->Dest;
    OrigY = ch0->Dest;
    OrigZ = ch2->Dest;
    printf("\nX: %i\tY: %i\tZ: %i\n", OrigX, OrigY, OrigZ);
    //Check that Bed Axis is at Zero

    //Move bed to scanning position
    Move(2, 0); // Move machine to just sit on the pillar supports (don't slam kinematic
coupling)
    while(!CheckDone(2));
    MoveAtVel(2, 0.3 * ZCountsPerInch, 0.5 * ZCountsPerInch);
    while(!CheckDone(2));

    //Take first set of pictures here
    printf("\nCountsPerMove: %i\tNumPictures: %i\n", cntsMove, numPics);
    while(!CommunicateTX2(numPics));
    int a;
    for( a = numPics-1; a > 0; a--)
    {
        MoveRelAtVel(4, -cntsMove, BedCountsPerDegree*25); // Move to Next
Position
        while(!CheckDone(4));
        Delay_sec(0.3);   // Give time to dampen motion
        //Signal TX2 to take a picture
        while(!CommunicateTX2(a));
    }

    // Return to original position
    Move(4,0.5*BedCountsPerDegree);
    while(!CheckDone(4));
    Delay_sec(1);
    MoveAtVel(2, 0, 0.5 * ZCountsPerInch);
    Move(4,0);
    while(!CheckDone(4));
    MoveAtVel(2, -0.5 *ZCountsPerInch, 1 * ZCountsPerInch);
    while(!CheckDone(2));
    Move(1,OrigX);
    Move(0,OrigY);
    while(!CheckDone(0));
    while(!CheckDone(1));
    Move(2,OrigZ);
    while(!CheckDone(2));
}


int CommunicateTX2(int flagLen)
{
    //Enter code here
    char MyMessage[80];  // String to be created and displayed

    sprintf(MyMessage,"%d pictures to go!",(flagLen-1));  // build the message we want
to show

    MsgBox(MyMessage,MB_OK);  // Show it
    return 1;
}
```

KinematicsCoreXY.cpp , C program to modify kinematic classes in KMotionCNC

```
// KinematicsCoreXY.cpp: implementation of the CKinematicsCoreXY class.
//
//////////////////////////////////////////////////////////////////////

#include "stdafx.h"
#include "KinematicsCoreXY.h"

#define sqr(x) ((x)*(x))


//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

CKinematicsCoreXY::CKinematicsCoreXY()
{
    m_MotionParams.MaxLinearLength = 0.1;  // limit the segment lengs for
nonlinear systems
}

CKinematicsCoreXY::~CKinematicsCoreXY()
{

}

int CKinematicsCoreXY::TransformCADtoActuators(double x, double y, double z,
double a, double b, double c, double *Acts, bool NoGeo)
{
    // find motor counts of each actuator

    GeoCorrect(x,y,z,&x,&y,&z);

    Acts[0] = (x + y)*m_MotionParams.CountsPerInchX;
    Acts[1] = (x - y)*m_MotionParams.CountsPerInchY;
    Acts[2] = z*m_MotionParams.CountsPerInchZ;

    Acts[3] = a*m_MotionParams.CountsPerInchA;
    Acts[4] = b*m_MotionParams.CountsPerInchB;
    Acts[5] = c*m_MotionParams.CountsPerInchC;

    return 0;
}


// perform Inversion to go the other way

int CKinematicsCoreXY::TransformActuatorstoCAD(double *Acts, double *xr, double
*yr, double *zr, double *ar, double *br, double *cr, bool NoGeo)
{
    return InvertTransformCADtoActuators(Acts, xr, yr, zr, ar, br, cr);
}
```

KinematicsCoreXY.h, Header file for KinematicsCoreXY.cpp

```
// KinematicsCoreXY.h: interface for the CKinematicsCoreXY class.
//
//////////////////////////////////////////////////////////////////////


#if
!defined(AFX_KINEMATICSCOREXY_H__876A0A72_6EC3_48D0_9040_60AE3DA2F3C7__INCLUDED_)
#define AFX_KINEMATICSCOREXY_H__876A0A72_6EC3_48D0_9040_60AE3DA2F3C7__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "stdafx.h"

class CKinematicsCoreXY : public CKinematics
{
public:
    CKinematicsCoreXY();
    virtual ~CKinematicsCoreXY();
    virtual int TransformCADtoActuators(double x, double y, double z, double a,
double b, double c, double *Acts, bool NoGeo = false);
    virtual int TransformActuatorstoCAD(double *Acts, double *x, double *y, double
*z, double *a, double *b, double *c, bool NoGeo = false);
};

#endif //
!defined(AFX_KINEMATICSCOREXY_H__876A0A72_6EC3_48D0_9040_60AE3DA2F3C7__INCLUDED_)
```

Modifications to CoordMotion.cpp to allow the KinematicsCoreXY class
(Inserted at line 120)

```
// check for a special Kinematics File

    FILE *f = fopen((CString)MainPath + "\\Data\\Kinematics.txt","rt");

    if (f)
    {
        char s[81];
        fgets(s, 80, f);
        // one exists, check if it is calling for Geppetto otherwise assume it is
the 3Rod

        if (strstr(s, "5AxisTableAC") != NULL)
            Kinematics = new CKinematics5AxisTableAC;
        else if (strstr(s, "5AxisTableBC") != NULL)
            Kinematics = new CKinematics5AxisTableBC;
        else if (strstr(s, "Kinematics5AxisTableAGimbalB") != NULL)
            Kinematics = new CKinematics5AxisTableAGimbalB;
        else if (strstr(s, "5AxisGimbalAB") != NULL)
            Kinematics = new CKinematics5AxisGimbalAB;
        else if (strstr(s, "5AxisGimbalCB") != NULL)
            Kinematics = new CKinematics5AxisGimbalCB;
        else if (strstr(s, "GeppettoExtruder") != NULL)
            Kinematics = new CKinematicsGeppettoExtrude;
        else if (strstr(s, "Geppetto") != NULL)
            Kinematics = new CKinematicsGeppetto;
        else if (strstr(s, "Kinematics3Rod") != NULL)
            Kinematics = new CKinematics3Rod;
        else
            Kinematics = new CKinematicsCoreXY;

        fclose(f);
    }
    else
    {
        m_TCP_affects_actuators = false;
        Kinematics = new CKinematics;
    }
```

Appendix B

Drawing Package

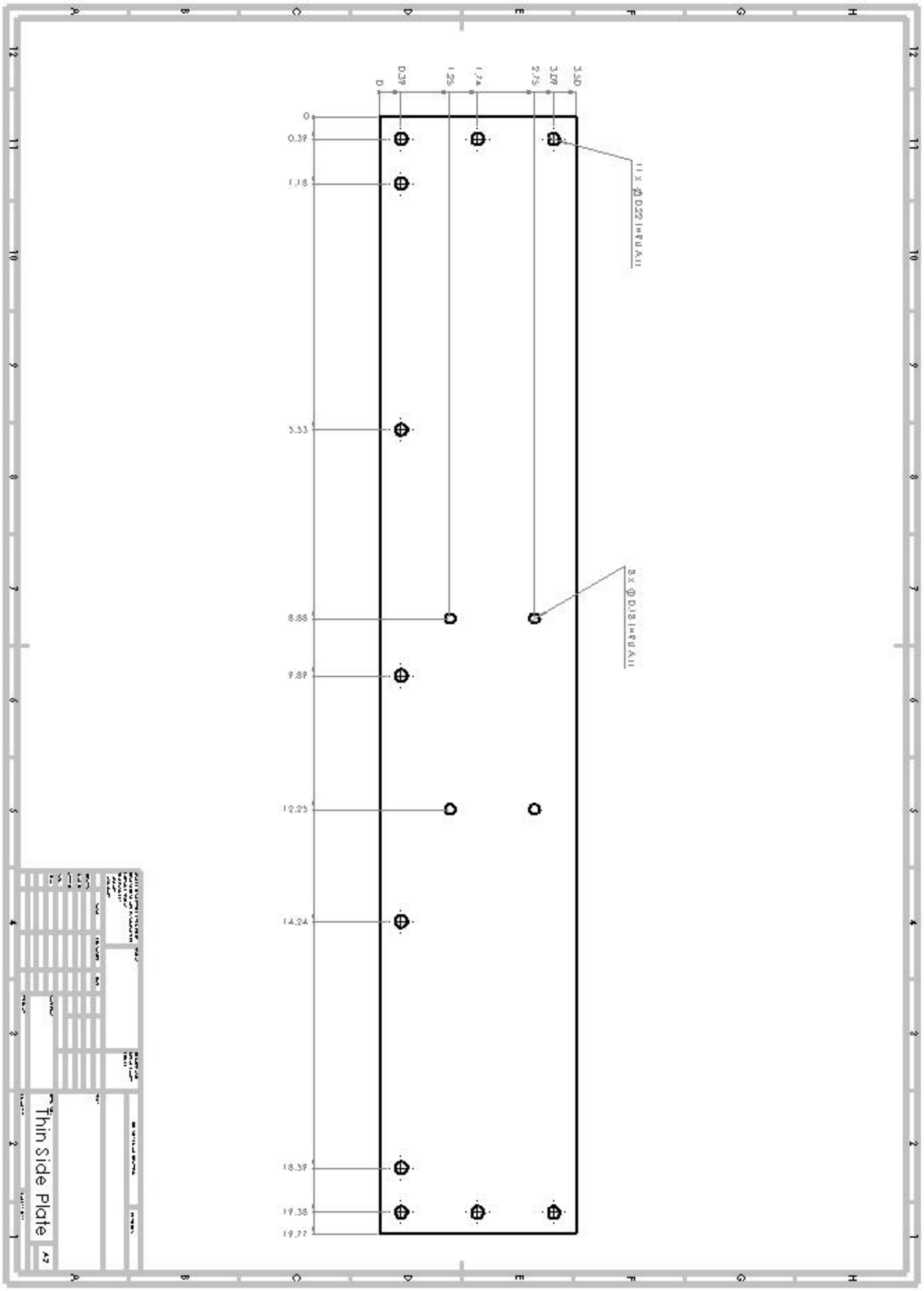


Figure B.1 Drawing for Frame Support Plate

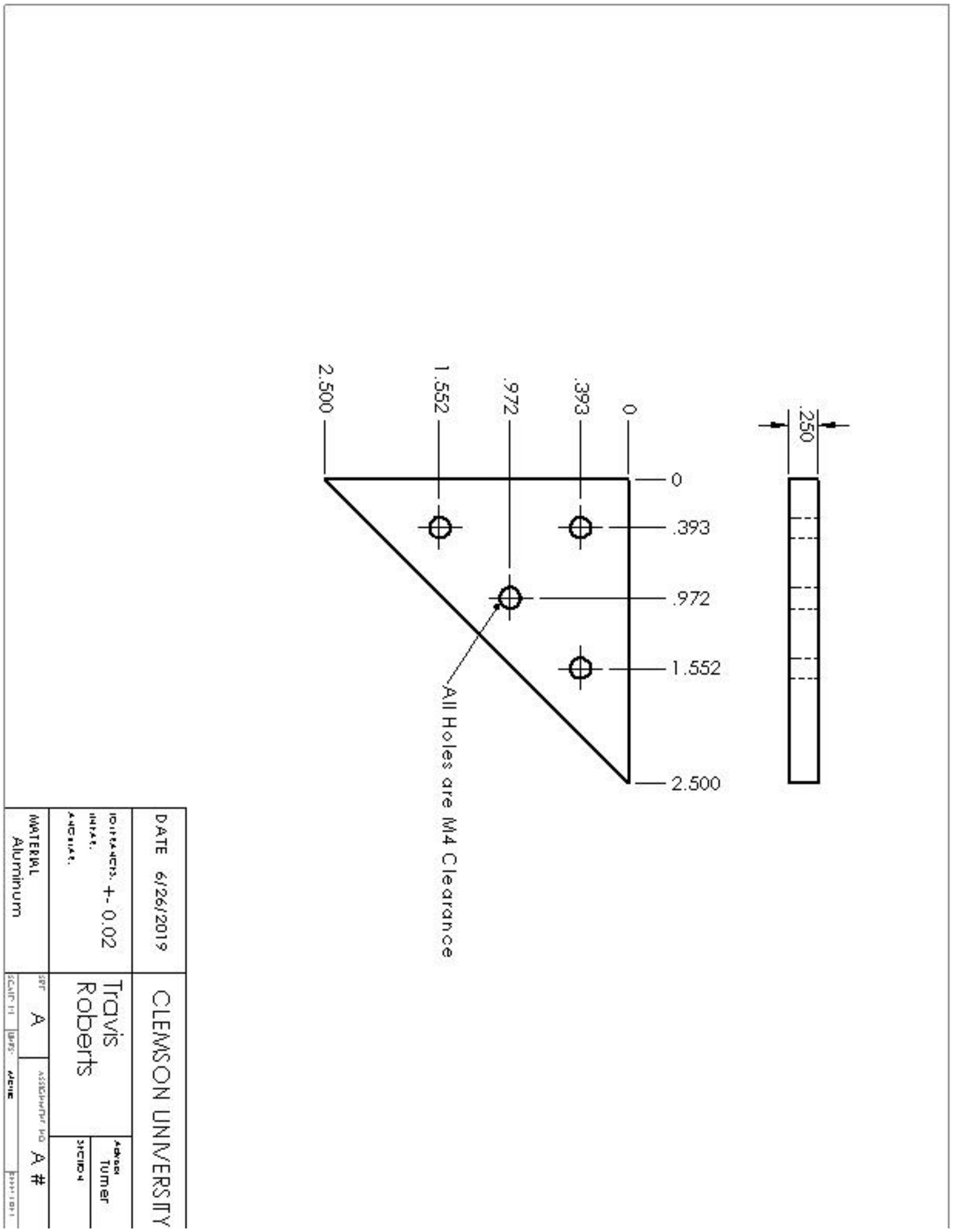


Figure B.2 Drawing for Corner Brace

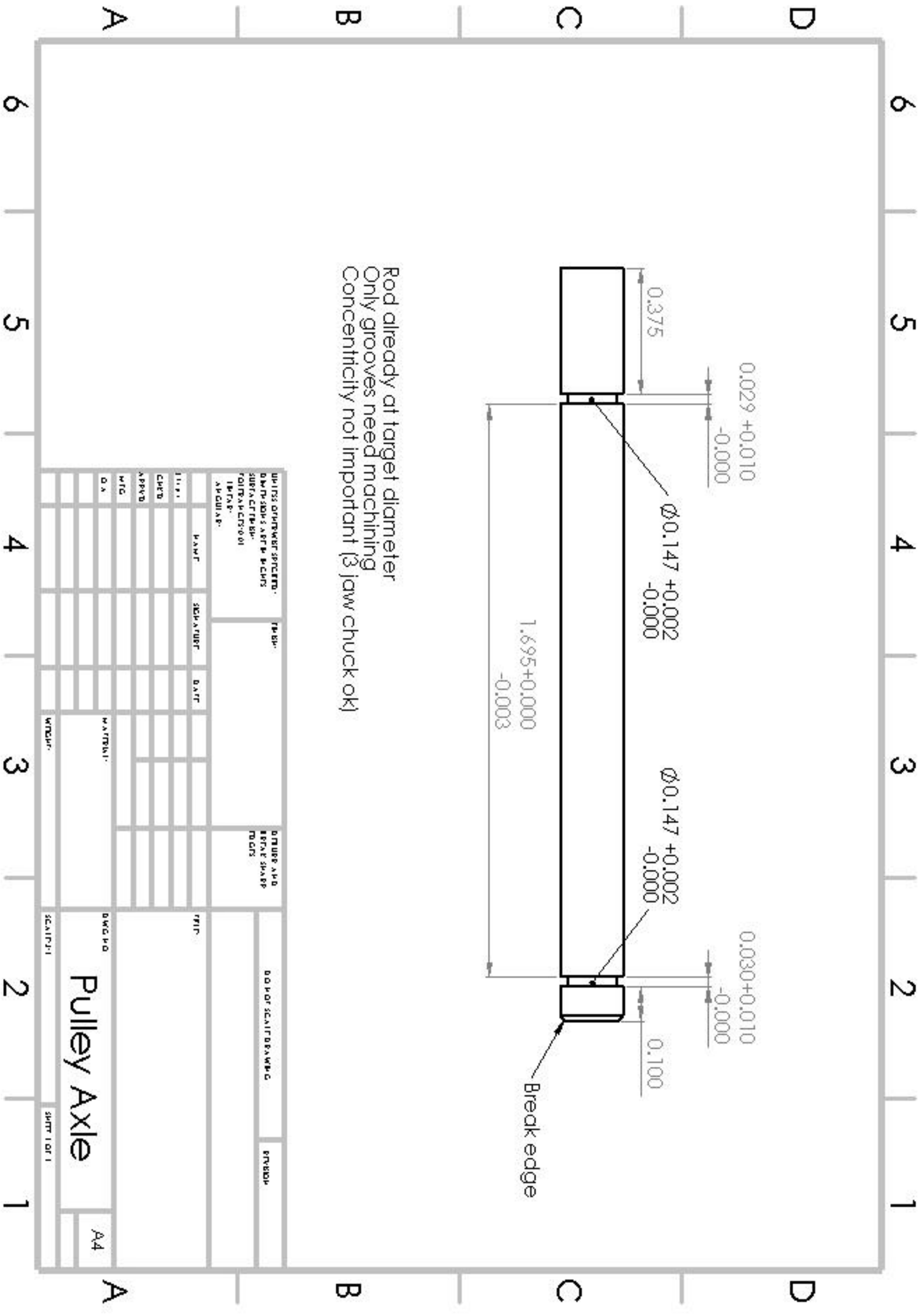


Figure B.3 Drawing for Pully Axle

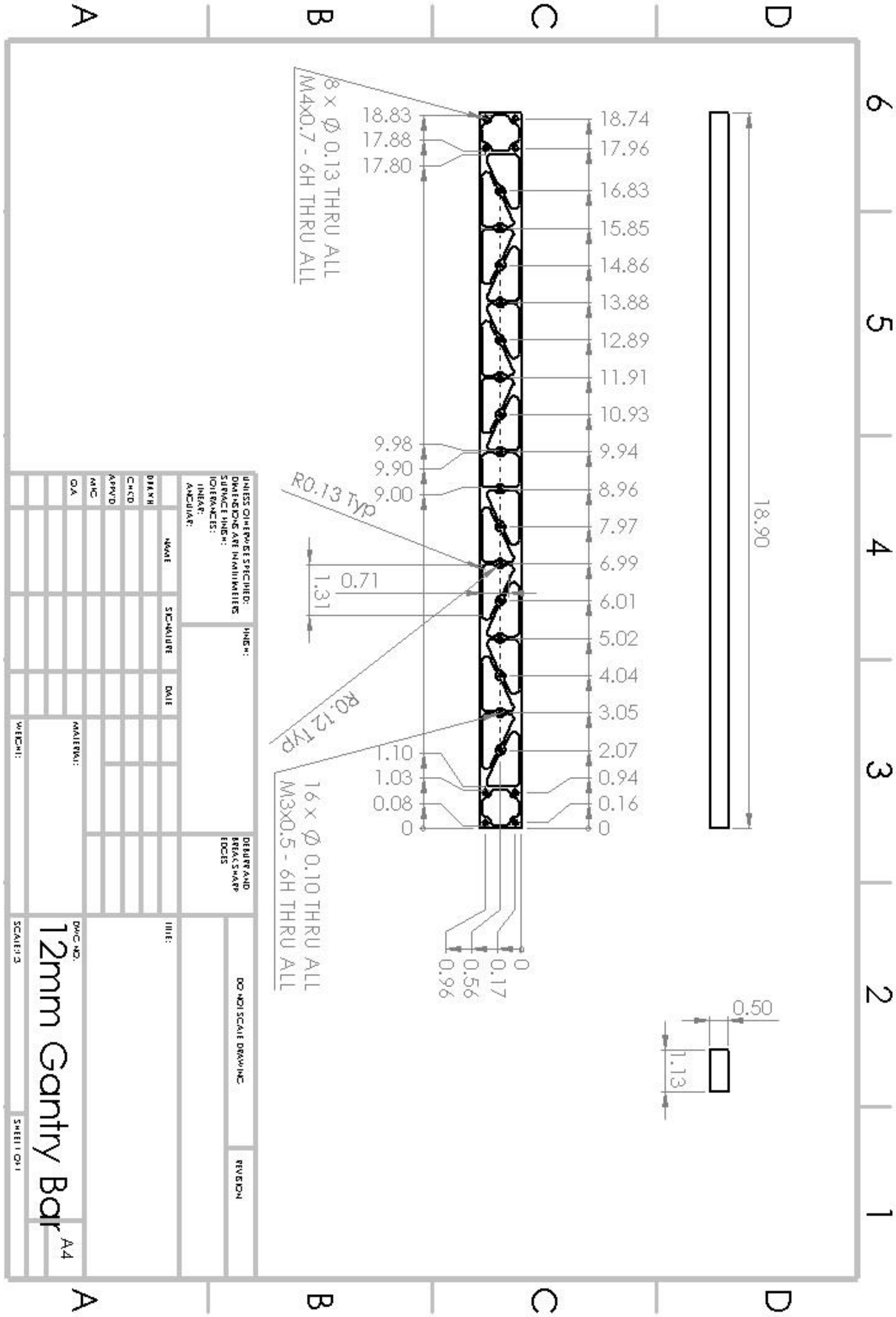


Figure B.4 Drawing for Gantry Bar

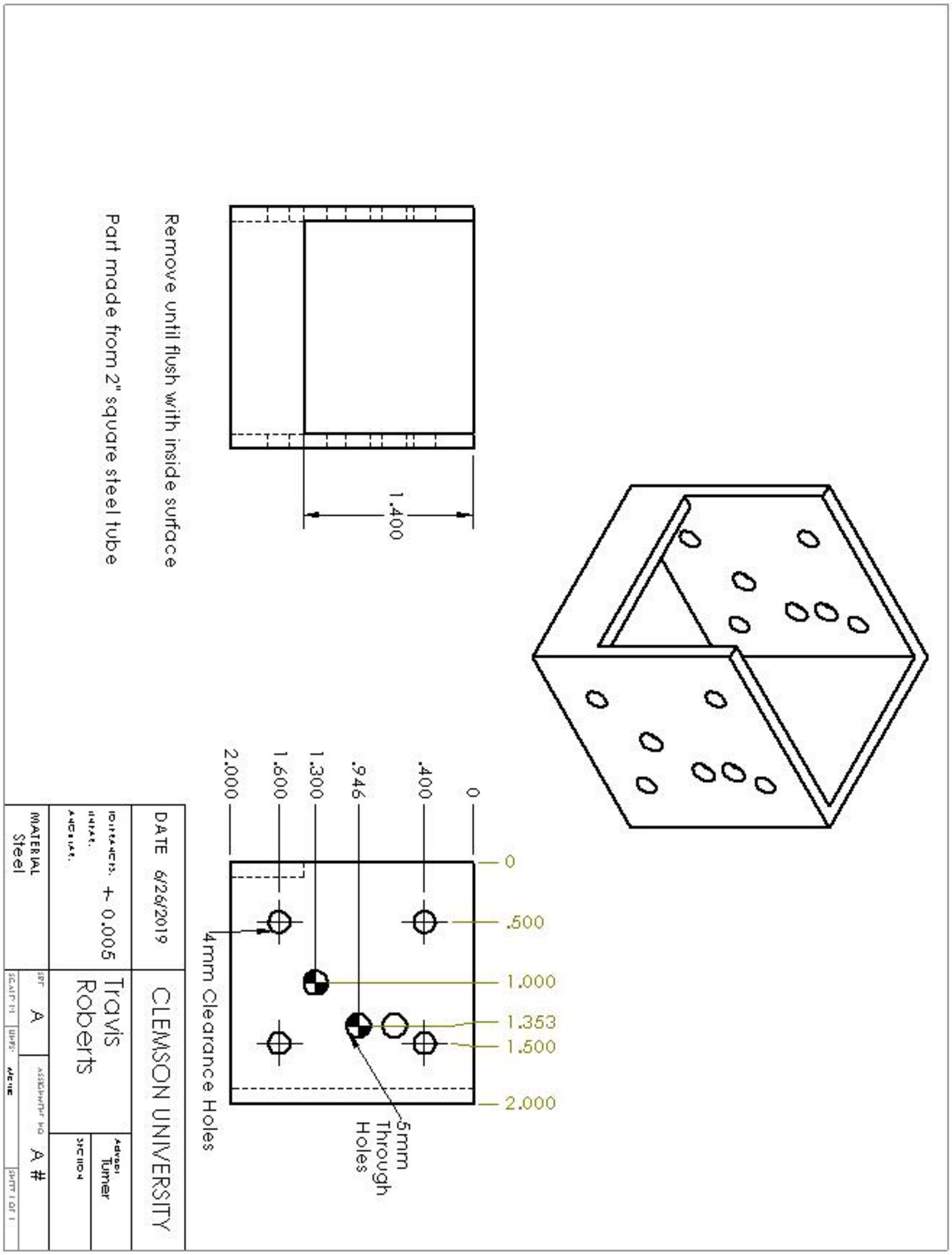


Figure B.5 Drawing for Right Pulley Mount

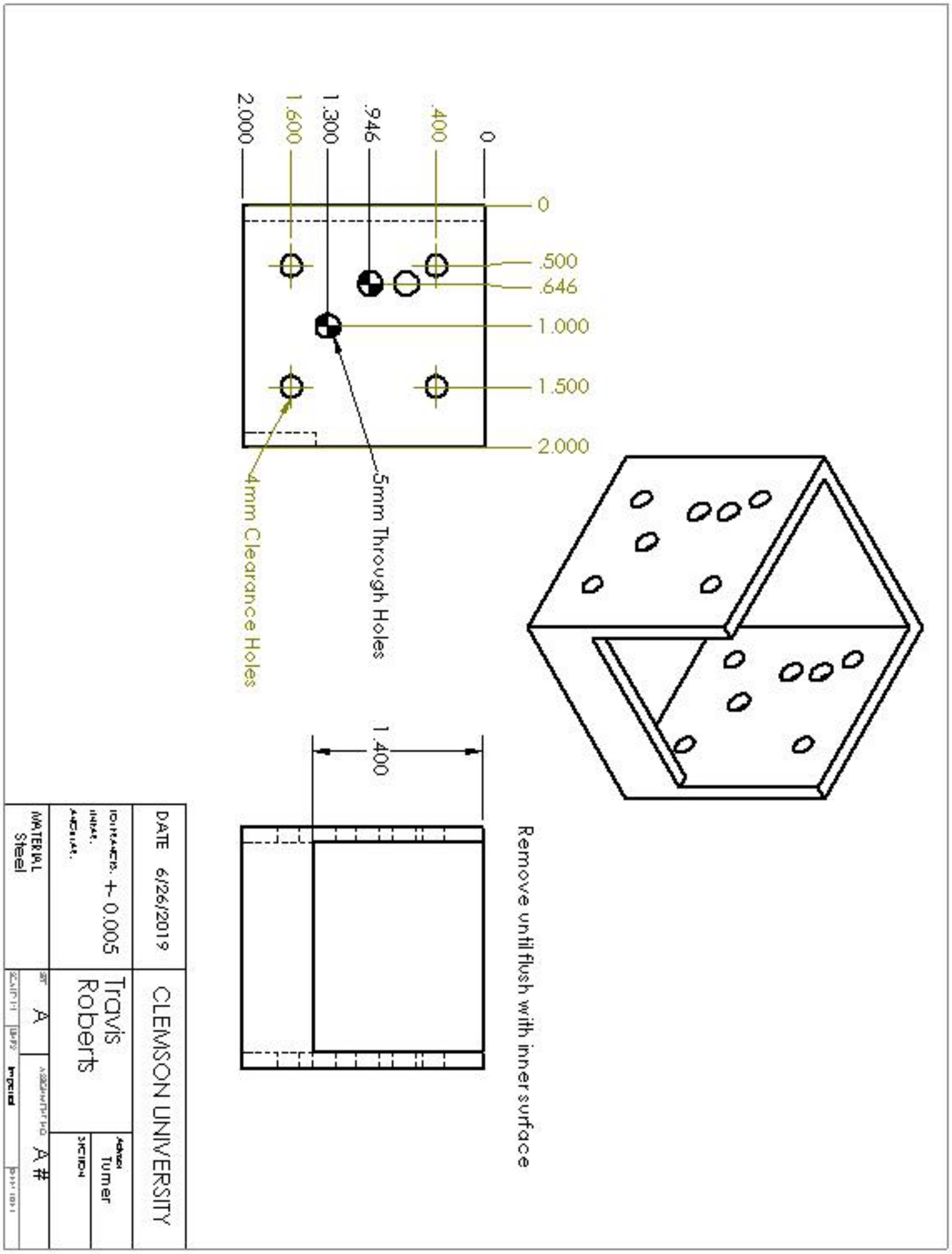


Figure B.6 Drawing for Left Pulley Mount

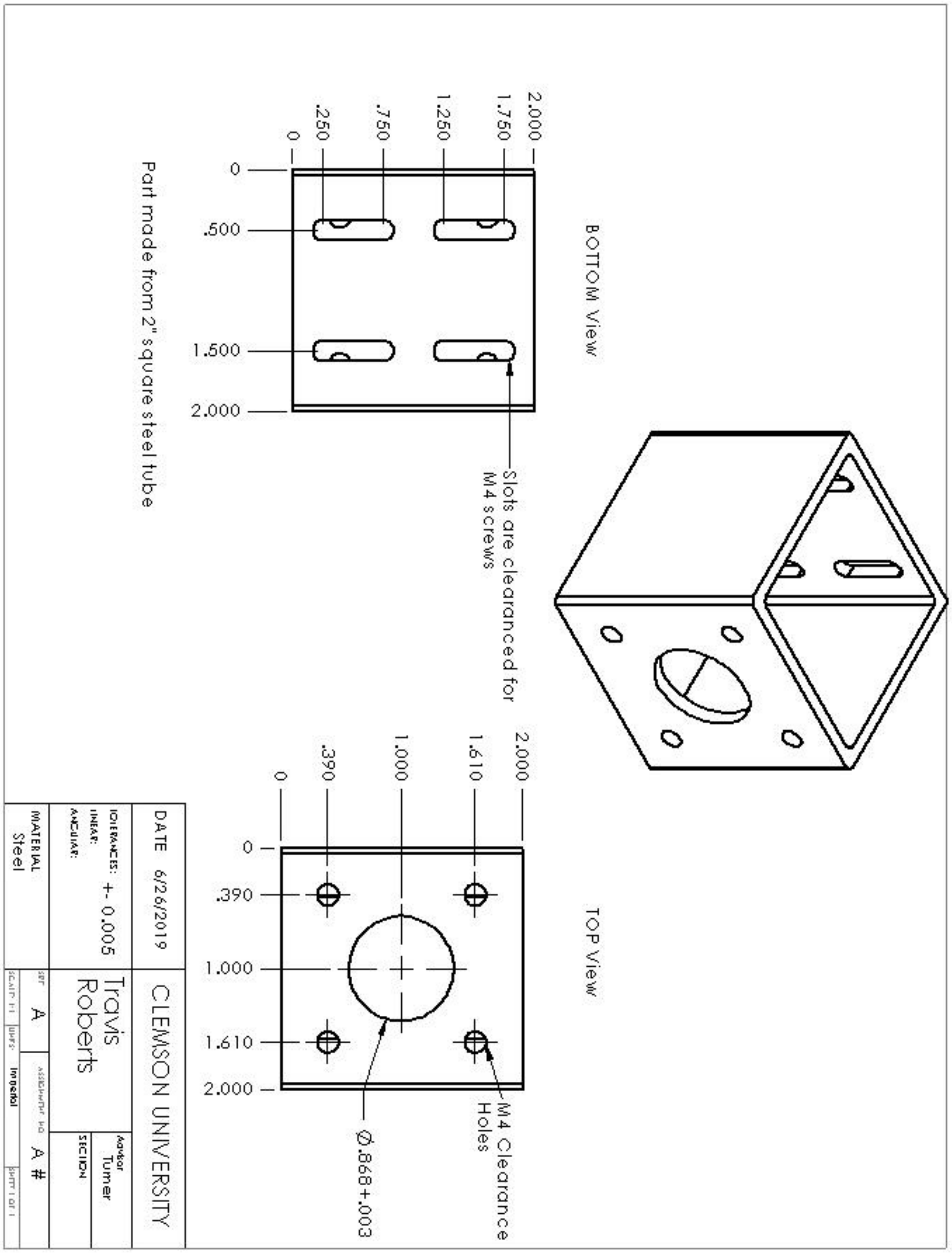


Figure B.7 Drawing for 2x2 Motor Standoff

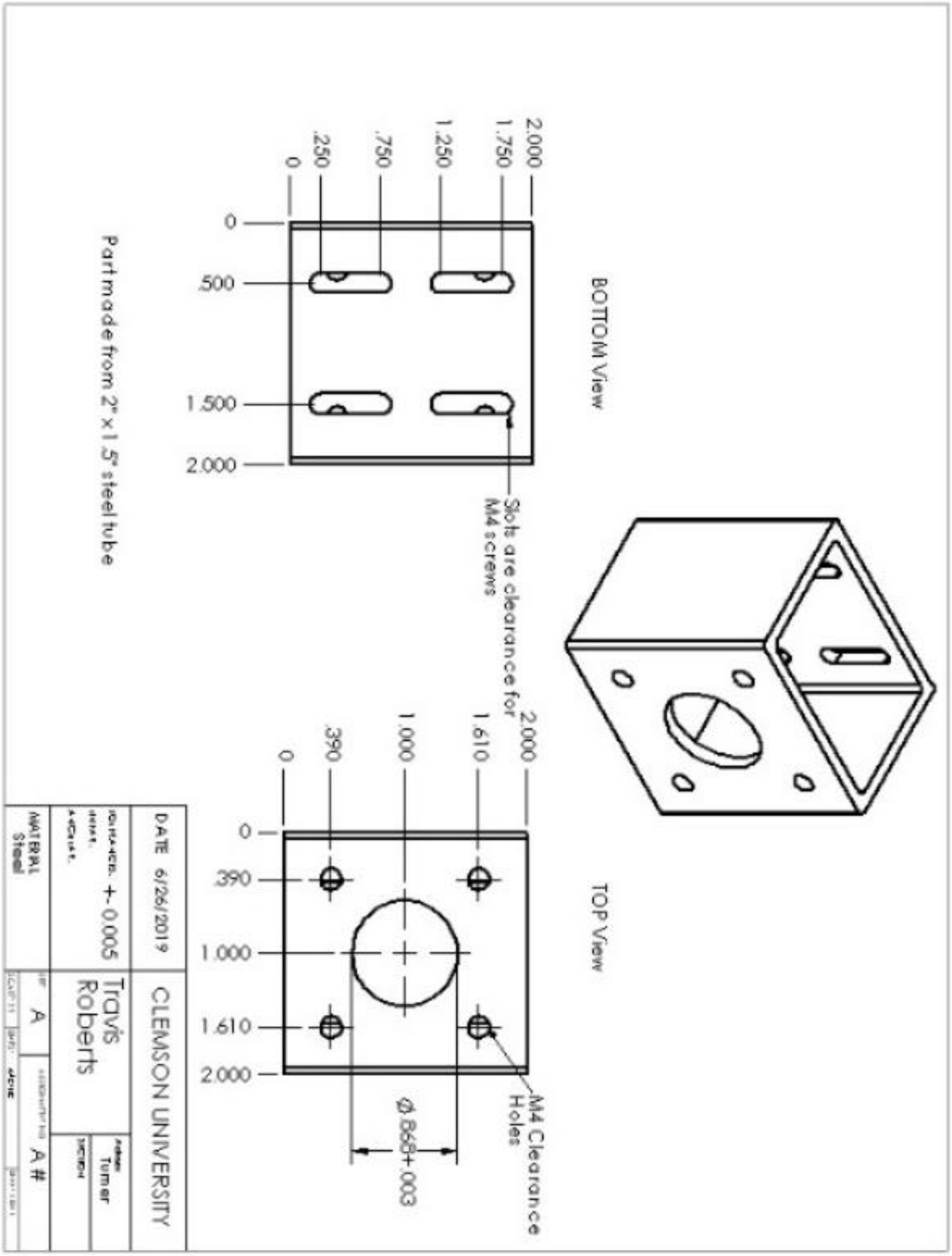


Figure B.8 Drawing for 2x1.5 Motor Standoff

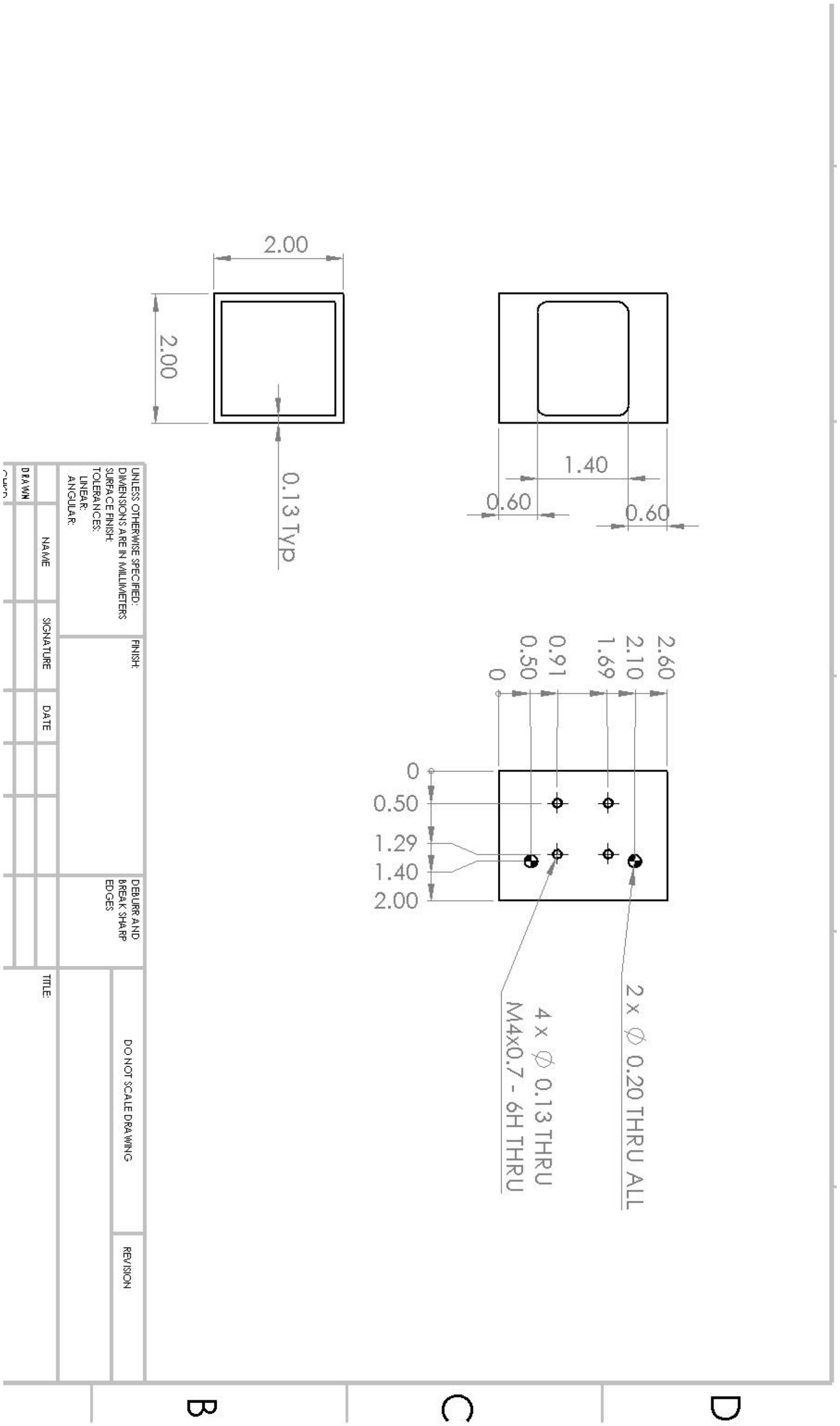


Figure B.9 Drawing for 2x2 Gantry Pulley Mount

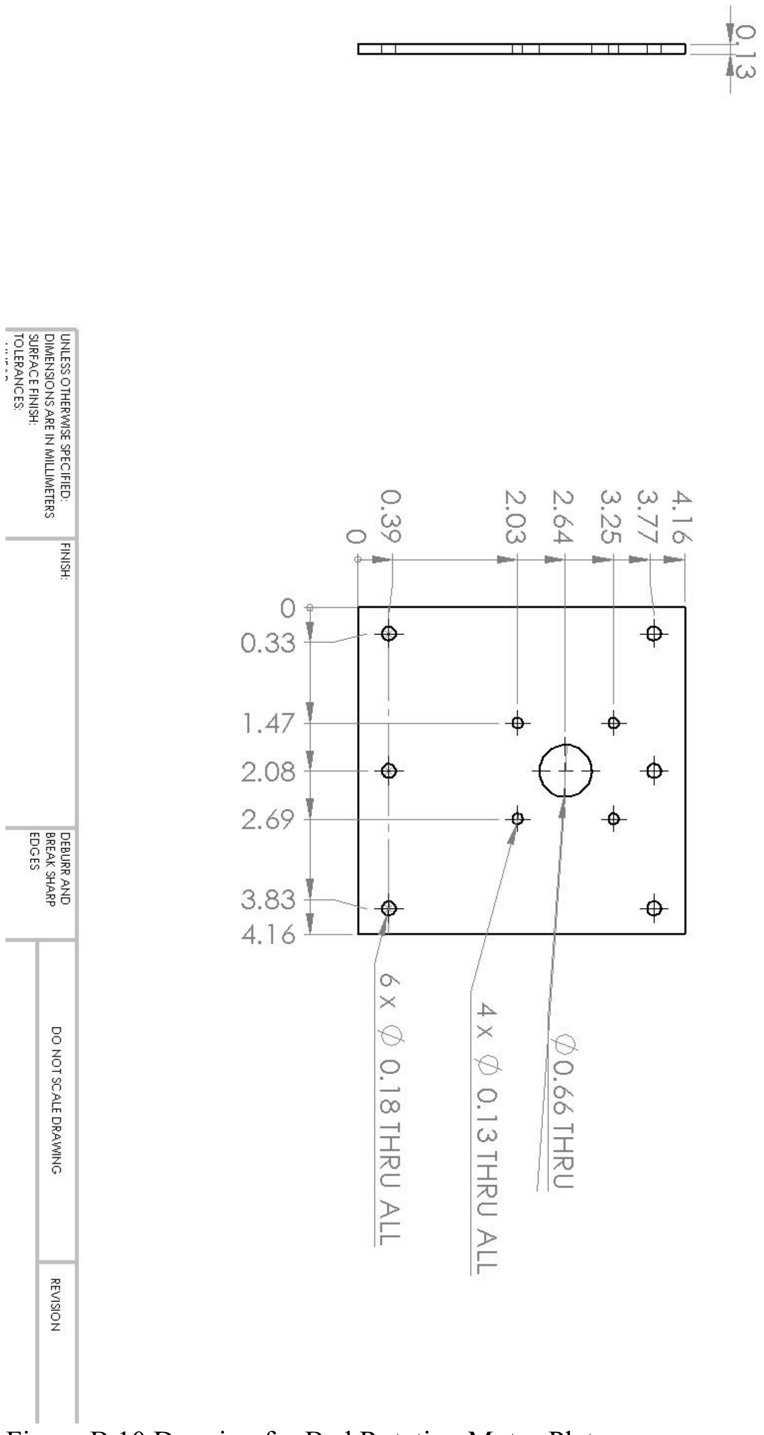


Figure B.10 Drawing for Bed Rotation Motor Plate

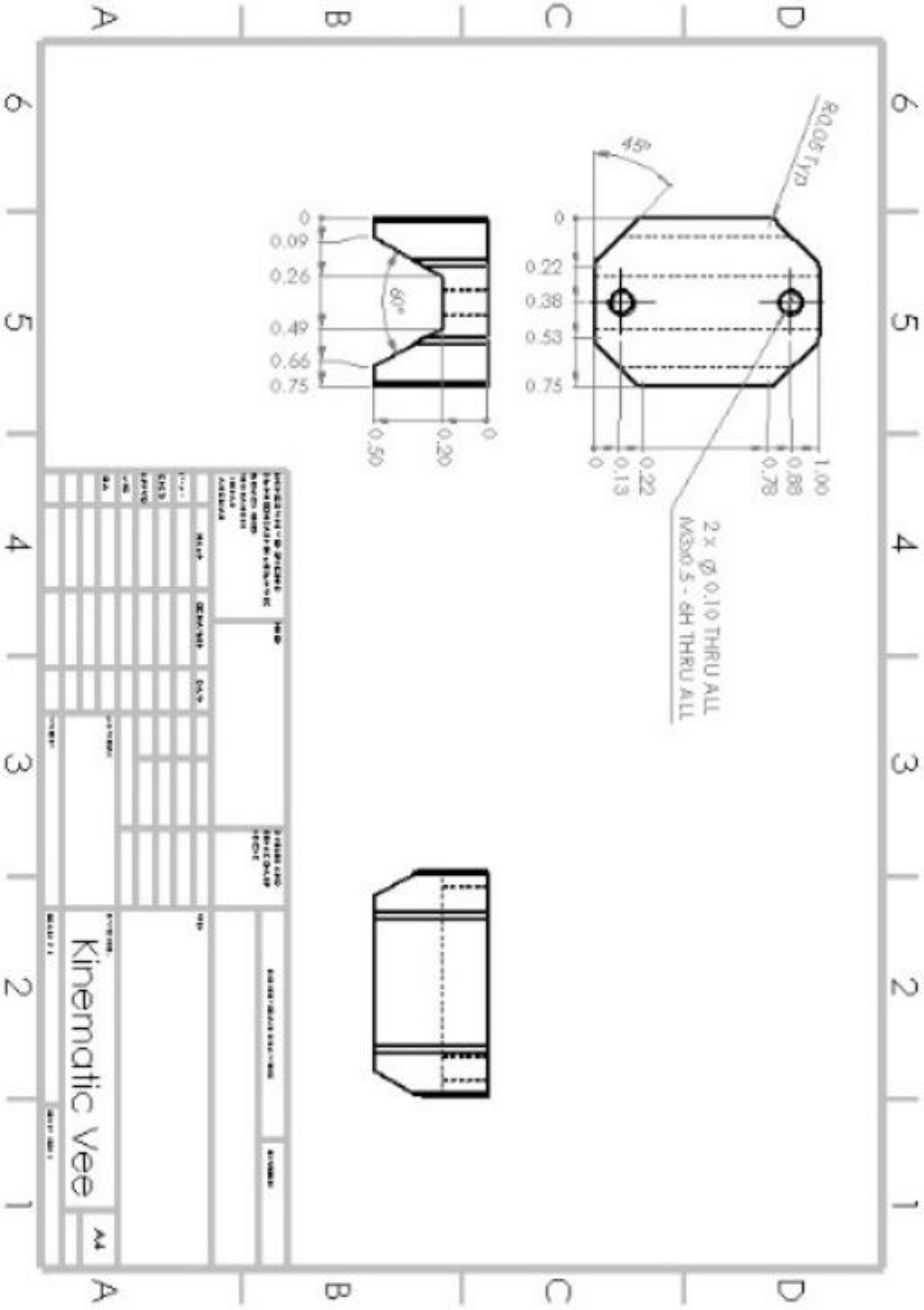


Figure B.11 Drawing for Kinematic Vee Block

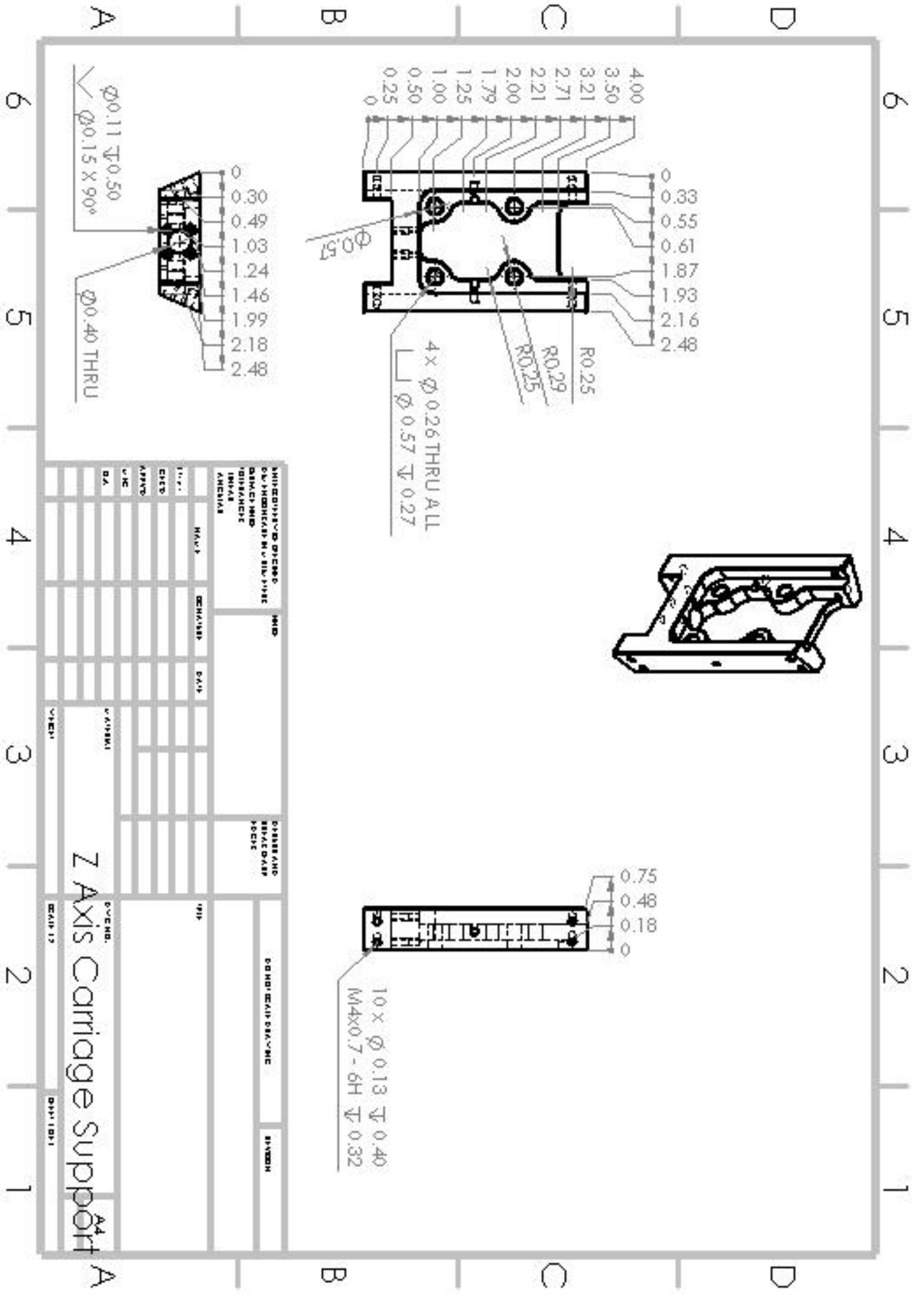


Figure B.12 Drawing for Z Axis Carriage Support

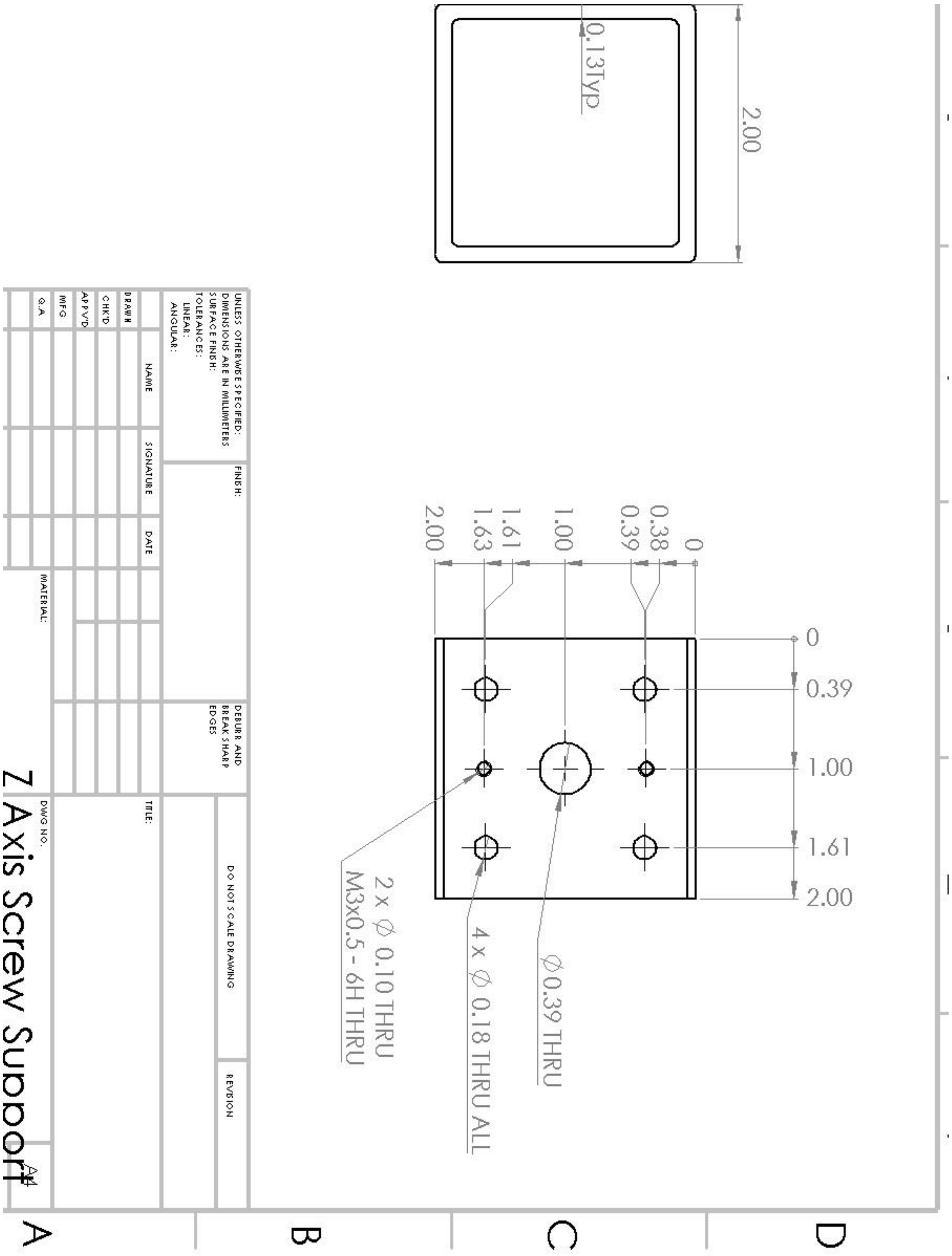


Figure B.13 Drawing for Z Axis Screw Support

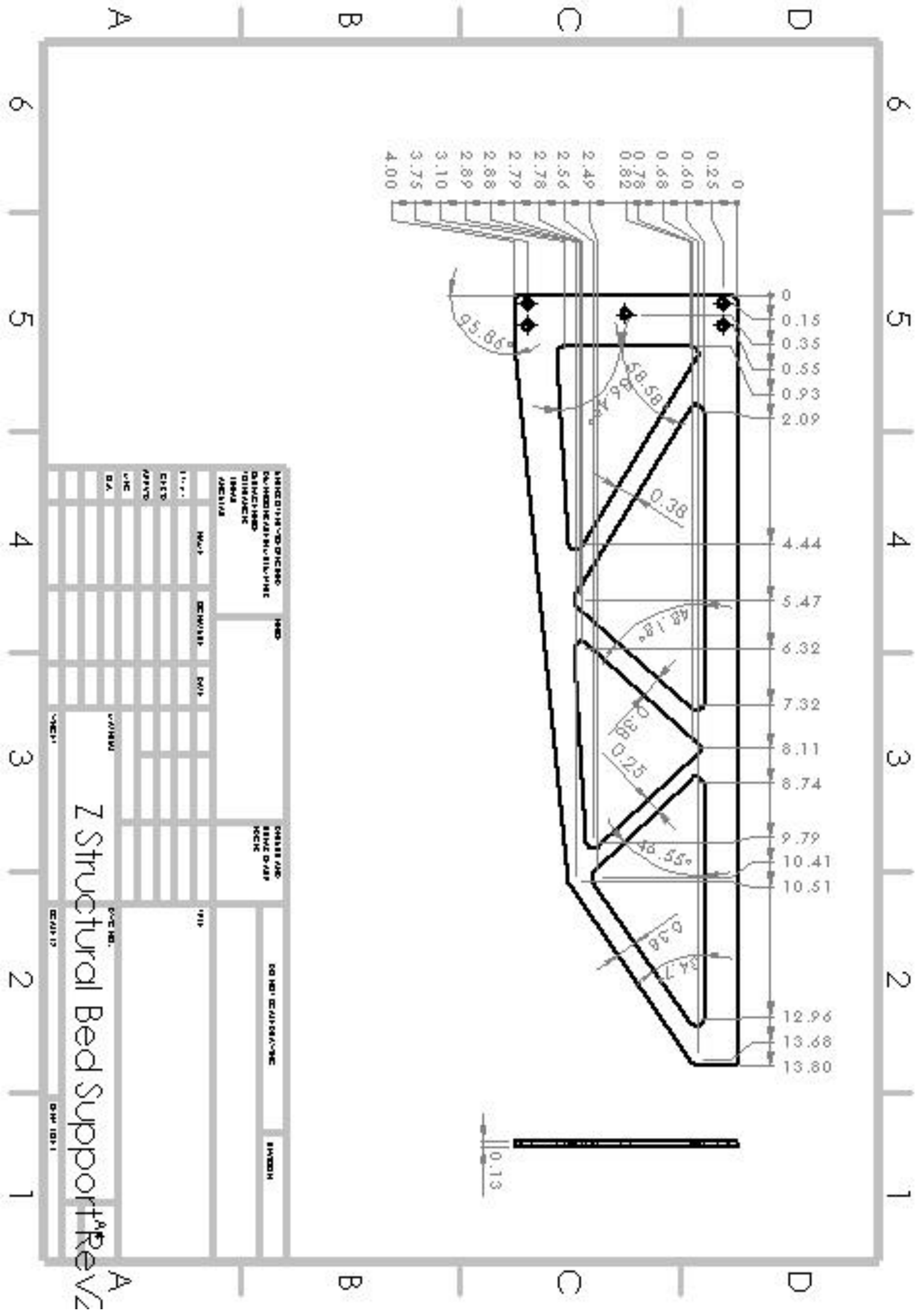


Figure B.14 Drawing of Z Bed Support

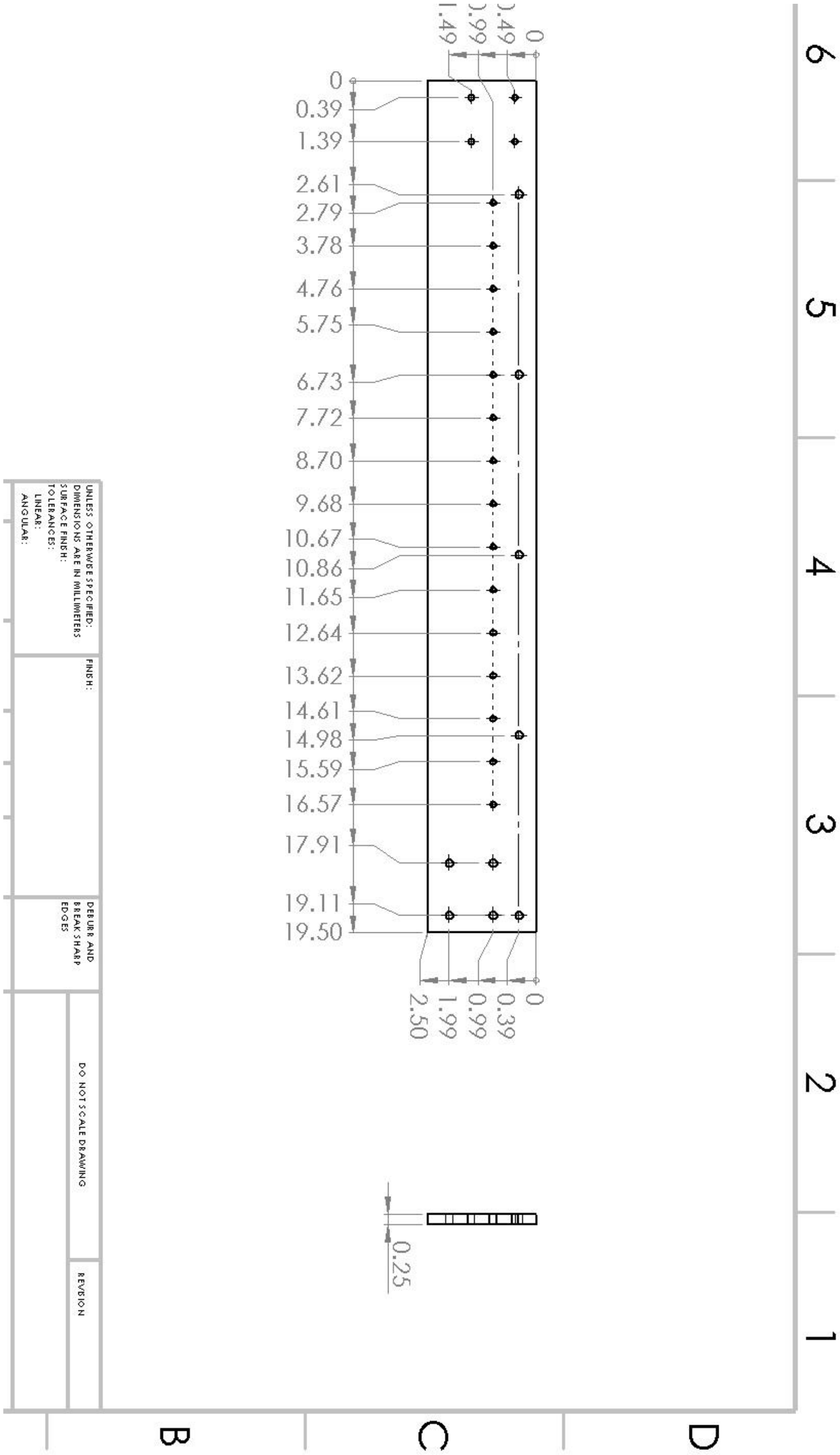


Figure B.15 Drawing of Top Plate

Appendix C

Additional Resources for OEM Parts

Dynomotion Online Help
https://dynomotion.com/Help/index.htm

Dynomotion Forum
https://www.dynomotion.com/forum/

Dynomotion Wiki
https://www.dynomotion.com/wiki/index.php/Main_Page

Pololu DRV8825 Motor Driver
https://www.pololu.com/product/2133/resources

E3D Titan Aqua Extruder
https://e3d-online.com/products/titan-aqua

E3D 300 mm x 300 mm Heated Bed
https://e3d-online.com/products/high-temperature-heated-beds

BLH Linear Rails
http://www.automation-overstock.com/pdfs/BLH_guideway.pdf

REFERENCES

[1] Q. Sun, G. M. Rizvi, C. T. Bellehumeur, and P. Gu, “Effect of processing conditions on the bonding quality of FDM polymer filaments,” *Rapid Prototyp. J.*, vol. 14, no. 2, pp. 72–80, doi: https://doi.org/10.1108/13552540810862028.

[2] T. Mukherjee, W. Zhang, and T. DebRoy, “An improved prediction of residual stresses and distortion in additive manufacturing,” *Comput. Mater. Sci.*, vol. 126, pp. 360–372, 2017, doi: https://doi.org/10.1016/j.commatsci.2016.10.003.

[3] A. H. Nickel, D. M. Barnett, and F. B. Prinz, “Thermal stresses and deposition patterns in layered manufacturing,” *Mater. Sci. Eng. A*, vol. 317, no. 1, pp. 59–64, 2001, doi: https://doi.org/10.1016/S0921-5093(01)01179-0.

[4] N. W. Klingbeil, J. L. Beuth, R. K. Chin, and C. Amon, “Residual stress-induced warping in direct metal solid freeform fabrication,” *Int. J. Mech. Sci.*, vol. 44, pp. 57–77, 2002, doi: 10.1016/S0020-7403(01)00084-4.

[5] T. Mukherjee, V. Manvatkar, A. De, and T. DebRoy, “Mitigation of thermal distortion during additive manufacturing,” *Scr. Mater.*, vol. 127, pp. 79–83, 2017, doi: https://doi.org/10.1016/j.scriptamat.2016.09.001.

[6] D. W. Manthey and D. Lee, “Vision-based strain measurement system,” *JOM*, vol. 47, no. 7, pp. 46–49, 1995.

[7] L. M. Galantucci, F. Lavecchia, and G. Percoco, “Multistack Close Range Photogrammetry for Low Cost Submillimeter Metrology,” *J. Comput. Inf. Sci. Eng.*, vol. 13, no. 4, 2013, doi: 10.1115/1.4024973.

[8] H. Wu, Y. Wang, and Z. Yu, “In situ monitoring of FDM machine condition via acoustic emission,” *Int. J. Adv. Manuf. Technol.*, vol. 84, no. 5, pp. 1483–1495, 2016, doi: 10.1007/s00170-015-7809-4.

[9] J. Liu, Y. Hu, B. Wu, and Y. Wang, “An improved fault diagnosis approach for FDM process with acoustic emission,” *J. Manuf. Process.*, vol. 35, pp. 570–579, 2018, doi: https://doi.org/10.1016/j.jmapro.2018.08.038.

[10] Y. Li, W. Zhao, Q. Li, T. Wang, and G. Wang, “In-Situ Monitoring and Diagnosing for Fused Filament Fabrication Process Based on Vibration Sensors,” *Sensors*, vol. 19, no. 11, 2019, doi: 10.3390/s19112589.

[11] Y. Tlegenov, W. F. Lu, and G. S. Hong, “A dynamic model for current-based nozzle condition monitoring in fused deposition modelling,” *Prog. Addit. Manuf.*, vol. 4, no. 3, pp. 211–223, 2019, doi: 10.1007/s40964-019-00089-3.

[12] C. Kim, D. Espalin, A. Cuaron, M. A. Perez, E. MacDonald, and R. B. Wicker, “A study to detect a material deposition status in fused deposition modeling technology,” in *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2015, pp. 779–783, doi: 10.1109/AIM.2015.7222632.

[13] W. De Backer, P. Sinkez, I. Chhabra, M. J. Van Tooren, and A. Bergs, “In-Process Monitoring of Continuous Fiber Additive Manufacturing through Force/Torque Sensing on the Nozzle,” in *AIAA Scitech 2020 Forum*, .

[14] M. Moretti, F. Bianchi, and N. Senin, “Towards the development of a smart fused filament fabrication system using multi-sensor data fusion for in-process monitoring,” *Rapid Prototyp. J.*, vol. 26, no. 7, pp. 1249–1261, 2020, doi: 10.1108/RPJ-06-2019-0167.

[15] J. E. Seppala and K. D. Migler, “Infrared thermography of welding zones produced by polymer extrusion additive manufacturing,” *Addit. Manuf.*, vol. 12, pp. 71–76, 2016, doi: https://doi.org/10.1016/j.addma.2016.06.007.

[16] E. Malekipour, S. Attoye, and H. El-Mounayri, “Investigation of Layer Based Thermal Behavior in Fused Deposition Modeling Process by Infrared Thermography,” *Procedia Manuf.*, vol. 26, pp. 1014–1022, 2018, doi: 10.1016/j.promfg.2018.07.133.

[17] E. Ferraris, J. Zhang, and B. Van Hooreweder, “Thermography based in-process monitoring of Fused Filament Fabrication of polymeric parts,” *CIRP Ann.*, vol. 68, no. 1, pp. 213–216, 2019, doi: https://doi.org/10.1016/j.cirp.2019.04.123.

[18] Z. Jin, Z. Zhang, and G. X. Gu, “Automated Real-Time Detection and Prediction of Interlayer Imperfections in Additive Manufacturing Processes Using Artificial Intelligence,” *Adv. Intell. Syst.*, vol. 2, no. 1, p. 1900130, 2020, doi: https://doi.org/10.1002/aisy.201900130.

[19] A. Saluja, J. Xie, and K. Fayazbakhsh, “A closed-loop in-process warping detection system for fused filament fabrication using convolutional neural networks,” *J. Manuf. Process.*, vol. 58, no. November 2019, pp. 407–415, 2020, doi: 10.1016/j.jmapro.2020.08.036.

[20] F. Baumann and D. Roller, “Vision based error detection for 3D printing processes,” *MATEC Web Conf.*, vol. 59, pp. 3–9, 2016, doi: 10.1051/matecconf/20165906003.

[21] B. Yao, F. Imani, A. S. Sakpal, E. W. Reutzel, and H. Yang, “Multifractal Analysis of Image Profiles for the Characterization and Detection of Defects in Additive Manufacturing,” *J. Manuf. Sci. Eng.*, vol. 140, no. 3, 2018, doi: 10.1115/1.4037891.

[22] I. Cummings, E. Hillstrom, R. Newton, E. Flynn, and A. Wachtor, “In-Process Ultrasonic Inspection of Additive Manufactured Parts,” 2016, pp. 235–247.

[23] T. Huang, S. Wang, S. Yang, and W. Dai, “Statistical process monitoring in a specified period for the image data of fused deposition modeling parts with consistent layers,” *J. Intell. Manuf.*, vol. 32, no. 8, pp. 2181–2196, 2021, doi: 10.1007/s10845-020-01628-4.

[24] K. Okarma and J. Fastowicz, “Adaptation of Full-Reference Image Quality Assessment Methods for Automatic Visual Evaluation of the Surface Quality of 3D Prints,” *Elektron. ir Elektrotechnika*, vol. 25, no. 5, pp. 57–62, 2019, doi: 10.5755/j01.eie.25.5.24357.

[25] K. Okarma and J. Fastowicz, “No-reference quality assessment of 3D prints based on the GLCM analysis,” in *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2016, pp. 788–793, doi: 10.1109/MMAR.2016.7575237.

[26] K. Okarma and J. Fastowicz, "Improved quality assessment of colour surfaces for additive manufacturing based on image entropy," *Pattern Anal. Appl.*, vol. 23, 2020, doi: 10.1007/s10044-020-00865-w.

[27] K. Okarma and J. Fastowicz, "Color Independent Quality Assessment of 3D Printed Surfaces Based on Image Entropy," in *Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017*, 2018, pp. 308–315.

[28] S. Nuchitprasitchai, "AN ALGORITHM FOR RECONSTRUCTING THREE-DIMENSIONAL IMAGES FROM OVERLAPPING TWO-DIMENSIONAL INTENSITY MEASUREMENTS WITH RELAXED CAMERA POSITIONING REQUIREMENTS, WITH APPLICATION TO ADDITIVE MANUFACTURING," 2017.

[29] S. Nuchitprasitchai, M. Roggemann, and J. M. Pearce, "Factors effecting real-time optical monitoring of fused filament 3D printing," *Prog. Addit. Manuf.*, vol. 2, no. 3, pp. 133–149, 2017, doi: 10.1007/s40964-017-0027-x.

[30] K. He, Q. Zhang, and Y. Hong, "Profile monitoring based quality control method for fused deposition modeling process," *J. Intell. Manuf.*, vol. 30, no. 2, pp. 947–958, 2019, doi: 10.1007/s10845-018-1424-9.

[31] S. Nuchitprasitchai, M. C. Roggemann, and J. M. Pearce, "Three Hundred and Sixty Degree Real-Time Monitoring of 3-D Printing Using Computer Analysis of Two Camera Views," *J. Manuf. Mater. Process.*, vol. 1, no. 1, 2017, doi: 10.3390/jmmp1010002.

[32] W. Lin, H. Shen, J. Fu, and S. Wu, "Online quality monitoring in material extrusion additive manufacturing processes based on laser scanning technology," *Precis. Eng.*, vol. 60, pp. 76–84, 2019, doi: https://doi.org/10.1016/j.precisioneng.2019.06.004.

[33] P. Rao, Z. Kong, C. Duty, and R. Smith, "Three Dimensional Point Cloud Measurement Based Dimensional Integrity Assessment for Additive Manufactured Parts Using Spectral Graph Theory," 2016, p. V002T04A048, doi: 10.1115/MSEC2016-8516.

[34] M. Khanzadeh, P. Rao, R. Jafari-Marandi, B. K. Smith, M. A. Tschopp, and L. Bian, "Quantifying Geometric Accuracy with Unsupervised Machine Learning: Using Self-Organizing Map on Fused Filament Fabrication Additive Manufacturing Parts," *J. Manuf. Sci. Eng. Trans. ASME*, vol. 140, no. 3, 2018, doi: 10.1115/1.4038598.

[35] C. Kopsacheilis, P. Charalampous, I. Kostavelis, and D. Tzovaras, "In Situ Visual Quality Control in 3D Printing," 2020, pp. 317–324, doi: 10.5220/0009329803170324.

[36] O. Holzmond and X. Li, "In situ real time defect detection of 3D printed parts," *Addit. Manuf.*, vol. 17, pp. 135–142, 2017, doi: https://doi.org/10.1016/j.addma.2017.08.003.

[37] A. Dickins, T. Widjanarko, S. Lawes, P. Stravroulakis, and R. Leach, “Design of a multi-sensor in-situ inspection system for additive manufacturing,” *Proc. - 2018 ASPE euspen Summer Top. Meet. Adv. Precis. Addit. Manuf.*, no. July, pp. 248–252, 2018.

[38] J. Straub, “An approach to detecting deliberately introduced defects and micro-defects in 3D printed objects,” in *Pattern Recognition and Tracking XXVIII*, 2017, vol. 10203, pp. 144–157, doi: 10.1117/12.2264588.

[39] J. Straub, “Identifying positioning-based attacks against 3D printed objects and the 3D printing process,” in *Pattern Recognition and Tracking XXVIII*, 2017, vol. 10203, no. June 2017, pp. 22–34, doi: 10.1117/12.2264671.

[40] Y. Fu, A. Downey, L. Yuan, A. Pratt, and Y. Balogun, “In situ monitoring for fused filament fabrication process: A review,” *Addit. Manuf.*, vol. 38, no. December 2020, pp. 0–2, 2021, doi: 10.1016/j.addma.2020.101749.

[41] D. Sims-Waterhouse, S. Piano, and R. Leach, “Verification of micro-scale photogrammetry for smooth three-dimensional object measurement,” *Meas. Sci. Technol.*, vol. 28, no. 5, 2017, doi: 10.1088/1361-6501/aa6364.

[42] G. Percoco, A. J. Sánchez Salmerón, and A.-J. Sánchez-Salmerón, “Photogrammetric measurement of 3D freeform millimetre-sized objects with micro features: An experimental validation of the close-range camera calibration model for narrow angles of view,” *Meas. Sci. Technol.*, vol. 26, no. 9, 2015, doi: 10.1088/0957-0233/26/9/095203.

[43] G. Percoco, F. Lavecchia, and A. J. S. Salmerón, “Preliminary study on the 3D digitization of millimeter scale products by means of photogrammetry,” *Procedia CIRP*, vol. 33, pp. 257–262, 2015, doi: 10.1016/j.procir.2015.06.046.

[44] D. Sims-Waterhouse, P. Bointon, S. Piano, and R. K. Leach, “Experimental comparison of photogrammetry for additive manufactured parts with and without laser speckle projection,” in *Optical Measurement Systems for Industrial Inspection X*, 2017, vol. 10329, pp. 220–226, doi: 10.1117/12.2269507.

[45] J. N. Jensen *et al.*, “Photogrammetry for Repositioning in Additive Manufacturing,” 2017.

[46] S. Catalucci, N. Senin, D. Sims-Waterhouse, S. Ziegelmeier, S. Piano, and R. Leach, “Measurement of complex freeform additively manufactured parts by structured light and photogrammetry,” *Meas. J. Int. Meas. Confed.*, vol. 164, p. 108081, 2020, doi: 10.1016/j.measurement.2020.108081.

[47] P. I. Stavroulakis and R. K. Leach, “Invited Review Article: Review of post-process optical form metrology for industrial-grade metal additive manufactured parts,” *Rev. Sci. Instrum.*, vol. 87, no. 4, 2016, doi: 10.1063/1.4944983.

[48] L. C. Hale and A. H. Slocum, “Optimal design techniques for kinematic couplings,” *Precis. Eng.*, vol. 25, no. 2, pp. 114–127, 2001, doi: 10.1016/S0141-6359(00)00066-0.

[49] A. H. Slocum, “Kinematic couplings for precision fixturing-Part I: Formulation of design parameters,” *Precis. Eng.*, vol. 10, no. 2, pp. 85–91, 1988, doi: 10.1016/0141-6359(88)90005-0.

[50] Markforged, “Markforged.” https://markforged.com/resources/blog/markforged-3d-print-farm.
[51] J. Vasquez, “Jubilee - Github.” https://github.com/machineagency/jubilee.
[52] K. Stevenson, “No Title.” https://www.fabbaloo.com/2019/11/a-revolutionary-revolving-3d-printer-extruder-from-fuselab.
[53] I. E. Moyer, “CoreXY Theory,” 2012. http://corexy.com/theory.html.