Cover Letter of Transmittal

UNCC Stippling Machine Design Team
9201 University City Blvd
Charlotte, NC 28223
May 2, 2018

Senior Design Committee
9201 University City Blvd,
Charlotte, NC 28223

Senior Design Committee,

This document is the full design package for the University of North Carolina at Charlotte Stippling Machine Design project for Senior Design II. This project is supported by and for an industry supported sponsor, Mammoth Machine and Design. The intent of this project was to design and develop a fully functioning prototype, computer numerically controlled multi-axis stippling machine that will repeatedly stipple any free form surface with a heating element.

The machine is categorized into five main sub-assemblies: The Frame, X-axis, Y-axis, Z-axis, B-axis, C-axis, articulated head enclosing the stippling mechanism, and control unit. The frame is made of T-slot railings. The X-axis consists of a linear actuator with a work table mounted on a carriage on top of the X-axis. The Y-axis is mounted on the top of the frame. The other axes namely, Z-axis, C-axis and B-axis are mounted on the Y-axis carriage respectively. The stippling operation is carried out by a soldering iron tip mounted inside the articulated head. The tip is driven by a crank slider mechanism which stipples at a frequency of 16 Hz. All the axes are driven by stepper motors and the stippling tip is driven by an independent DC motor. The controls are enclosed in a control module attached on the side of the machine. The machine is controlled using LinuxCNC software running on the Linux Ubuntu distribution.

The Stippling Machine Design team is confident that this design meets the majority of all the minimum requirements defined in the statement of work provided by Mammoth Machine and Design. The team is thankful to the Senior Design Committee for sanctioning this project and Mammoth Machine and Design, for sponsoring it and providing necessary resources. The team also acknowledges the contribution of mentor Dr. Terence Fagan for his constant support and guidance throughout the course of the project.

Deliverables

Deliverable	File Name
Prototype Status Review Presentations (PSR)	MAM_STIPPL PSR.pdf
Prototype Review Presentations (PRP)	MAM_STIPPL PRP.pdf
Expo Poster	MAM_STIPPL_POSTER
Project Video	MAM_STIPPL_Video.mp4.mp4
Final Timesheet	MAM_STIPPL_TimeSheet7.xlsx
Progress Report #1	MAM_STIPPL_ProgressReport1.pdf
Progress Report #2	MAM_STIPPL_ProgressReport2.pdf
Project Summary	MAM_STIPPL_ProjectSummary.pdf

Division of Duties Summary Table

	Team Member #1 Talal	Team Member #2 Daniel	Team Member #3 Gokul	Team Member #4 Matthew	Team Member #5 Sergio	Team Member #6 Corey	Total
Fabrication of Parts	0%	5%	0%	5%	5%	85%	100%
Design of X axis (NEW)	0%	30%	0%	30%	10%	30%	100%
Design of Y axis (NEW)	0%	85%	0%	15%	0%	0%	100%
Design of Z axis (NEW)	0%	5%	10%	5%	0%	30%	100%
Design of C axis (NEW)	0%	25%	0%	50%	0%	25%	100%
Design of B axis (NEW)	0%	30%	5%	40%	5%	20%	100%
Machine Controls	50%	5%	0%	45%	0%	0%	100%
LinuxCNC Programming	5%	0%	0%	95%	0%	0%	100%
Machine Kinematics	0%	5%	90%	5%	0%	0%	100%
Documentation	20%	20%	20%	20%	20%	20%	100%

MAM_STIPPL Project – Final Project Report – Senior Design II

Date	Revision	Author	Comments
2018-05-02	-	Team MAM_STIPPL	Original Document

Table of Contents

Overview of this Document

This document describes the design of the Stippling Machine Design project and its end end-product for Senior Design. The project was to design and develop a prototype fully functioning computer numerically controlled multi-axis stippling machine to repeatedly stipple any free form surface with a heating element for an industrial sponsor of the ISL program, Mammoth Machine and Design. The stippling operation involves marking the surface of the workpiece for increasing friction, by surface manipulation. The machine is designed to automate the stippling process, to repeatedly stipple a polymer surface of any arbitrary shape with a heating element on an articulated head to locally melt and form a pattern on the surface.

The timeline of the project was split into two semesters. The team covered the literature review and design phase in the first semester and, manufacturing, control & testing in second semester. Along with design, research on the control part of the machine also commenced. The team started with the design of the stippling head and eventually moved out to rest of the machine, its linear and rotational axes and eventually the frame. Multiple design approaches were considered, and the design team finalized the Moving Bridge Gantry style design at the end of the first semester. In the beginning of the second semester the team met with the supporter who wished that changes to the design be made to ensure the success of the project. The team then decided to move to a stationary gantry style design where the work piece was placed on one axis while the tool was coupled with the gantry design.

Project Overview / Statement of Work Summary

The objective of this project was to design and build a computer numerically controlled stippling machine for Mammoth Machine and Design. The stippling operation involves the surface of the work piece for increasing friction and aesthetics, by surface manipulation. Conventionally, stippling is performed using a heated tool tip by hand or by vaporization using a laser. The machine in this project is to be designed to automate the stippling process, to repeatedly stipple a polymer surface of any arbitrary shape with a heating tool on an articulated head which can locally melt and form a pattern for increased friction, thereby reducing time consumption and effort.

Design Narrative

Machine Design

At the beginning of the year we discussed with Mammoth Machine and Design the machine design. From that discussion, the machine design was overhauled and changes to the x-axis, y-axis, z-axis and c-axis were made. The original design can be seen in Figure 1 while the new machine design can be seen in **Error! Reference source not found.**.

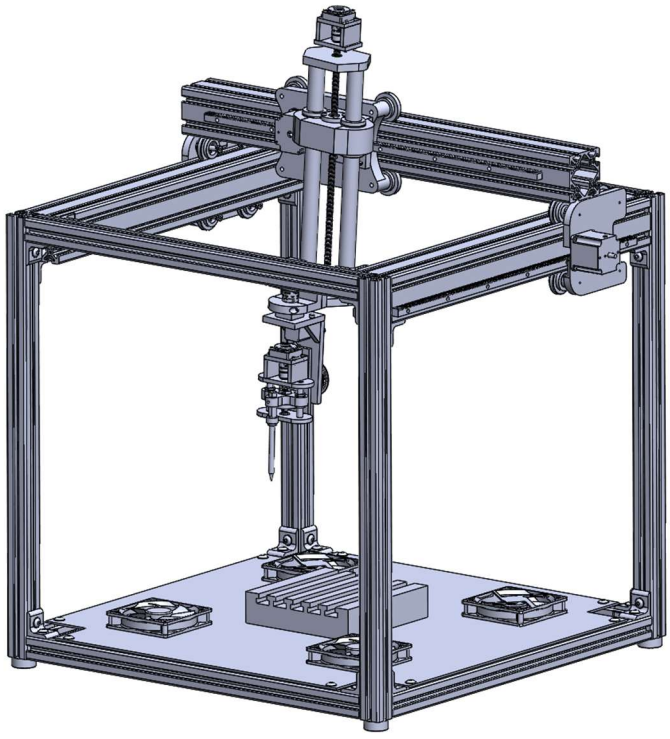


Figure 1 Old Machine Design

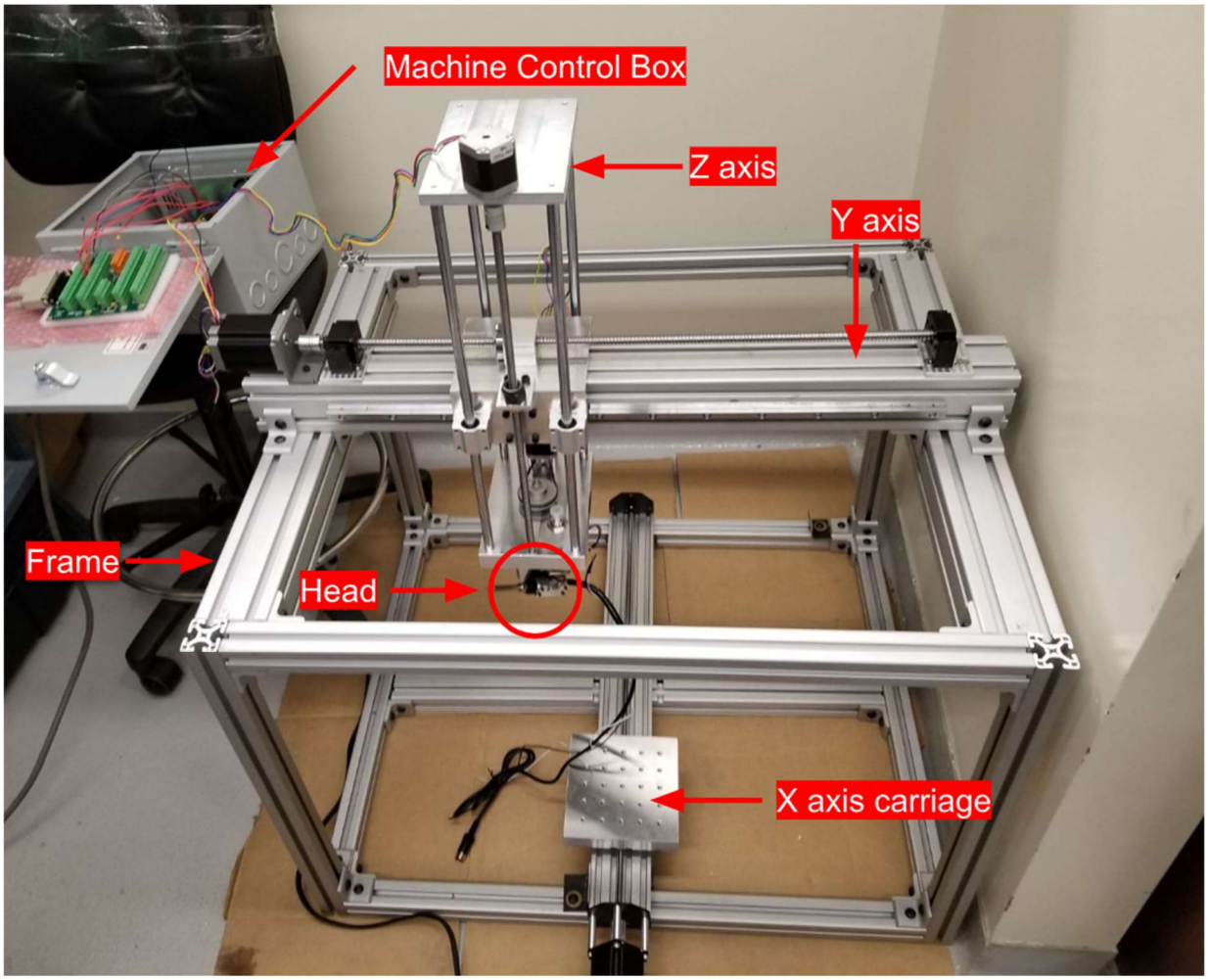


Figure 2 Full Assembled Machine

The first major change was that the X-axis was no longer stacked and moved to be separate from the other stacked axes. The X-axis is seen in Figure 3. The X-axis was then changed to hold the workpiece and would be bought and machined to the correct size. The idea for purchasing the X-axis would allow more time to be devoted to redesigning the other axes

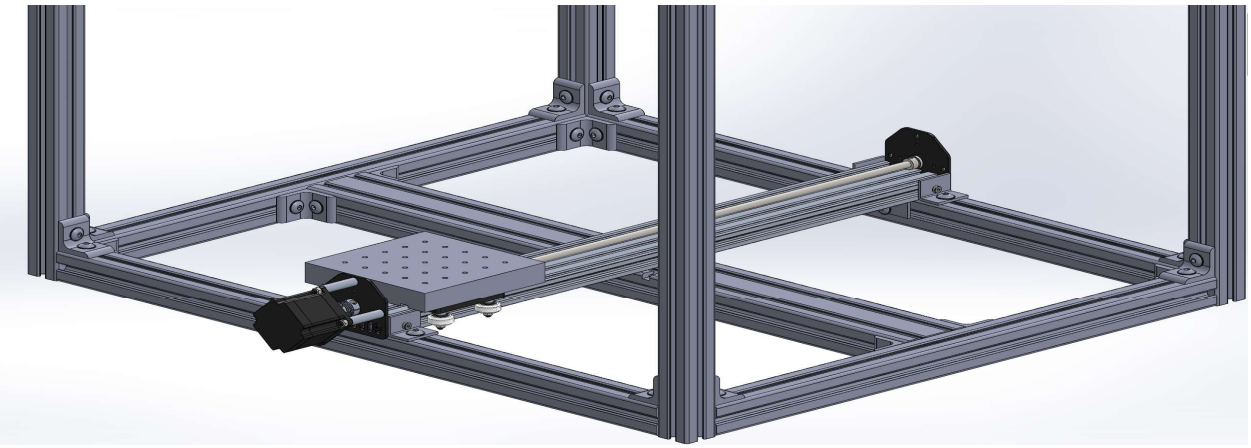


Figure 3 New X-Axis

The x-axis beam is currently machined to length, the carriage is assembled, and the work table has been bought and received. The carriage and the machined rail is shown in Figure 4.

Figure 4 X-Axis Carriage Assembly

The y-axis was changed to a ball screw instead of a rack and pinion and the carriage was changed to be single piece instead of multiple pieces. The old design is seen in Figure 5. The carriage plate for the y-axis is one of the critical components in the machine due to the supporter not having time to machine the carriage, tooling was bought and one of our members machined the carriage plate. The y-axis rail was machined and the linear guide ways were mounted on the rails. The spacer blocks supporting the ball screw bearing blocks were machined and the bracket for the mounting the motor was machined as well. The linear ball bearing housings were mounted onto the carriage as well as the ball screw mount. With all these components mounted to the carriage, the carriage was mounted to the linear guide blocks thus completing the assembly of the y-axis.

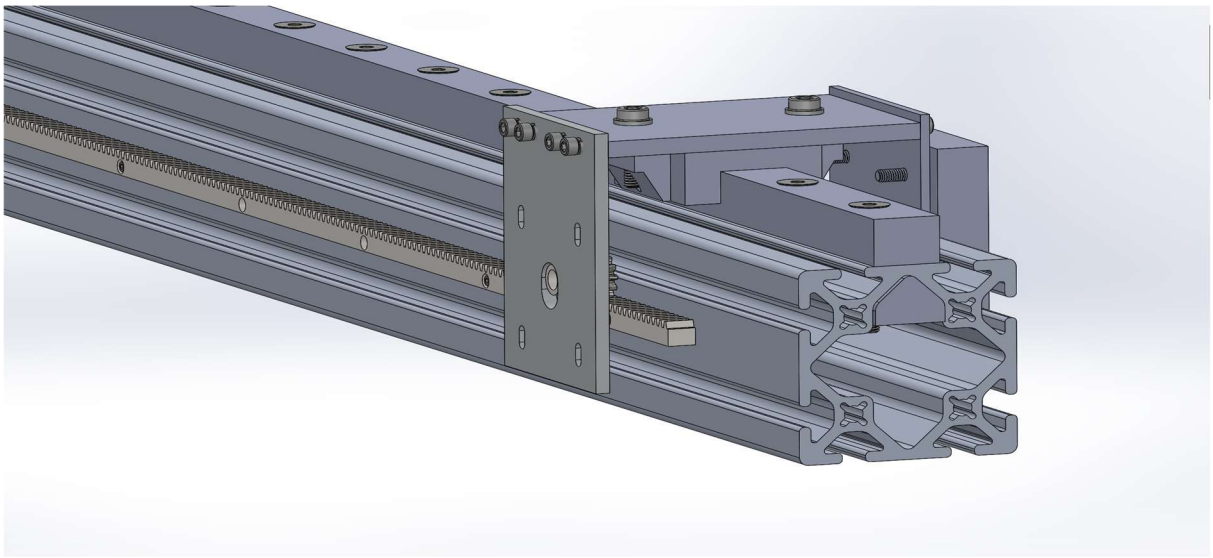


Figure 5 Old Y-Axis Design

The position of one of the linear guide rails was changed to better accommodate the ball screw. The ball screw also provided a better resolution than the rack and pinion, although it is at the cost of running the stepper motor faster. The new innovative design is seen in **Error! Reference source not found.**.

Figure 6 New Y-Axis Design

The Z-axis was changed per request of Mammoth Machine and Design, to use four guide rails as support as opposed to only two that were used as previously. The Z-axis was also changed to a

lead screw instead of a ball screw to decrease the chance of overhauling happening. The old design is seen in **Error! Reference source not found.** while the new design is seen in Figure 8.

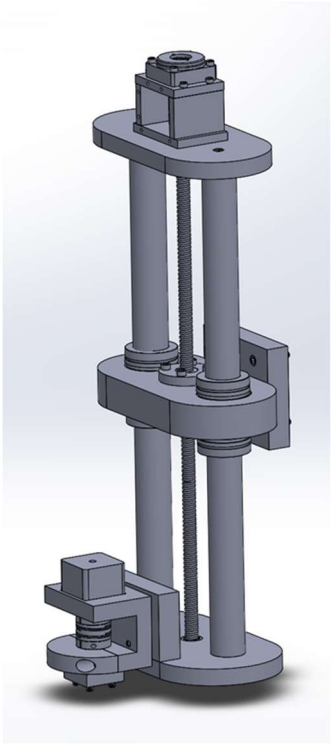


Figure 7 Old Y-Axis Design

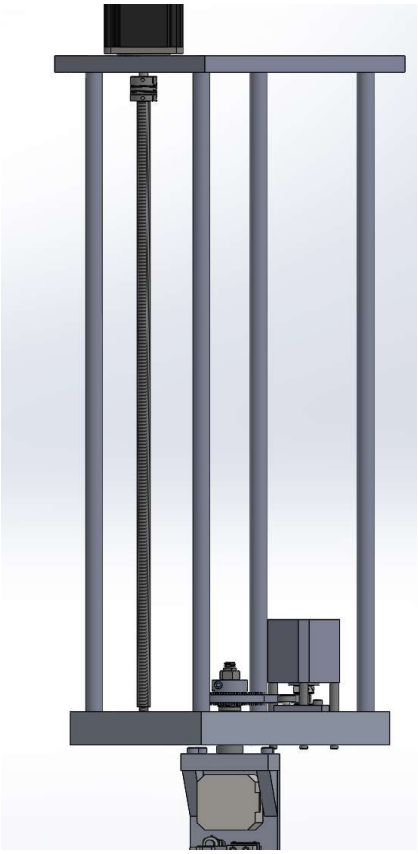


Figure 8 New Z-Axis Design

The original plan for the z-axis guide rods was to drill and tap both ends, so the plates could be fastened to the ends of guide rods. Unfortunately not all the material was the same since one guide rod was drilled and tapped however the other guide rods were hardened on the ends so with the standard tooling available, it was not possible to finish drilling and tapping the other guide rods. To get around this, the new plan was to use set screws hold the guide rods on the side through the plate. A calculation was done to see if the set screw could provide enough force to hold up the weight of the z-axis and the head. The calculation showed that it could provide more than enough force to hold the z-axis and the head. With this in mind the proposed plan went through and has proved to be successful as seen below. To complete the Z-axis, the lead screw nut was machined and the lead screw was cut to length. The end of the lead screw was also turned down to fit into the bearing on the c-axis plate. The component that attached the z-axis to the y-axis was also machined as well. With all of these components completed the Z-axis was mounted to the carriage.

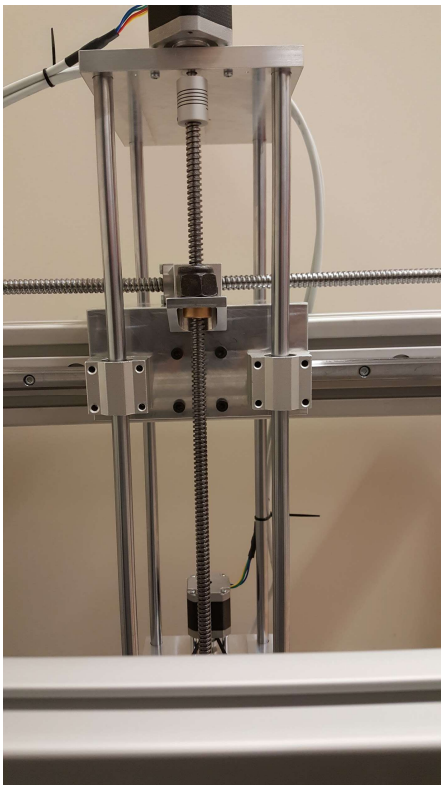


Figure 9 Assembled Z-Axis

The c-axis was changed from a direct drive to a timing pulley design to increase the angular resolution and increase the torque. The c-axis shaft was made using stainless steel after the material that was supposed to be used ended up being too hard to machine with the tooling available. The angular contact bearings were press fit into the c-axis plate by cooling the bearings and pressing them into the plate. The head can be seen in the figure below.

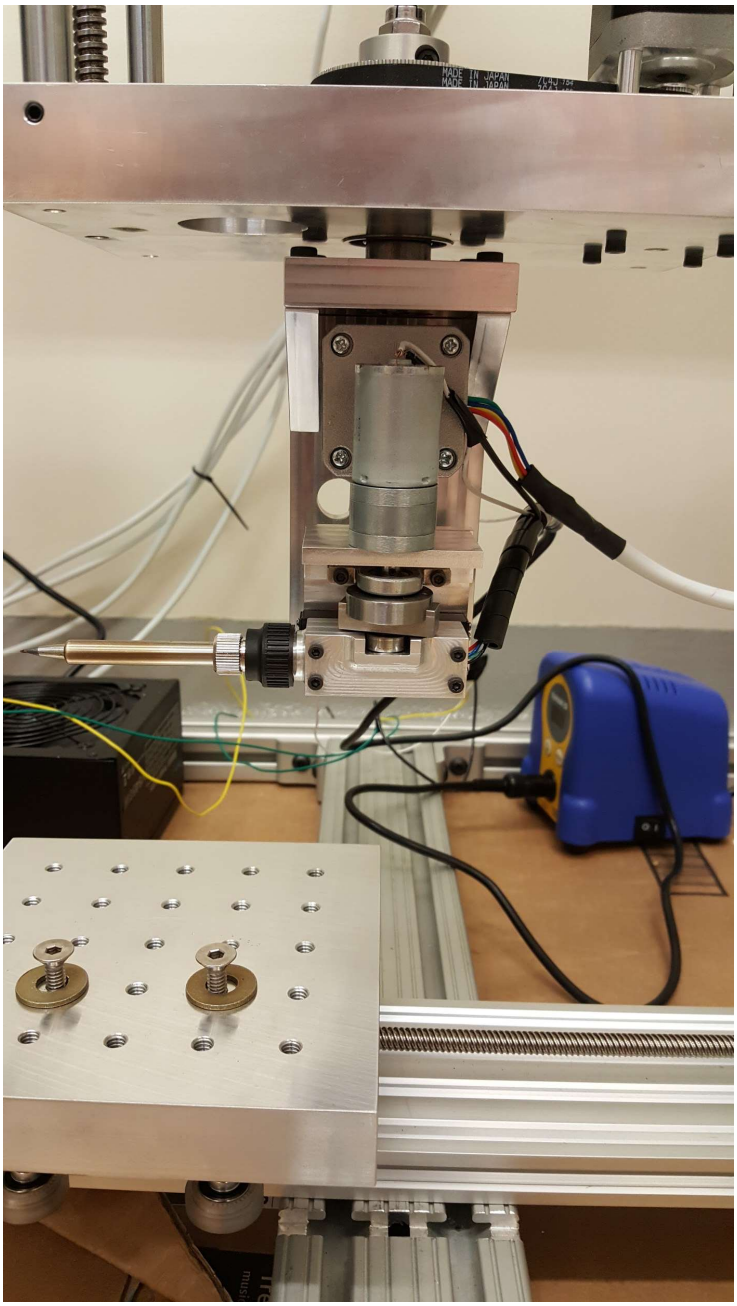


Figure 10 Head Assembly

The head assembly remained unchanged, which used an upside-down L-shaped bracket to connect to the c-axis shaft. The short side of the L-shaped bracket connected to the c-axis shaft and the longer side of the L-shaped bracket was used to give an addition degree of freedom to the head, called the b-axis. A ball bearing was press-fit into the material and was sized to fit a half-inch shaft. The half-inch shaft connected to the assembly which produced the stippling motion. The b-axis shaft was drive by a belt and pulley system, using a 3:1 ratio. The motor for the shaft was mounted in the corner of the L-shaped bracket, allowing the assembly rotating to pass underneath. The stippling motion design implemented a modified type of the crank-slider. A tradition crank-slider uses one arm, the crank, connected to a rotational shaft with a second arm connected to the opposing end of the crank. The second arm provides linear motion to a piston or

slider. For the modified design, the first arm is replaced with a circular crank and has an eccentrically inserted pin. Attached to the pin is a bearing which rides in a slot cut from the object to be moved. The object to be moved was constrained by a linear bearing. As the motor shaft rotates, the eccentric pin moves the object, in the slot, along the path of the linear bearing. The object to be move for this instance is the soldering iron assembly. The soldering iron assembly was modeled after a soldering iron. The handle of the soldering was reverse engineered to accommodate the tip of the tool, which housed the heating element. Using the rest of the soldering iron temperatures could be controlled and the stippling motion created. With the modified crank-slider and soldering iron, high actuation speeds and adjustable temperatures were achieved with a simplistic design.

Machine Kinematics

Inverse kinematics given the endpoint of the structure, what positions do the joints need to be in the achieve that end would be an apt definition. The team started developing inverse kinematics from scratch. Multiple literature sources were looked into which dealt with inverse kinematics of robotics with single degree of freedom joints which included prismatic and revolute joints. After the overall axes and frame design was finalized, a 2D kinematic chain diagram as well as 3D kinematic block diagram was drafted for better understanding of kinematics. This assumed that the kinematic chain of machine tools can be treated like those of robotic manipulators, and the stippling tool can be viewed as the end effector.

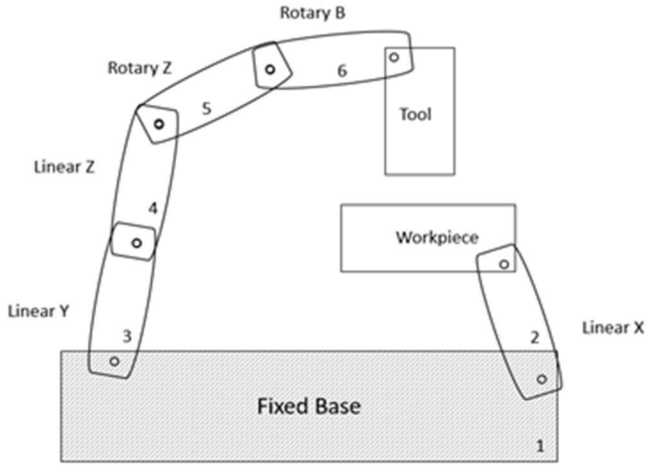


Figure 11: 2D Kinematic Chain Link Diagram Diagram

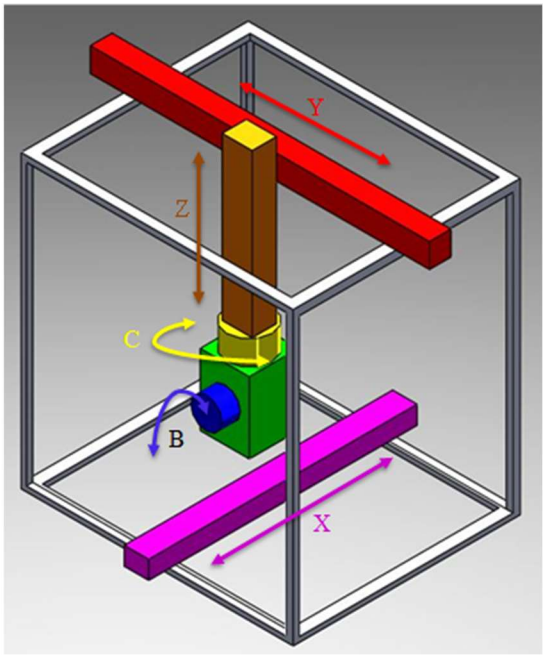


Figure 12: 3D Kinematic Block

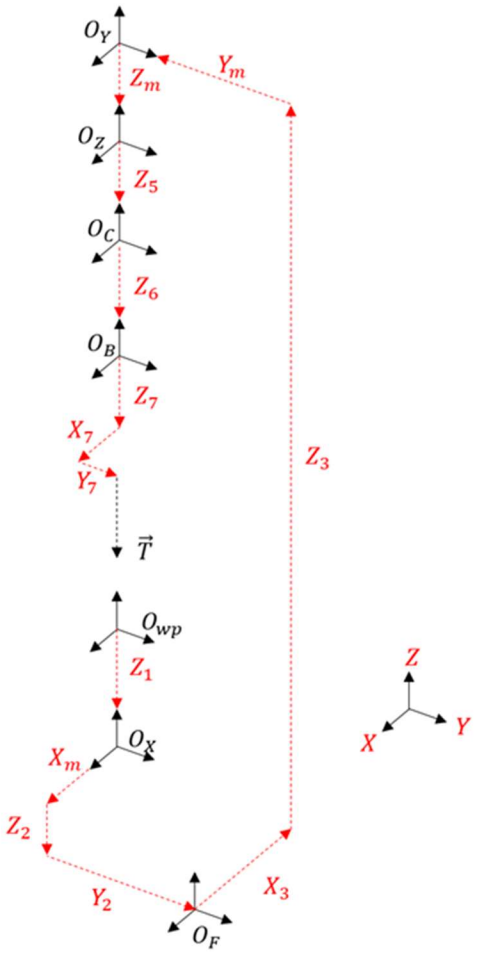


Figure 13: Coordinate Diagram

A closed-chain kinematic model is adopted to represent the machine kinematics. The Denavit–Hartenberg (D–H) convention is used in the modeling of the machine kinematic chain. Referring to Figure 13, assuming the origin are at the center of the axes. Now, starting from the workpiece and moving outward, we observe that, O_{wp} is the origin of the workpiece coordinate system. It is located Z_1 distance offset from the X carriage along the central axis. The origin of X axis, O_x is defined at the center of the workpiece carriage. The X carriage can move in X_m direction. Origin of the frame is located on one of the edges of the frame as O_f. Moving all the way to the top, origin of Y, O_y is located. The Y carriage moves in Y_m direction. The Z axis mounted on the Y carriage can move along Z_m direction. The distance between the origin of Z and origin of C axis is Z_5. Similarly, the offset between C and B is Z_6. To reach the tool axis, the coordinate system must be transformed along Z_7, X_7 and Y_7. T is the tool length or in our case, the stippling tip length.
Now that the coordinate systems were defined, the team obtained the kinematic equations. These equations were essentially homogeneous matrices which enabled the team to convert a position from one coordinate system to another.

CAD/CAM systems are typically used to generate the 3D CAD models of the workpiece as well as the CAM data for input to the CNC 5-axis machine. The tool or cutter location (CL) data, is composed of the cutter tip position and the cutter orientation relative to the workpiece coordinate system. Two vectors, as generated by most CAM systems, contain this information:

$$K = \begin{bmatrix} K_x \\ K_y \\ K_z \\ 0 \end{bmatrix}$$ orientation vector;

$$Q = \begin{bmatrix} Q_x \\ Q_y \\ Q_z \\ 0 \end{bmatrix}$$ position vector;

Translation and Rotation Matrices:

There are four fundamental transformation matrices on which 5-axis kinematics can be based:

$$T = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R(X,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sin\theta & 0 \\ 0 & sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R(Y,\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R(Z,\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix T implies a translation in the X, Y, Z coordinate directions by the amounts T_x, T_y, T_z respectively. The R matrices imply rotations of the angle θ about the X, Y and Z coordinate axes respectively.

Forward Transformation:

The transformation can be defined by the sequential multiplication of the matrices:

$$^{W}_{T}T = {}^{W}_{X}T.\,{}^{X}_{F}T.\,{}^{F}_{Y}T.\,{}^{Y}_{Z}T.\,{}^{Z}_{C}T.\,{}^{C}_{B}T.\,{}^{B}_{T}T$$

with the matrices built up as follows:

$$^{W}_{X}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -Z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \qquad {}^{X}_{F}T = \begin{bmatrix} 1 & 0 & 0 & X_m \\ 0 & 1 & 0 & Y_2 \\ 0 & 0 & 1 & -Z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \qquad {}^{F}_{Y}T = \begin{bmatrix} 1 & 0 & 0 & -X_3 \\ 0 & 1 & 0 & -Y_m \\ 0 & 0 & 1 & Z_3 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$^{Y}_{Z}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -Z_m \\ 0 & 0 & 0 & 1 \end{bmatrix}; \qquad {}^{Z}_{C}T = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & -Z_5 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$^{C}_{B}T = \begin{bmatrix} cos\beta & 0 & sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\beta & 0 & cos\beta & -Z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \qquad {}^{B}_{T}T = \begin{bmatrix} 1 & 0 & 0 & X_7 \\ 0 & 1 & 0 & Y_7 \\ 0 & 0 & 1 & -Z_7 - T_L \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where, X_m, Y_m, Z_m are the commanded positions along X, Y and Z axes respectively. θ & β are the angles of rotation about C and B axes respectively. Other terms with numbers in subscript are fixed dimensions of the model which do not change irrespective of position of axes. These are measured to be as follows in inches:

Dimension	Value	Dimension	Value
Z_1	1.68898	Z_m	0 < 9.208365
Z_2	0.78740	Z_5	0.63435
X_m	-10.84843 < 0 < 10.84843	Z_6	4.43812
Y_2	15.750005	X_7	1.07865
X_3	13.25	Y_7	0.51870
Z_3	24.75	Z_7	1.84730
Y_m	-9.054945< 0 < 9.054945	T_L	2.61134

On multiplication we get,

$$
{}^{W}_{X}T = \begin{bmatrix} cos\theta.cos\beta & -sin\theta & cos\theta.sin\beta & cos\theta.cos\beta.X_7 - sin\theta.Y_7 - (Z_7+T_L)cos\theta.sin\beta + X_m - X_3 \\ sin\theta.cos\beta & cos\theta & sin\theta.sin\beta & sin\theta.cos\beta.X_7 + cos\theta.Y_7 - (Z_7+T_L)sin\theta.sin\beta + Y_2 - Y_m \\ -sin\beta & 0 & cos\beta & Z_3 - Z_1 - Z_2 - Z_m - Z_5 - Z_6 - sin\beta.X_7 - (Z_7+T_L).cos\beta \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

We can now equate the third column of this matrix with our given tool orientation vector K, ie.:

$$
K = \begin{bmatrix} K_x \\ K_y \\ K_z \\ 0 \end{bmatrix} = \begin{bmatrix} cos\theta.sin\beta \\ sin\theta.sin\beta \\ cos\beta \\ 0 \end{bmatrix}
$$

From these equations we can solve for the rotation angles θ, β. From the third row we find:

$$\theta = \cos^{-1}(K_z) \qquad (-180^\circ < \theta < 180^\circ)$$

and by dividing the second row by the first row we find:

$$\beta = \tan^{-1}(K_y\, K_x) \qquad (0 < \beta < 150^\circ)$$

Equating the last column with the tool position vector Q, we can write:

$$
Q = \begin{bmatrix} Q_x \\ Q_y \\ Q_z \\ 0 \end{bmatrix} = \begin{bmatrix} cos\theta.cos\beta.X_7 - sin\theta.Y_7 - (Z_7+T_L)cos\theta.sin\beta + X_m - X_3 \\ sin\theta.cos\beta.X_7 + cos\theta.Y_7 - (Z_7+T_L)sin\theta.sin\beta + Y_2 - Y_m \\ Z_3 - Z_1 - Z_2 - Z_m - Z_5 - Z_6 - sin\beta.X_7 - (Z_7+T_L).cos\beta \\ 1 \end{bmatrix}
$$

This can be rewritten to give:

$$Q_x = cos\theta.cos\beta.X_7 - sin\theta.Y_7 - (Z_7+T_L)cos\theta.sin\beta + X_m - X_3$$

$$Q_y = sin\theta.cos\beta.X_7 + cos\theta.Y_7 - (Z_7+T_L)sin\theta.sin\beta + Y_2 - Y_m$$

$$Q_z = Z_3 - Z_1 - Z_2 - Z_m - Z_5 - Z_6 - sin\beta . X_7 - (Z_7 + T_L) . cos\beta$$

which is the forward transformation of the kinematics.

Inverse Transformation:

We can solve for P from equation (48) directly to get:

$$X_m = -cos\theta . cos\beta . X_7 + sin\theta . Y_7 + (Z_7 + T_L) cos\theta . sin\beta + Q_x + X_3$$

$$Y_m = sin\theta . cos\beta . X_7 + cos\theta . Y_7 - (Z_7 + T_L) sin\theta . sin\beta + Y_2 - Q_y$$

$$Z_m = Z_3 - Z_1 - Z_2 - Z_5 - Q_z - Z_6 - sin\beta . X_7 - (Z_7 + T_L) . cos\beta$$

Programming and Motion Control

The programming of the machine started by creating a simulated machine in the LinuxCNC software using trivial kinematics to model the geometry of the machine. The simulated machine did not require any external hardware and could be tested without risk of damage to a physical machine. This was done by modifying an included simulation configuration files. The INI and HAL files in the directory required the addition of the 4th and 5th rotary axes to simulate the geometry of the machine. A Vismach python script was written to model and simulate the machine geometry with the input of G-code files in LinuxCNC, the code is attached below in Appendix B. The Vismach python script was executed during the program startup, and displayed the simulated machine geometry, allowing the machine movements to be visualized during operation.

After the successful testing and implementation of the simulated machine, and the acquisition of the mesa motion control cards, a finalized set of configuration file were developed and tested. The computer that was purchased to act as the machine controller had an outdated version of the application preinstalled. The computer operating system was restricted to the use of a Realtime kernel to allow for precise motion control, and communication through the field inputs and outputs. The version of LinuxCNC was updated manually on the machine to the most current stable release of the software. But after updating the software it was discovered that the application would hang upon starting. After attempting various methods to mitigate the issue, the original OS reinstalled onto the hard drive due to time constraints. The use of custom machine kinematics required that the version be upgraded the developmental edition to allow for the kinematic files to compiled and executed by the software.

The Mesa 5i25 and 7i76 mother daughter motion control cards were installed on the lower PCI slot of the control pc. The cards were configured to have the breakout board logic power supplied through the field programmable gate array (FPGA) card, the 5i25, which also generates the stepping and control signals for the 7i76 breakout board. The hardware generated signals were advantageous over software generated signals due to the clarity and accuracy of the signals being produced. Without the ability to incorporate a closed loop stepper system to monitor positioning in the design due to budget constraints, the accuracy of the controls signals sent to the stepper motors were critical. The mesa cards required the drivers to be installed and copied to the correct directories such that they were accessible by LinuxCNC. The field i/o power has been selected to be powered externally from a 24V power supply. The motors drivers were configured for each

individual axis to match the current, timing, and resolution requirements of each driving motor. The linear axis motor drivers were configured to operate with a half-step micro stepping of the motor to further decrease the resolution of axes. The rotary axis motor drivers were configured to operate with a one-eighth micro stepping to smooth out the motion and increase the ability to position the soldering tip accurately. These motors attached to the breakout board step and direction outputs and ran a simulated numeric program to verify motor direction and operation speed.

In the linuxCNC software the PNC configurator application was used to determine the maximum velocity and acceleration each subassembly could tolerate dynamically. Each axis was positioned in approximately the center of its travel and ran at incrementally increasing velocities and accelerations between a set distance, where the motor was required to change direction rapidly. The maximum velocity of the axes were determined first incrementally before the maximum acceleration was determined incrementally using the experimentally determined maximum velocity. An increment of 2 machine units were used for the determination of these limits.

The machine homing process required the use of a single limit switch on each of the linear axes, and a single PNP proximity sensor on the rotary axes. On the linear axes two limit switches were placed at each end of the axis, and on the rotary axes the proximity sensors were placed at key locations to allow for homing and limits to monitored. The linear axes homing process involved programming the axes to travel towards and trip the limit switch traveling at a high federate. The axes were then backed of the switch, and the again moved towards the switch a slower velocity to allow for a more accurate machine position to be determined. The axes were then translated to the predetermine origin location.

The machine kinematic file was written in the language C and can be seen below in Appendix B. The kinematic file was required to be precompiled using the “halcomplie” command, where the file was then compiled and stored into the appropriate location and format for interpretation by the software. The file incorporates the use of packages to preform base calculations required for determining the machine positioning. The current version of the kinematics are incorrect as they do not allow for the rotation of the C axis. This may be due to the tool tip to be modeled by the U,V, and W position axes even though they are not present the physical design of the machine.

Test Results

Positional Accuracy

The project specifications demanded precision of 0.01 inches and working volume is 150mm x 150mm x 150mm. Taking into consideration the size of the machine and consulting Dr. Jimmie Miller from the Center for Precision Metrology Department of UNC Charlotte, the team decided to go with Laser Interferometry for testing the machine.

Below shown is the test setup for X, Y and Z axis precision measurement. This same setup can be used to verify the linear range of each axis.

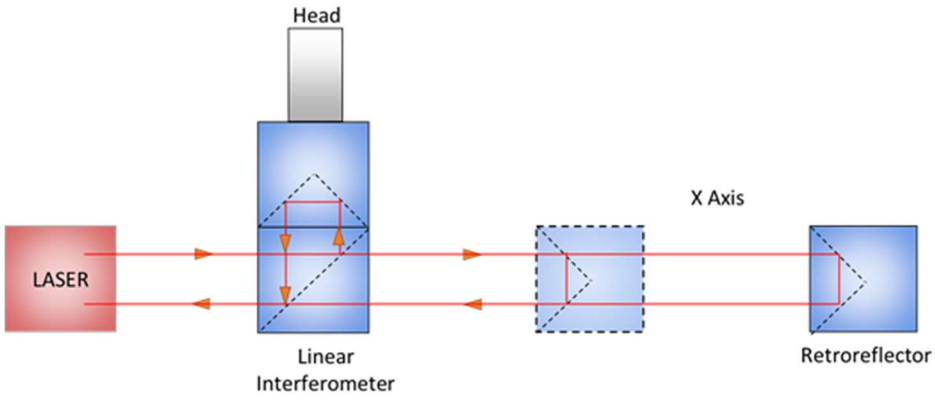


Figure 14: X – Axis Test Setup Plan

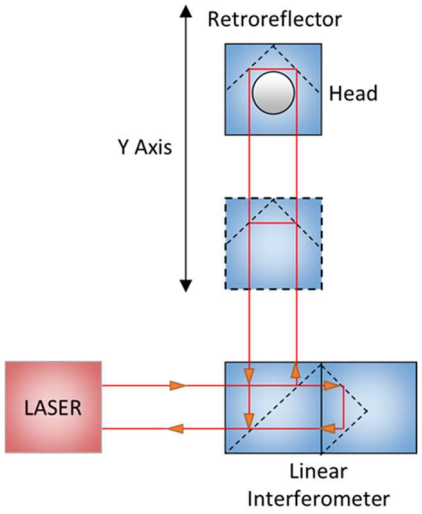


Figure 15: Y – Axis Test Setup Plan

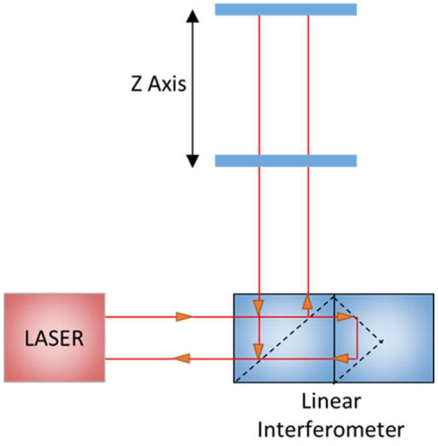


Figure 16: Z – Axis Test Setup Plan

The setup consists of a Laser beam generated by laser source manufactured by Automated Precision Inc. The optics consist of a linear interferometer cube and a retroreflector cube. The linear interferometer splits the laser beam into two parts, one is reflected to the retroreflector and

the other is reflected to the laser source. The retroreflector is attached to the moving carriage/head and the linear interferometer is held stationary. The return beam from the interferometer combines with the beam reflected from the retroreflector, back into the laser aperture. The laser with the help of software analyzes both these beams and depending upon the change in wavelength due to motion of retroreflector, it detects the error.

Laser System Specifications:
Max O/P - 1MW
Class – II
Pulse Spec - Continuous Wave
Laser Medium – Helium Neon
Max Axis Velocity – 0.7m/s
Resolution - 1nm

The team efforts to proceed with this test plan was hindered by the carriage capacity of the mounting bed to mount the machine. Due to the size of the machine, the team was unable to perform the test on the mounting bed. Any attempts to perform the test on ground level was futile because, the laser arrangement was kept on the floor too which made vertical adjustments not possible.
As the team found that, laser interferometer was not feasible due to lack of proper equipment, the team moved to more crude method of testing. This was performed using an inch dial gauge. The dial gauge had precision of 0.01 inches which was apt for testing the machine. The team fixed the dial gauge to a magnetic base and pushed it against X axis carriage. The machine was then commanded to move 0.01 inches in both direction. This was verified with the reading of the dial gauge at multiple intervals over the complete range.

Figure 17: Dial Gauge Test Setup

The team would recommend testing the precision and linear accuracy of the machine by using laser interferometry according to above set up, at testing phase of the final product.

Adequate Ventilation
The ventilation of the machine was determined by visually monitoring the air quality inside the working are of the machine during the stippling process. The fumes produced by the process were minimal and quickly dispersed by the air currents passing through the work area of the machine due the open work volume.

Operational Temperature

The operational tool tip temperature of the stippling mechanism was to operate at a temperature of less than 320°C. The temperate of the soldering iron was originally to be determined using a laser thermometer, using the temperature dependent wavelength shift principle to determine the temperature of an object. This method first tested to measure the temperature but was found to produce inaccurate results due to the small surface area of the tip, and the reflectivity of the surface. Thus, the thermocouple housed in the soldering iron was verify the tip temperature from the temperature reading on the station readout.

Stippling Mechanism Force

The stippling force requirement was experimentally determined during the first design phase of the machine by monitoring the average force required to stipple a generic polymer surface at varying rates using a scale. The force data was then statistically analyzed to generate a 98% confidence interval around the mean force required for stippling. From the data there is a 98% confidence that the average force lies between 0.42 to 0.5 pounds of force. The stippling design of the machine was designed to produce twice the required force. From testing of the stippling mechanism of the machine the force being produced adequately meets this requirement from the ability of the mechanism to successful stipple a polymer surface, this can be seen in Figure 18 below. The machine also exceeds this requirement due to the mechanism's ability to plastically stipple an aluminum surface. This was found due an incident involving the work table and soldering tip of the machine.

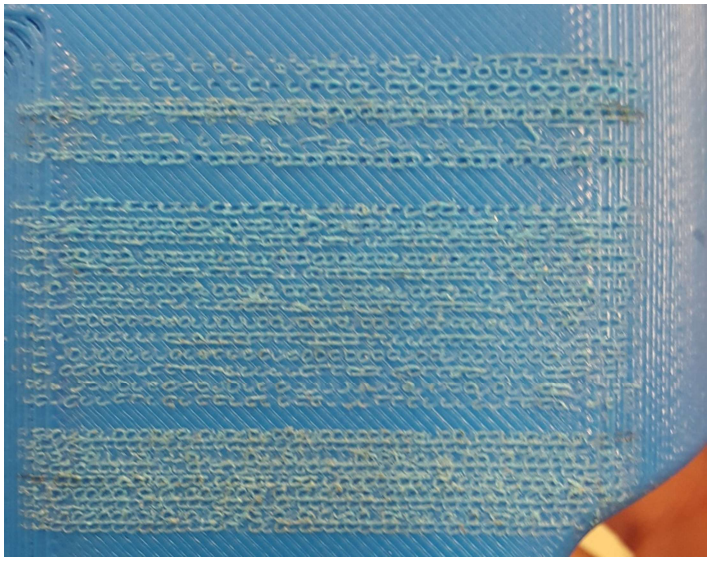


Figure 18 Stippling Test

Evaluation of Prototype as Compared to Project Performance Specification Document

The major project specifications originally considered for the design of the machine are mentioned in the Specifications Document[1]. Following are the specifications:

- PS1: The working area of the machine must be 150mm × 150mm × 150mm (or 75mm).
    - The team was ineffective in verifying PS1 using laser interferometry. However, design of the axes indicates the desired range is met.

- PS2: Adequate ventilation to remove harmful gases from the stippling process.
    - This was achieved through the open design of the machine.
- PS3: The operational temperature of the tip will be less than 320°C.
    - The modification of stippling head and including a temperature-controlled stippling tip from soldering iron, the stippling tip temperature can be varied between 100°C to 450°C, meeting this specification.
- PS4: The stippling mechanism should have adequate force to stipple a variety of polymers.
    - This was proved by stippling two types of polymers successfully.
- PS5: Precision of the machine should be less than or equivalent to 0.01 inches.
    - This was verified using Inch Dial Indicator for X-axis in both directions at multiple instances across the axis.

Recommendations for Further Development

One major issue we noticed on during Z-axis motion was that whole z-axes and head assembly would vibrate during translation due to the play in the self-aligning bearings. The vibrations could be mitigated by incorporating at least one rigid bearing into the y-axis carriage assembly. The current Z-axis design was over constrained kinematically due to the four guide rails used for constraint. By implementing one fixed alignment bearing or bushing to the design would allow for the axis to be properly constrained and remove the play in the design of the current bearings.

Another issue that draws concern it the heat transfer from the soldering iron into the crank slider mechanism. This raises concerns due to the maximum operating temperature of the ball bearings in the telescoping slide are limited to 180°F and the soldering iron temperature needs to be at least 280°F, therefore the telescoping slide is capable of reach 180°F if proper precaution is not taken. Three possible solutions were found to mitigate this heat transfer from the soldering iron to the bearing. The first solution would be to add an appropriately sized heatsink to the soldering iron housing, to reduce the amount of heat transfer to the bearing by additional convective heat transfer. The second solution is to change the soldering iron housing material to a less conductive material, effectively reducing the rate of heat transfer through the bearing. The third proposed solution was to implement a ceramic heat shield between the soldering iron housing and the telescopic bearing. As the ceramic would act as insulator and lower the amount of heat transferred to the bearing.

The last major design change that could be implemented is the complete removal of the linear rods in the Z-axis subsystem. An alternative method of constraint while allowing translation would be constraining the axis using a linear rail and linear guide blocks. One more option to improve the rigidity of the machine would be to replace the aluminum frame with a more rigid alternative albeit at the cost of weight and price.

Impact

Stippling can be done using two methods; one method is laser vaporization while the other method is done by hand using a soldering iron and indenting the material. Stippling parts by hand is a long and tedious process that can take up to eight hours. Stippling is mostly done on handgun grips which releases fumes when the plastic melts during the stippling process. The fumes when

breathed may be nauseating so stippling has to be done in a ventilated area. The automation of the stippling process can help people who stipple by hand by increasing the number of stippled grips and won't pose a risk to the health of the person stippling. Moreover, stippling can be applied to myriad of other applications both for increasing function and aesthetic qualities.

Bill of Materials (BOM)

Table 1 Frame subassembly

Vendor	Description	QTY	Unit Price	Price
80/20	1.50” X 1.50” Lite Smooth Surface T-Slotted Profile - Four Open T-Slots Length(Inches) 82	1	$ 38.85	$ 38.85
80/20	3.00" X 3.00" Lite Smooth T-Slotted Profile - Eight Open T-Slots Length(Inches) 90	1	$ 110.60	$110.60
80/20	1.50” X 1.50” Lite Smooth Surface T-Slotted Profile - Four Open T-SlotsLength(Inches) 100	2	$ 46.95	$ 93.90
80/20	Bolt Assembly: 5/16-18 x .625" Black BHSCS and Slide-In Economy T-Nut - Offset Thread - Black Zinc	87	$ 0.50	$ 43.50
80/20	1.50” X 1.50” Lite Smooth Surface T-Slotted Profile - Four Open T-Slots Length(Inches) 92	1	$ 43.35	$ 43.35
80/20	15 & 40 Series M4 Standard Drop-in T-Nut	10	$ 0.74	$ 7.40
80/20	15 Series & Ready Tube 2 Hole - Inside Corner Bracket	41	$ 2.95	$120.95
80/20	1.50” X 3.00” Lite vs Smooth Surface T-Slotted Profile - Six Open T-Slots LENGTH:30.25 IN	1	$ 30.58	$ 30.58
80/20	1/4-20 Economy Drop-in T-Nut	4	$1.05	$ 4.20
Total				**$493.33**

Table 2 X-Axis Subassembly

Vendor	Description	QTY	Unit Price	Price
Open Builds	V-Slot® NEMA 23 Linear Actuator Bundle (Lead Screw)	1	$ 163.95	$163.95
McMaster-Carr	Low-Carbon Steel 90 Degree Angle1/4" Wall Thickness, 1" x 1" Outside Size-1FT	1	$ 7.50	$ 7.50
80/20	Bolt Assembly: 5/16-18 x .625" Black BHSCS and Slide-In Economy T-Nut - Offset Thread - Black Zinc	4	$ 0.50	$ 2.00
McMaster-Carr	Highly Machinable MIC6 Aluminum Sheet1" Thick, 6" x 6"	1	$ 35.77	$ 35.77
Total				**$209.22**

Table 3 Y-Axis Subassembly

Vendor	Description	QTY	Unit Price	Price
Amazon	uxcell CNC Motor Shaft Coupler 6.35mm to 8mm Flexible Coupling 6.35x8mm	1	$ 9.16	$ 9.16
Automation Overstock	BLH15mm size linear rail|BLH15R Rail Sections: - 601mm-700mm|exact length (mm): - 690	2	$ 46.48	$ 92.96
Automation Overstock	15mm flange-type carriage (= Hiwin HGW15CCZ0C)	2	$ 24.00	$ 48.00
Bang good	SC12UU Metal 12mm Linear Ball Bearing Motion Bearing For CNC	4	$ 4.99	$ 19.96
Bang good	Machifit SFU1204 700mm Ball Screw with BK10 BF10 End Support and 6.35x10mm Coupler CNC Tool	1	$ 63.69	$ 63.69
Bang good	Ball Screw Nut Housing Seat Mount Bracket Holder for SFU1204 Ball Screw	1	$ 5.59	$ 5.59
McMaster-Carr	Low-Carbon Steel 90 Degree Angle3/16" Wall Thickness, 2-1/2" x 2-1/2" Outside Size	1	$ 7.62	$ 7.62
McMaster-Carr	Black-Oxide Alloy Steel Socket Head ScrewM5 x 0.8 mm Thread, 15 mm Long	1	$ 14.72	$ 14.72
McMaster-Carr	316 Stainless Steel Hex Drive Flat Head Screw82 Degree Countersink Angle, 1/4"-20 Thread Size, 3/4" Long	1	$ 4.24	$ 4.24
McMaster-Carr	Zinc-Plated Alloy Steel Socket Head Screw8-32 Thread Size, 7/8" Long	1	$ 5.79	$ 5.79
Online Metals	6 inch Extruded Aluminum Bare Square 6061 T6511 CUSTOM LENGTH: 3.25	1	$ 61.59	$ 61.59
80/20	15 & 40 Series M4 Standard Drop-in T-Nut	10	$ 0.74	$ 7.40
Total				**$340.72**

Table 4 Z-Axis Subassembly

Vendor	Description	QTY	Unit Price	Price
Amazon	Outer Diameter 12mm x 500mm Cylinder Liner Rail Linear Shaft Optical Axis	4	$ 27.35	$109.40
Roton	Acme Lead Screw, 3/8 X .125, RH, Steel LENGTH:18 INCHES	2	$ 17.16	$ 34.32
Roton	Acme Sleeve Nut, 3/8 X .125, RH, Bronze	1	$ 22.78	$ 22.78
Online Metals	1" x 6" Extruded Aluminum Bare Rectangle 6061 T6511CUSTOM LENGTH: 7.75	1	$ 30.77	$ 30.77
McMaster-Carr	Stainless Steel Ball BearingShielded, Trade Number R168-2Z	1	$ 5.28	$ 5.28
Online Metals	1" x 2" Cold Finish Mild Steel Rectangle 1018: LENGTH 2.125	1	$ 4.50	$ 4.50
Online Metals	0.625" x 6" Extruded Aluminum Bare Rectangle 6061 T6511CUSTOM LENGTH: 7.75	1	$ 22.94	$ 22.94
McMaster-Carr	Black-Oxide Alloy Steel Socket Head Screw10-32 Thread Size, 1-1/4" Long	1	$ 11.79	$ 11.79
McMaster-Carr	High-Strength Steel Nylon-Insert Locknut .Grade 8, 5/8"-18 Thread Size	1	$ 3.43	$ 3.43
Total				**$245.21**

Table 5 C-Axis Subassembly

Vendor	Description	QTY	Unit Price	Price
McMaster-Carr	Angular-Contact Ball BearingOpen, Trade Number 7201, for 12mm Shaft Diameter	2	$ 31.78	$ 63.56
McMaster-Carr	Low-Carbon Steel Round Shim, 0.25" Thick, 1/2" ID	1	$ 5.13	$ 5.13
McMaster-Carr	Rotary Shaft1566 Carbon Steel, 12 mm Diameter, 200 mm Long	1	$ 7.71	$ 7.71
McMaster-Carr	Wave Disc Spring0.393" Minimum ID, 0.662" Maximum OD, 0.012" Thick	1	$ 6.43	$ 6.43
McMaster-Carr	Mil. Spec. 18-8 Stainless Steel Thin Hex Nut3/8"-16 Thread Size, MS-35691-19	1	$ 3.07	$ 3.07
SDP-SI	2 mm (GT2) Pitch,80 Teeth, 0.3125" Bore, No Flange / Fairloc Hub, Aluminum Alloy Timing Pulley for .236 (6MM)" Wide Belt	1	$ 23.70	$ 23.70
SDP-SI	2 mm (GT2) Pitch,15 Teeth, 0.1875" Bore, 2 Flanges / Fairloc Hub, Aluminum Alloy Timing Pulley for .236 (6MM)" Wide Belt	1	$ 19.54	$ 19.54
SDP-SI	2 mm (GT2) Pitch, 125 Teeth, 6mm wide Single Sided Neoprene Belt with Fiberglass Cords	1	$ 5.44	$ 5.44
SDP-SI	2 mm (GT2) Pitch, 127 Teeth, 6mm wide Single Sided Neoprene Belt with Fiberglass Cords	1	$ 5.46	$ 5.46
SDP-SI	2 mm (GT2) Pitch, 128 Teeth, 6mm wide Single Sided Neoprene Belt with Fiberglass Cords	1	$ 5.47	$ 5.47
McMaster-Carr	Aluminum Unthreaded Spacer 6 mm OD, 22 mm Long, for M3 Screw Size	4	$ 1.85	$ 7.40
McMaster-Carr	Black-Oxide Alloy Steel Socket Head Screw M3 x 0.5 mm Thread, 50 mm Long	1	$ 3.49	$ 3.49
McMaster-Carr	Ball Bearing Shielded, Trade No. R3-2Z, for 3/16" Shaft Diameter	1	$ 6.56	$ 6.56
McMaster-Carr	316 Stainless Steel Round Shim 0.125" Thick, 3/8" ID	1	$ 6.88	$ 6.88
McMaster-Carr	18-8 Stainless Steel Flex-Top Locknut for Heavy Vibration 5/16"-18 Thread Size	1	$ 2.90	$ 2.90
Total				**$ 172.74**

Table 6 Head Subassembly

Vendor	Description	QTY	Unit Price	Price
McMaster-Carr	Telescoping Slide	1	$ 102.67	$102.67
McMaster-Carr	Head Mounted Bearing	1	$ 15.14	$ 15.14
McMaster-Carr	Head Rotating Bearing	1	$ 15.60	$ 15.60
McMaster-Carr	Rotational A Bearing	1	$ 13.22	$ 13.22
McMaster-Carr	4-40 5/16 Socket Head Bolt	1	$ 8.25	$ 8.25
McMaster-Carr	4-40 7/8 Socket Head Bolt	1	$ 10.30	$ 10.30
McMaster-Carr	8-32 9/16 Socket Head Bolt	1	$ 9.14	$ 9.14
McMaster-Carr	10-32 5/8 Socket Head Bolt	1	$ 10.28	$ 10.28
McMaster-Carr	M3 14mm Socket Head Bolt	1	$ 10.57	$ 10.57
McMaster-Carr	M3 9mm Flathead Bolt	1	$ 4.81	$ 4.81
McMaster-Carr	2-56 3/32 Set Screw	1	$ 14.55	$ 14.55
McMaster-Carr	1144 Carbon Steel Rod, High-Strength, 1-1/4" Diameter	1	$ 15.83	$ 15.83
SDP-SI	Rotational A Upper Pulley	1	$ 19.54	$ 19.54
SDP-SI	Rotational A Lower Pulley	1	$ 20.87	$ 20.87
SDP-SI	Rotational A Idler Pulley	1	$ 10.58	$ 10.58
SDP-SI	Rotational A Belt	1	$ 5.16	$ 5.16
SDP-SI	Rotational A Shaft	1	$ 21.81	$ 21.81
SDP-SI	2 mm (GT2) Pitch, 92 Teeth, 6mm wide Single Sided Neoprene Belt with Fiberglass Cords	1	$ 5.18	$ 5.18
SDP-SI	2 mm (GT2) Pitch, 93 Teeth, 6mm wide Single Sided Neoprene Belt with Fiberglass Cords	1	$ 5.19	$ 5.19
Total				**$318.69**

Table 7 BOM for all Subassemblies

Sub Assembly	Total
Electronics	$1,018.15
Frame	$ 493.33
Y-Axis	$ 340.72
Head	$ 318.69
Z-Axis	$ 245.21
X-Axis	$ 209.22
C-Axis	$ 172.74
Total	**$2,798.06**

Budget

The total cost to produce the machine is $2933.30 where the breakdown for the various subassemblies are shown in Table 8. The majority of the budget went to the electronics which are shown in Table 10 and the tools/miscellaneous breakdown is shown in Table 9.

Table 8 Total Cost Breakdown

Sub Assembly	Total
Electronics	$1,018.15
Frame	$ 493.33
Y-Axis	$ 340.72
Head	$ 318.69
Z-Axis	$ 245.21
X-Axis	$ 209.22
C-Axis	$ 172.74
Tools/Misc	$ 135.24
Total	**$2,933.30**

Pro

Table 9 Tools and Misc. Costs

Vendor	Description	QTY	Unit Price	Price
McMaster-Carr	General Purpose Tap for Through-Hole Threading, 2-56 Thread Size	1	$ 10.20	$ 10.20
McMaster-Carr	TiN Coated High-Speed Steel Square-End End Mill4 Flute, 1/2" Mill Diameter, 5" Overall Length	1	$ 37.07	$ 37.07
McMaster-Carr	Hex L-Key0.035" Size, 1-5/16" Overall Length	1	$ 0.24	$ 0.24
Amazon	18/4 AWG wire	1	$ 29.97	$ 29.97
Amazon	Red High Strength Threadlocker	1	$ 11.88	$ 11.88
Amazon	Magnets	1	$ 17.99	$ 17.99
Amazon	Sheet Metal Box with Knockout and Hinged Cover, 10" Width x 10" Height x 6" Depth	1	$ 27.89	$ 27.89
Total				**$ 135.24**

Table 10 Electronics Cost Breakdown

Vendor	Description	QTY	Unit Price	Price
Probotix	LinuxCNC Control PC	1	$389.95	$ 389.95
MESA Electronics	7I76-5I25 Plug-N-Go Kit	1	$199.00	$ 199.00
Digikey	Raspberry Pi	1	$40.00	$ 40.00
Amazon	Digital Soldering Iron	1	$ 95.72	$ 95.72
Amazon	Digital Stepper Driver 1.0-4.2A 20-50VDC for Nema 17, 23, 24 Stepper Motor	2	$ 33.95	$ 67.90
Amazon	Digital Stepper Driver 1.8-5.6A 20-50VD	2	$ 39.95	$ 79.90
Amazon	Digital Stepper Driver 0.3A-2.2A 18-30VD	1	$ 18.99	$ 18.99
Stepper Online	Switching Power Supply 350W 24V 14.6A	1	$ 31.97	$ 31.97
Amazon	TEMCo 6 QTY Plunger Limit Switch Tools	1	$ 30.60	$ 30.60
Amazon	CNC Touch Plate	1	$ 13.99	$ 13.99
Pololu	GameSir G3w Wired Gaming Controller for PC	1	$ 17.49	$ 17.49
Amazon	9.7:1 Metal Gearmotor 25Dx48L mm HP 12	1	$ 21.95	$ 21.95
Amazon	Baomain Red Sign Emergency Stop Push Button Weatherproof Pushbutton Switch 6	1	$ 10.69	$ 10.69
Total				**$1,018.15**

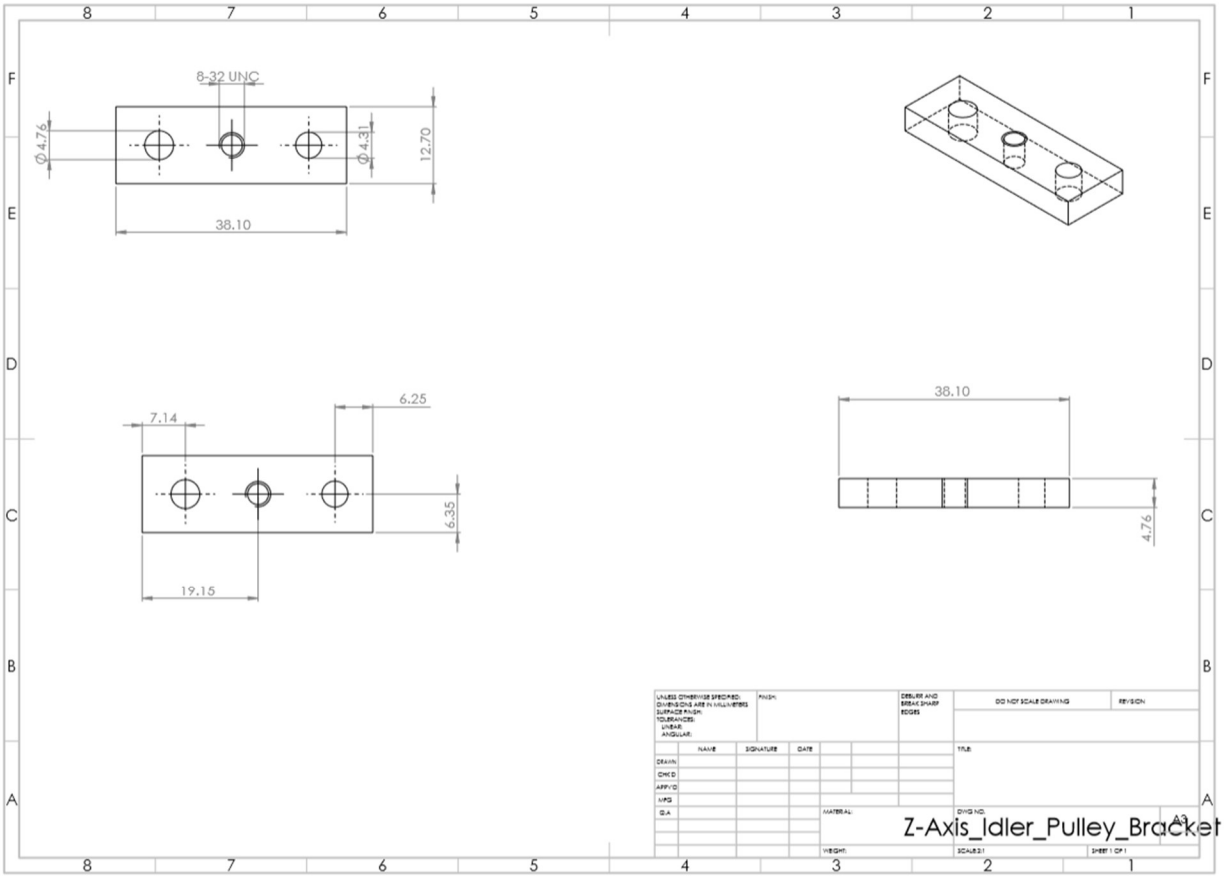
Conclusions

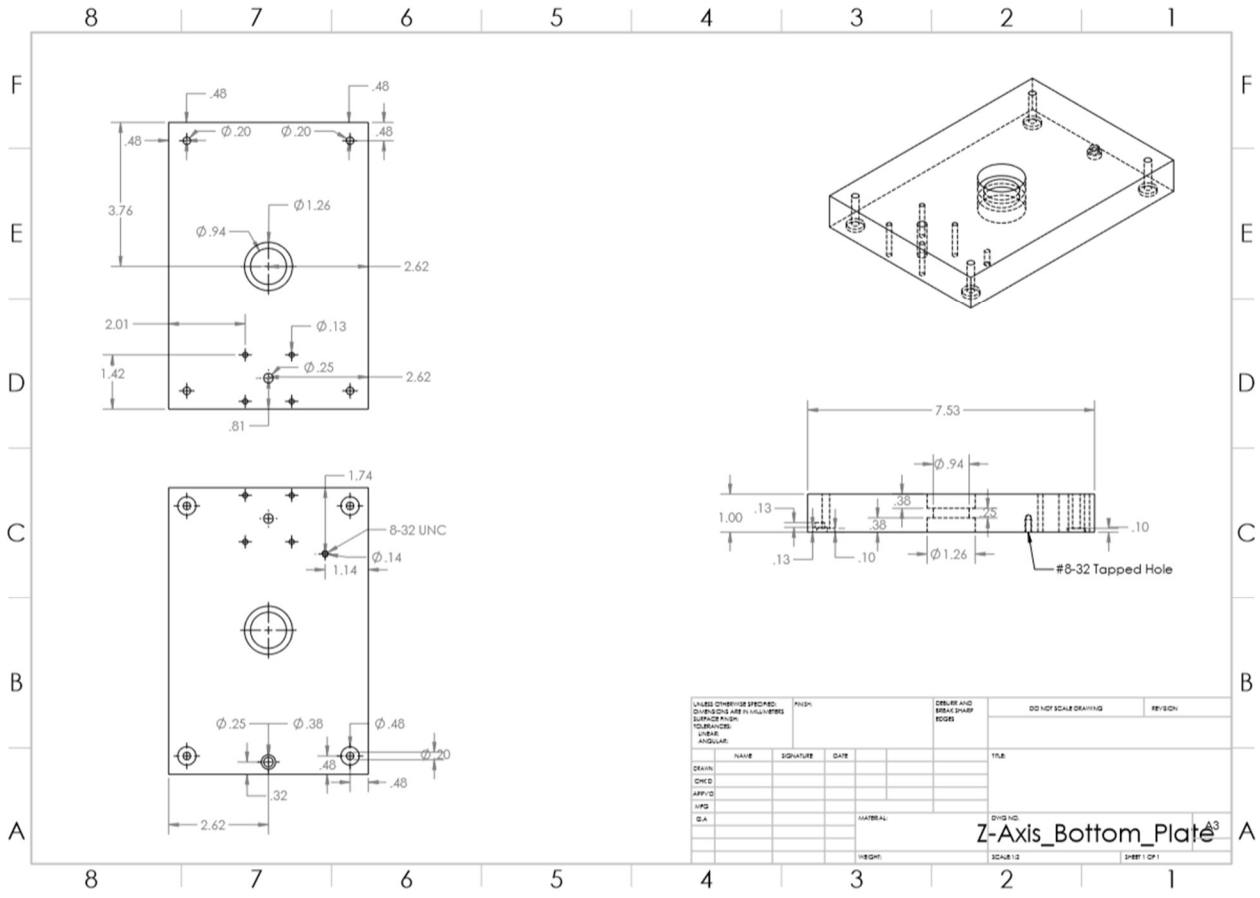
The purpose of the project was to design a stippling machine capable of stippling any freeform surface. The machine was to have a working volume of 150mm x 150mm x150mm, a precision of 0.010 inches, adequate ventilation, a temperature limit of 320°C, and adequate force to stipple a wide range of polymers. The project succeeded in the design and fabrication of a stippling machine that can stipple any freeform surface. The machine can stipple the desired working volume, had adequate ventilation, adequate force to stipple a wide variety of polymers, and was able to reach the temperature limit of 320°C. The machine designed was a 5-axis stationary gantry style machine with an articulated head. Process began will selecting possible options that could achieve the goal. After a process of elimination, the gantry system was selected. CAD designs were the next step. A CAD design was made for every component on the machine. After final CAD designs were made, the machining of parts came next. Next came the assembly which included putting together the frame, X, Y, Z Axes, and the head assembly. Once all subassemblies were put together, the software and electrical components were added. Testing different parts of the machine came last. Each axis along with the stippling head was moved independently. In the end the machine resulted to be about 70% faster compared to the tradition by hand method. The machine was able to produce a stippled pattern across a polymer surface using a G-code inputs. The quality of the stippling was able to be adjusted by changing the workpiece offset, tooltip temperature, and stroke length of the crack side mechanism.

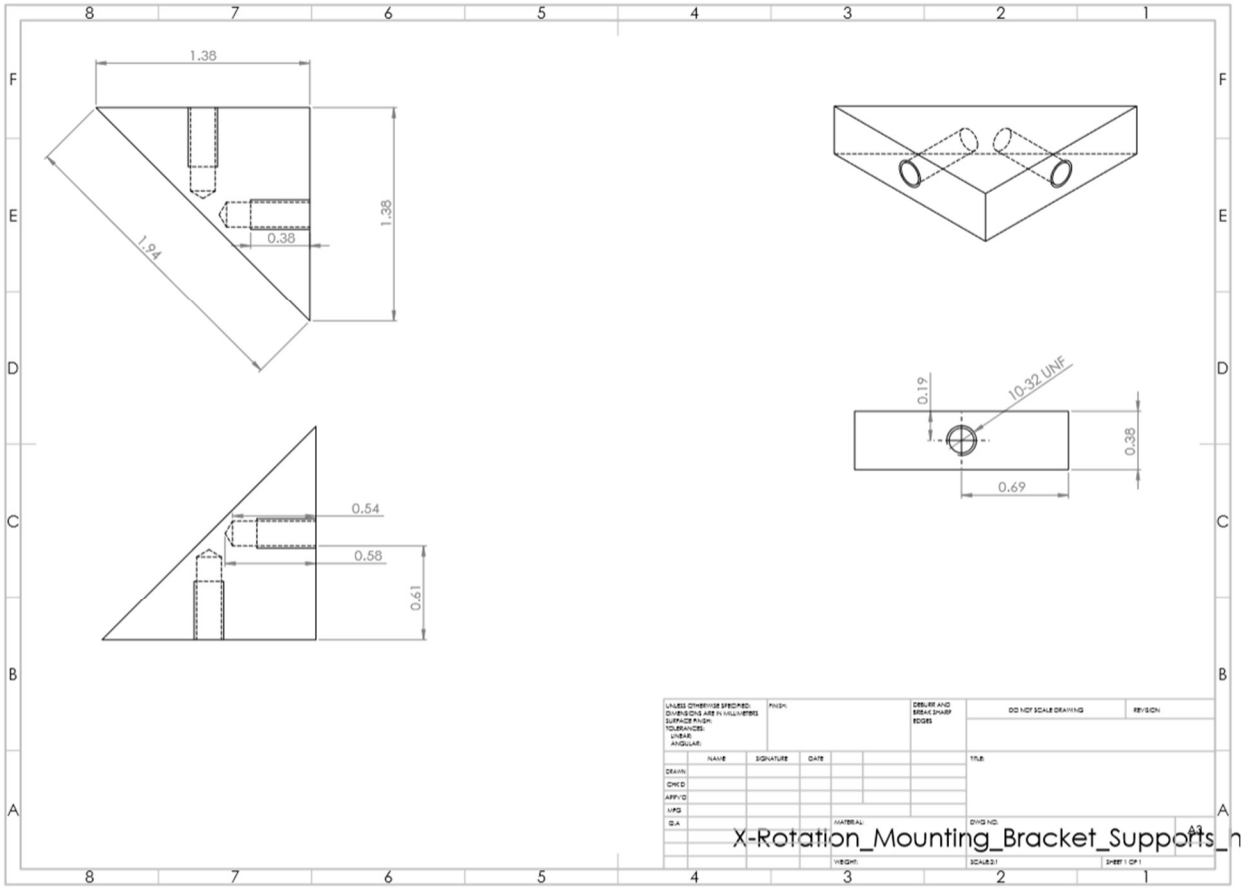
References

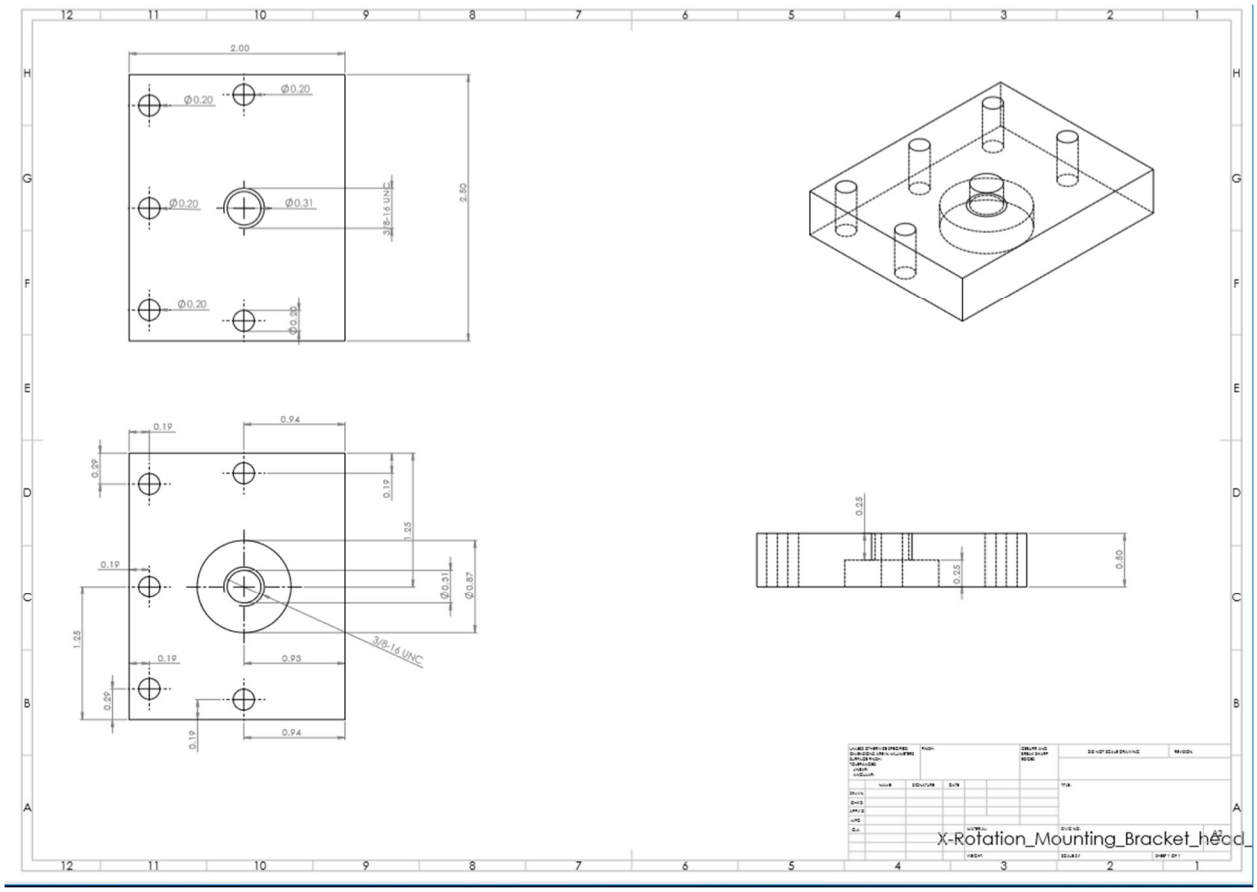
[1] Team MAM_STIPPL, “Project Performance Specifications” (senior design project performance specifications, University of North Carolina at Charlotte, 2017), 2-3, original
[2] Team MAM_STIPPL, “Statement of Work” (senior design statement of work, University of North Carolina at Charlotte, 2017), 1-6, original
[3] Team MAM_STIPPL, “Project Plan” (senior design project plan, University of North Carolina at Charlotte, 2017), original

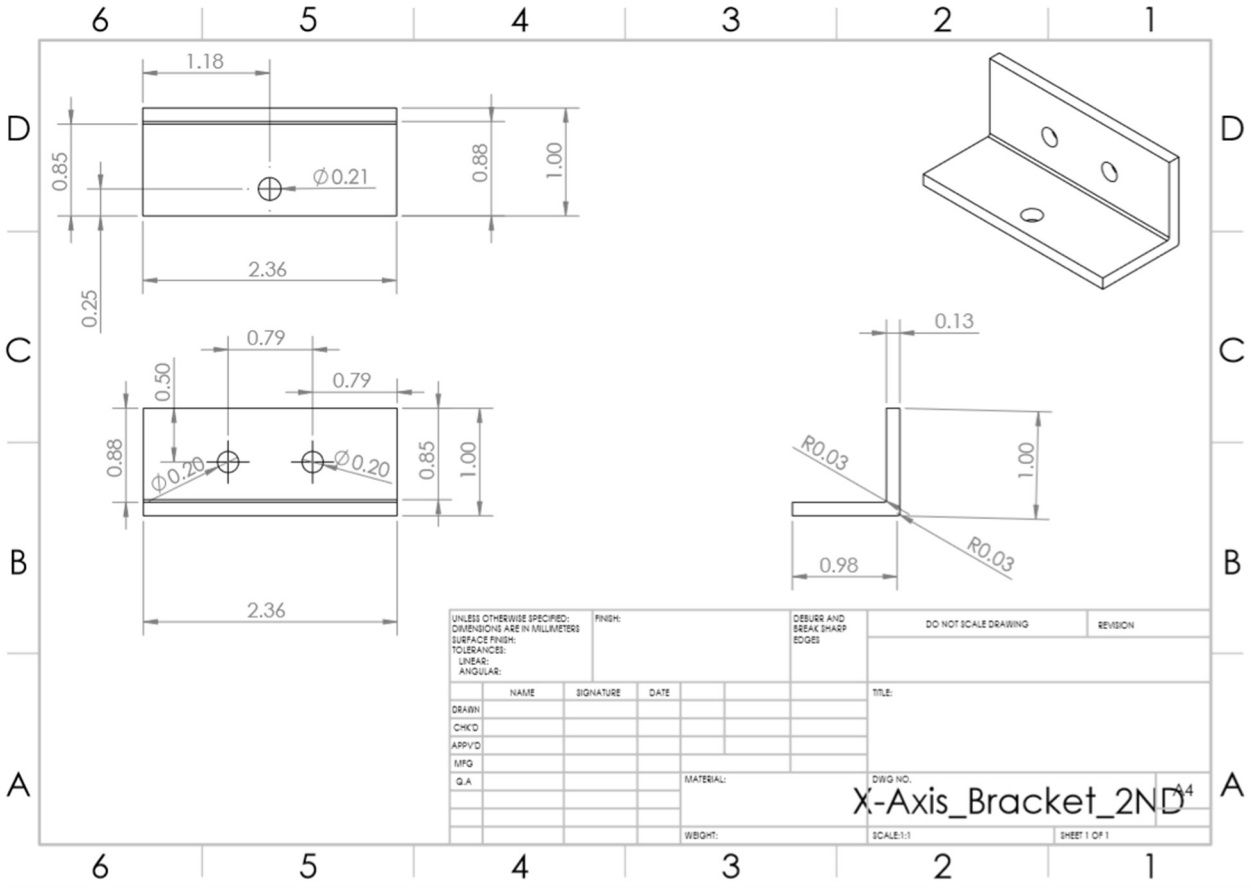
Appendix A: Drawings

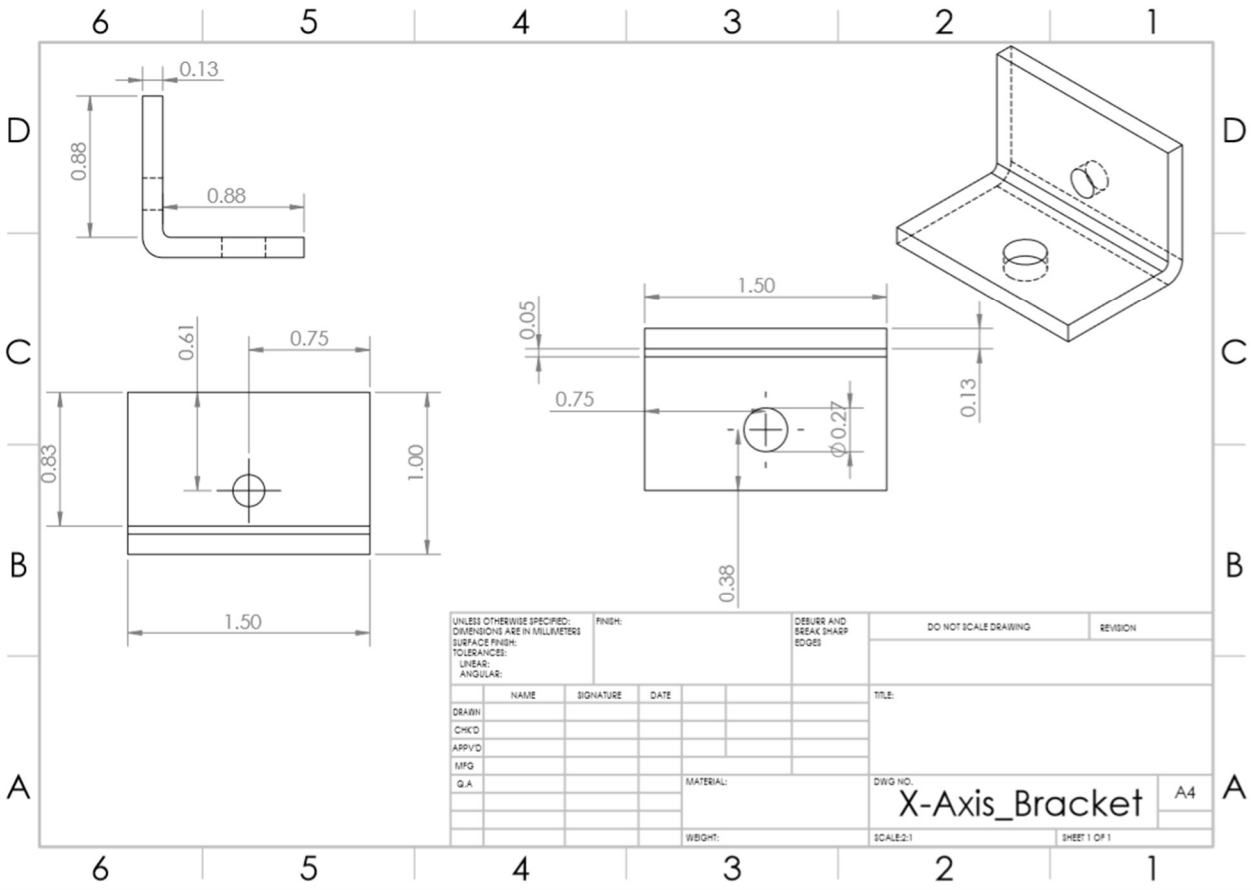


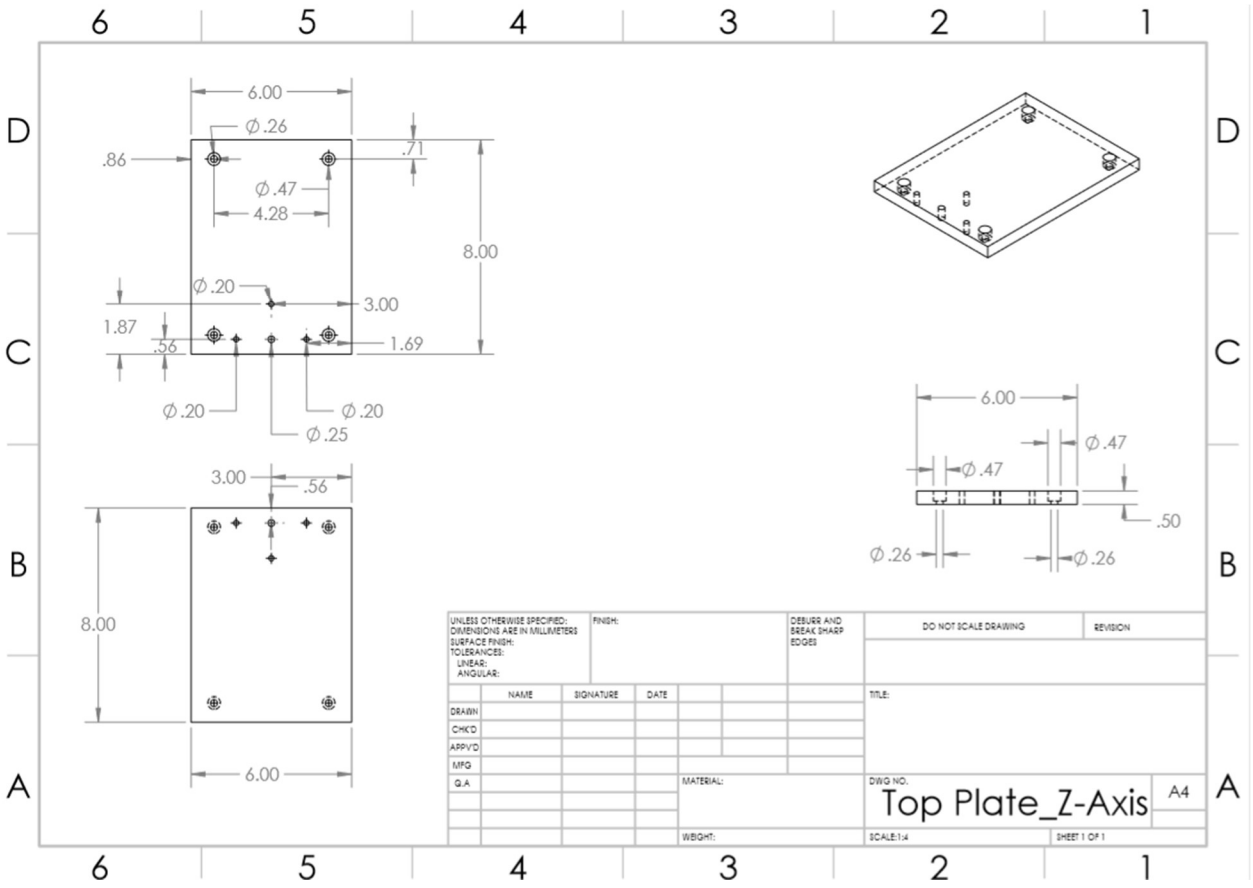

.48
.48
Ø.20
Ø.20
.48
.48
3.76
Ø1.26
Ø.94
2.62
2.01
Ø.13
1.42
Ø.25
2.62
.81
1.74
8-32 UNC
Ø.14
1.14
Ø.25
Ø.38
Ø.48
Ø.20
.48
.48
.32
2.62
7.53
Ø.94
.38
.25
1.00
.13
.38
.10
.13
.10
Ø1.26
#8-32 Tapped Hole
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION:
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APP'D
MFG
Q.A
TITLE:
MATERIAL:
DWG NO.
Z-Axis_Bottom_Plate
A3
WEIGHT:
SCALE:1:2
SHEET 1 OF 1

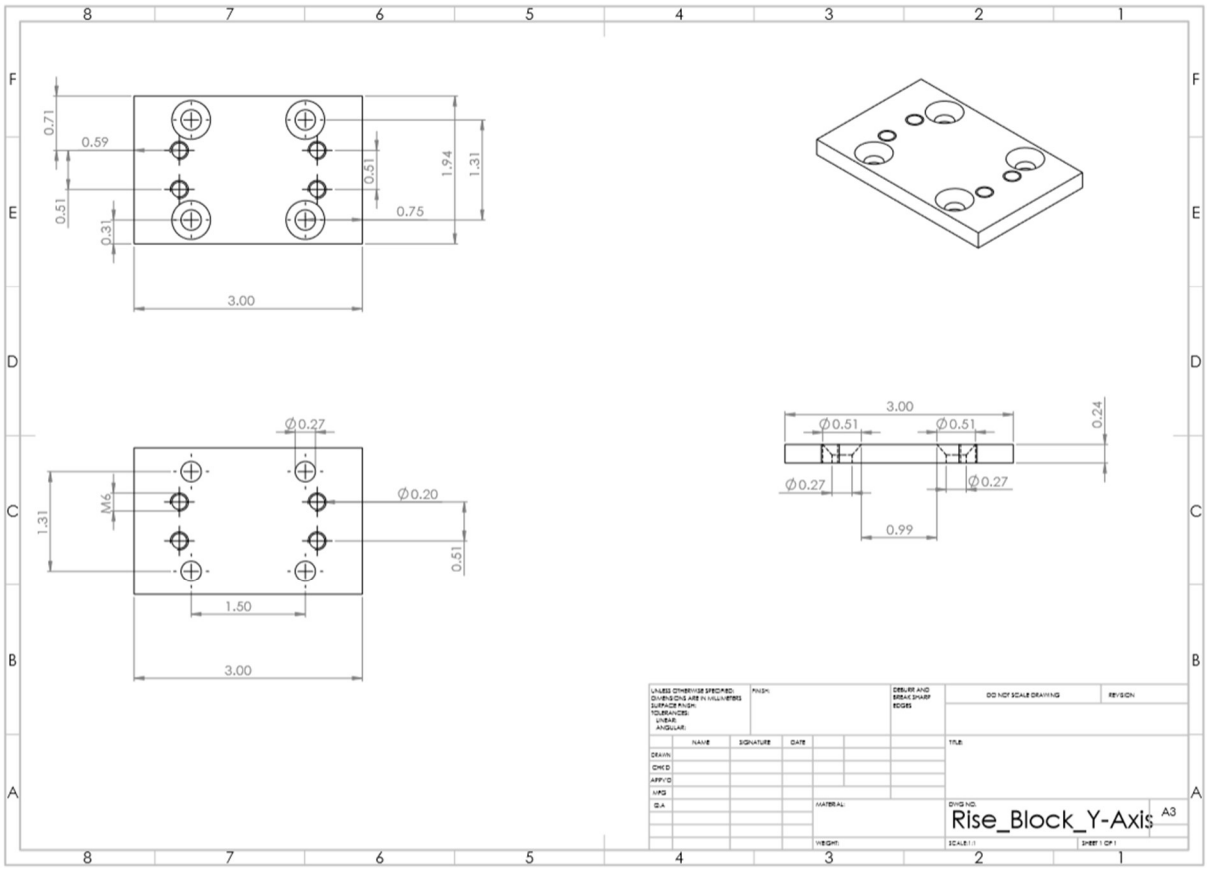

1.38
1.38
1.94
0.38
0.54
0.58
0.61
0.19
10-32 UNF
0.38
0.69
X-Rotation_Mounting_Bracket_Supports_n

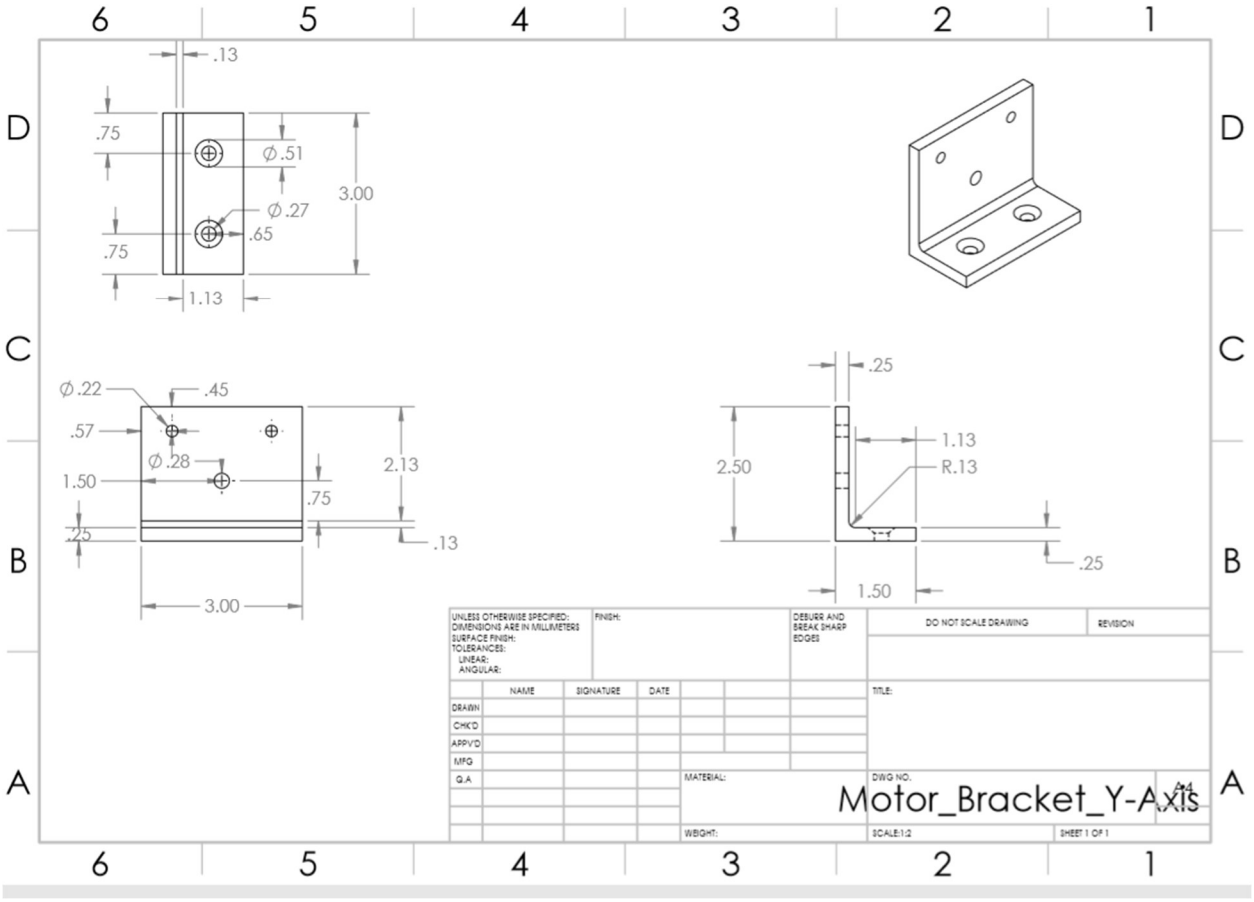

2.00
Ø0.20
Ø0.20
Ø0.20
Ø0.31
3/8-16 UNC
2.50
Ø0.20
Ø0.20
0.19
0.94
0.19
0.19
1.25
0.19
Ø0.31
Ø0.87
3/8-16 UNC
1.25
0.19
0.93
0.19
0.94
0.25
0.25
0.60
X-Rotation_Mounting_Bracket_head_

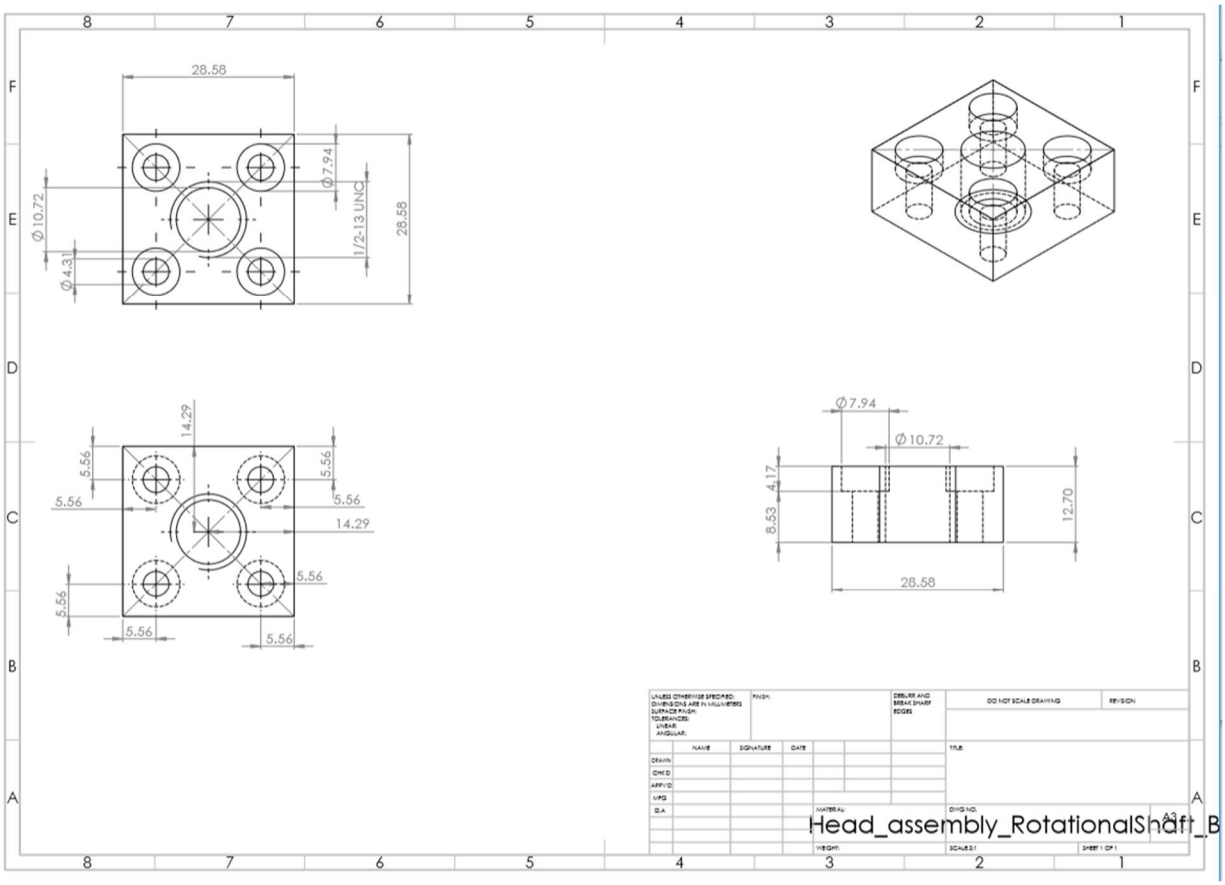

1.18
0.85
Ø0.21
0.88
1.00
2.36
0.25
0.79
0.79
0.50
0.88
Ø0.20
Ø0.20
0.85
1.00
2.36
0.13
R0.03
1.00
R0.03
0.98
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
MATERIAL:
DWG NO.
A4
X-Axis_Bracket_2ND
WEIGHT:
SCALE:1:1
SHEET 1 OF 1

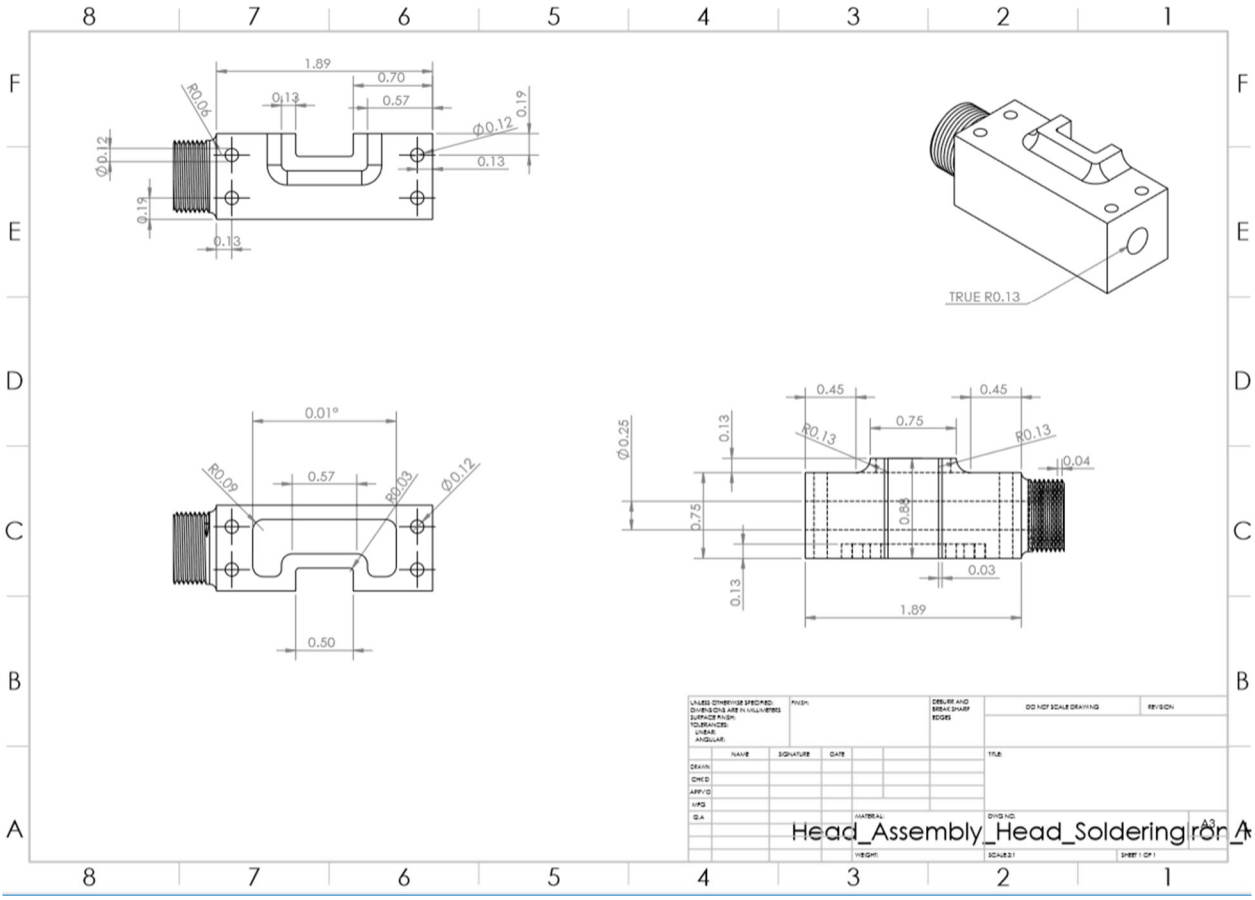

0.13
0.88
0.88
0.61
0.75
0.83
1.00
1.50
0.05
1.50
0.75
Ø0.27
0.13
0.38
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
MATERIAL:
DWG NO.
X-Axis_Bracket
A4
WEIGHT:
SCALE:2:1
SHEET 1 OF 1

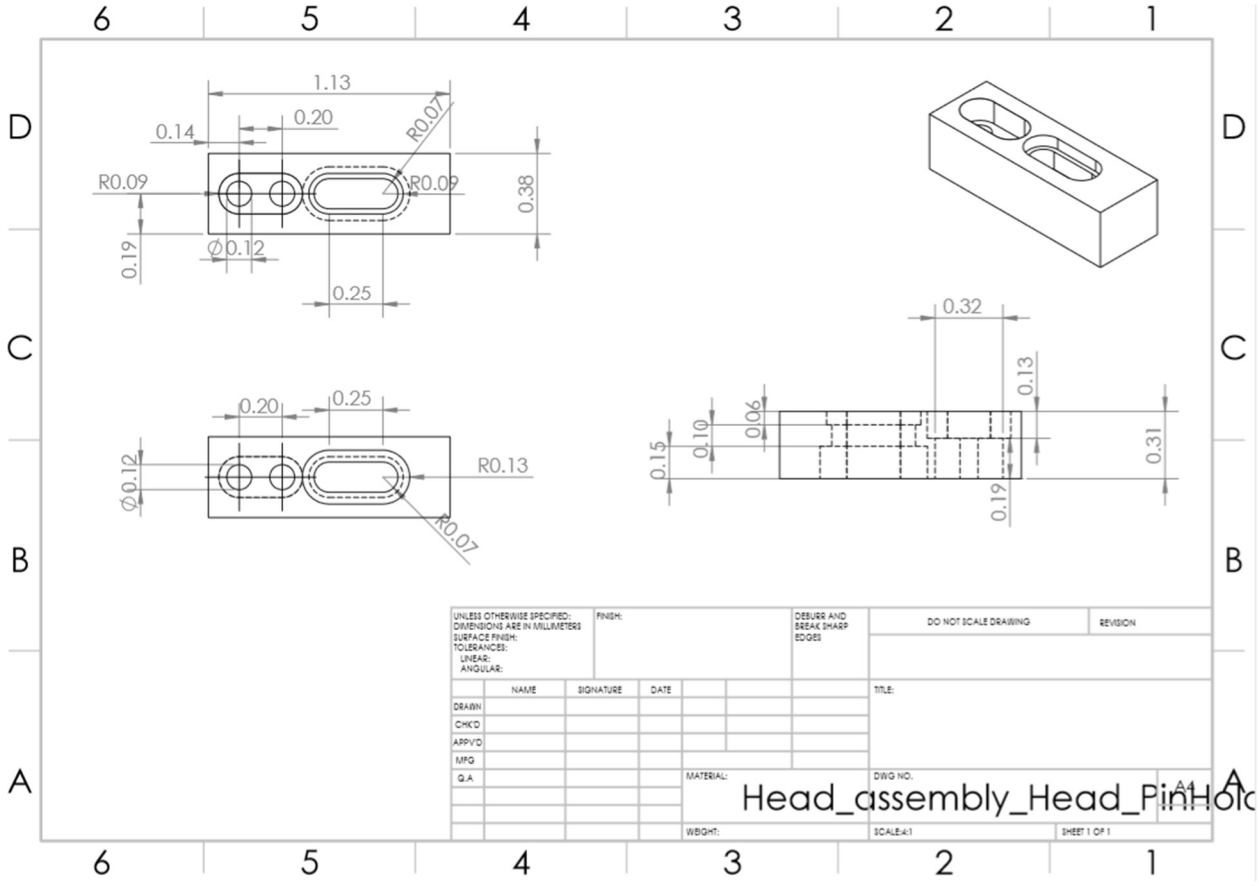

6.00
Ø.26
.86
.71
Ø.47
4.28
8.00
Ø.20
3.00
1.87
.56
1.69
Ø.20
Ø.20
Ø.25
3.00
.56
8.00
6.00
6.00
Ø.47
Ø.47
.50
Ø.26
Ø.26
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
MATERIAL:
DWG NO.
Top Plate_Z-Axis
A4
WEIGHT:
SCALE:1:4
SHEET 1 OF 1

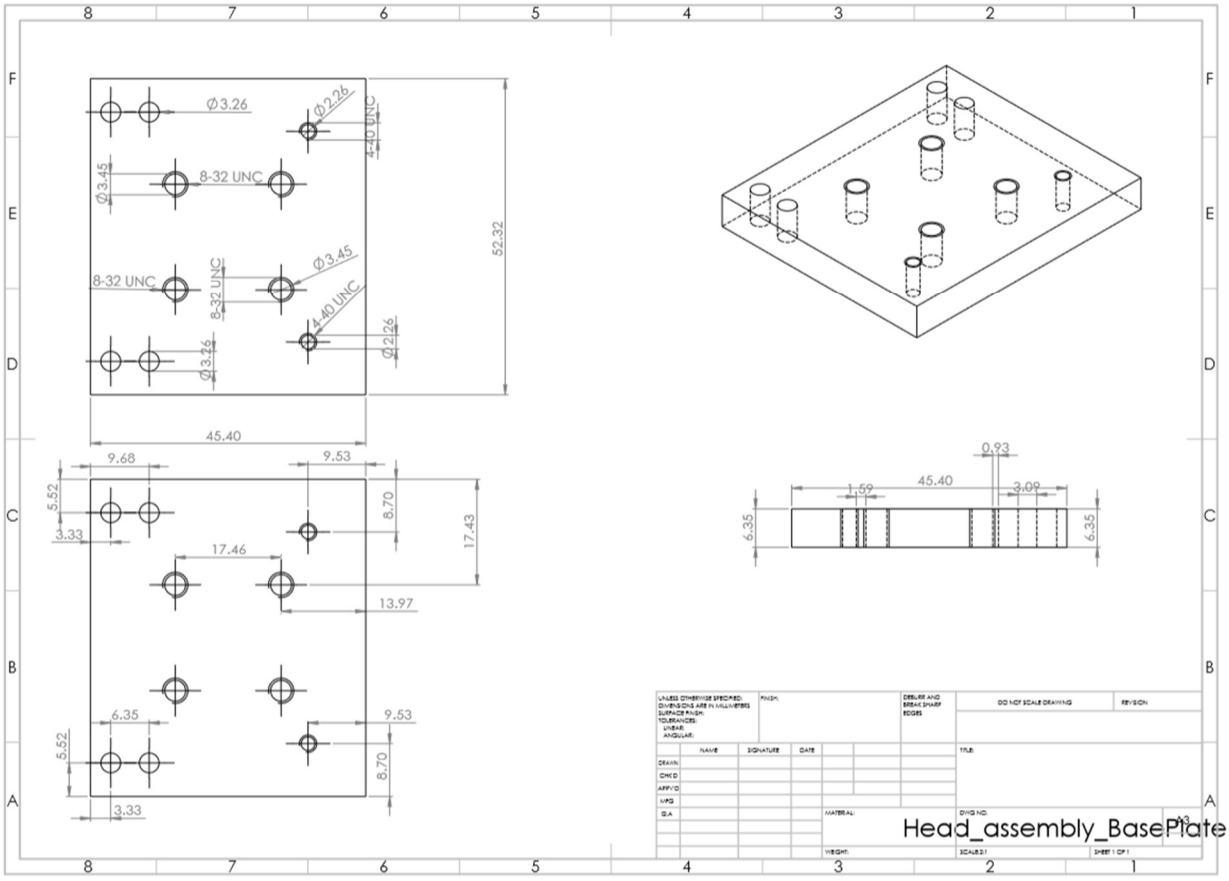

0.71
0.59
0.51
0.31
0.51
1.94
1.31
0.75
3.00
Ø0.27
M6
1.31
Ø0.20
0.51
1.50
3.00
3.00
Ø0.51
Ø0.51
0.24
Ø0.27
Ø0.27
0.99
Rise_Block_Y-Axis
A3

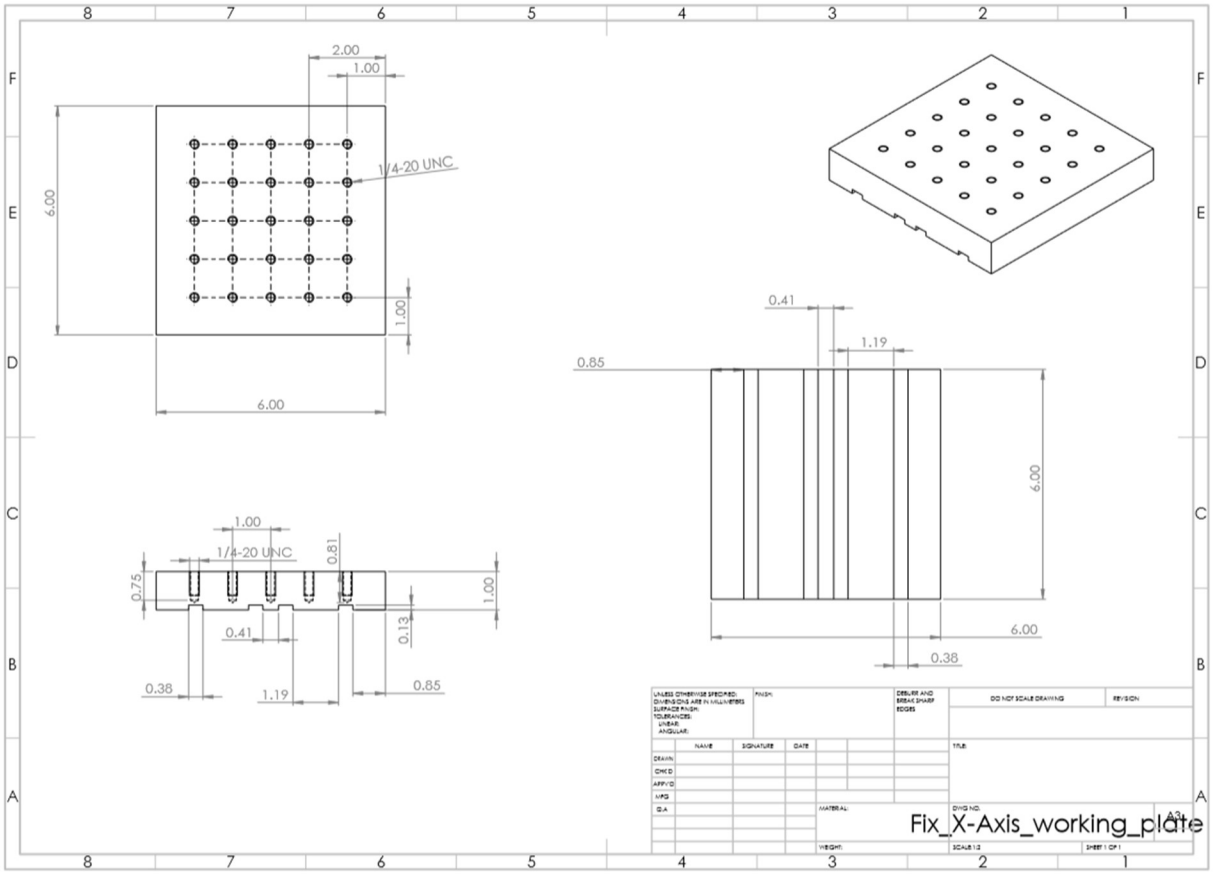

.13
.75
Ø.51
3.00
Ø.27
.65
.75
1.13
Ø.22
.45
.57
Ø.28
1.50
2.13
.75
.25
.13
3.00
.25
2.50
1.13
R.13
.25
1.50
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
MATERIAL:
DWG NO.
A4
Motor_Bracket_Y-Axis
WEIGHT:
SCALE:1:2
SHEET 1 OF 1

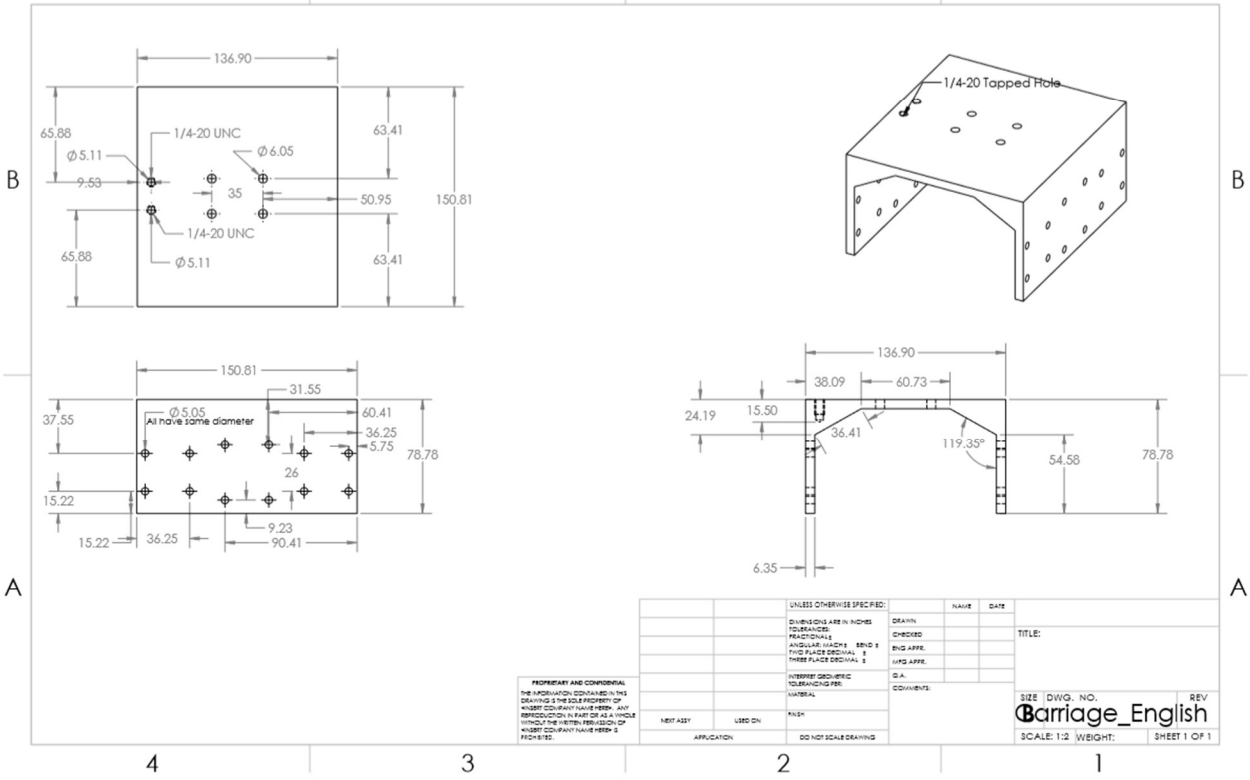

28.58
Ø7.94
Ø10.72
1/2-13 UNC
28.58
Ø4.31
14.29
5.56
5.56
5.56
5.56
14.29
5.56
5.56
5.56
Ø7.94
Ø10.72
4.17
8.53
12.70
28.58
Head_assembly_RotationalShaft_B

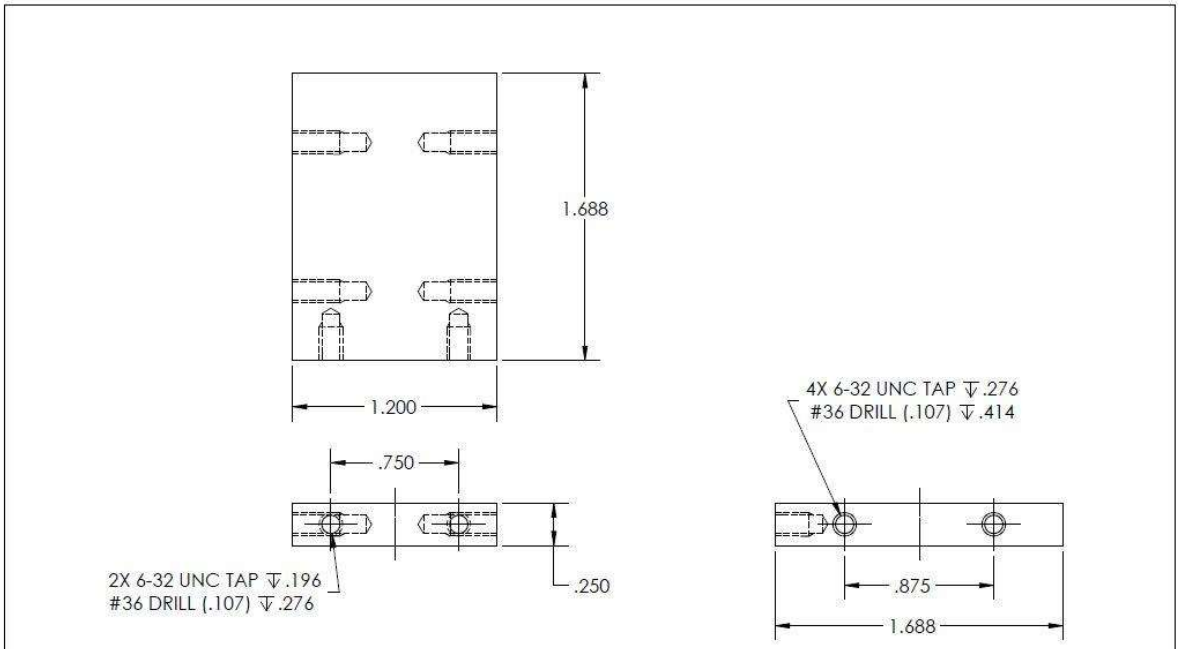

1.89
0.70
0.57
R0.06
0.13
0.19
Ø0.12
0.13
Ø0.12
Ø0.19
0.13
TRUE R0.13
0.01°
R0.09
0.57
R0.03
Ø0.12
0.50
0.45
0.45
0.13
R0.13
0.75
R0.13
Ø0.25
0.04
0.75
0.85
0.03
0.13
1.89
DO NOT SCALE DRAWING
Head_Assembly_Head_Soldering Iron_A


1.13
0.20
0.14
R0.07
R0.09
R0.09
0.38
0.19
Ø0.12
0.25
0.32
0.13
0.06
0.10
0.15
0.31
0.19
0.20
0.25
Ø0.12
R0.13
R0.07
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
MATERIAL:
DWG NO.
A1
Head_assembly_Head_PinHold
WEIGHT:
SCALE:4:1
SHEET 1 OF 1


Ø3.26
Ø2.26
4-40 UNC
Ø3.45
8-32 UNC
8-32 UNC
8-32 UNC
Ø3.45
4-40 UNC
Ø3.26
Ø2.26
52.32
45.40
9.68
9.53
5.52
3.33
8.70
17.43
17.46
13.97
6.35
9.53
5.52
8.70
3.33
0.93
45.40
1.59
3.09
6.35
6.35
DO NOT SCALE DRAWING
Head_assembly_BasePlate


2.00
1.00
1/4-20 UNC
6.00
1.00
6.00
0.41
1.19
0.85
6.00
0.38
1.00
1/4-20 UNC
0.81
0.75
1.00
0.41
0.13
0.38
1.19
0.85
Fix_X-Axis_working_plate


136.90
65.88
1/4-20 UNC
Ø5.11
Ø6.05
9.53
35
50.95
63.41
150.81
1/4-20 UNC
65.88
Ø5.11
63.41
1/4-20 Tapped Hole
150.81
31.55
37.55
Ø5.05
All have same diameter
60.41
36.25
5.75
78.78
26
15.22
9.23
15.22
36.25
90.41
136.90
38.09
60.73
24.19
15.50
36.41
119.35°
54.58
78.78
6.35
Barriage_English



The WILLIAM STATES LEE COLLEGE of ENGINEERING
UNC CHARLOTTE

The University of North Carolina at Charlotte

HEADASSEMBLY_MOTORSUPPORT_SIDE

Scale: 3:2	Sheet 1 of 1	Rev:--	A

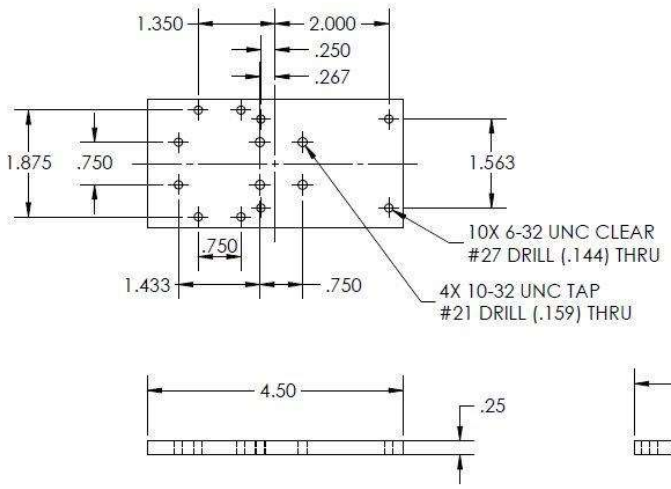
Tolerances UOS	
2-place decimal	±.01
3-place decimal	±.005
Angular--bent	±2°
Angular--machined	±.5°
Do not scale drawing	

	Initials	Date
Drawn by:	CAW	12/06/2017
Checked by:		mm/dd/yyyy
Approved by:		mm/dd/yyyy

Proprietary and Confidential

The information contained in this drawing is the property of the University of North Carolina at Charlotte (the "University"). All rights are reserved by the University. No right to reproduce this document in part or in whole is granted except by separate written agreement with the University.

5 4 3 2 1



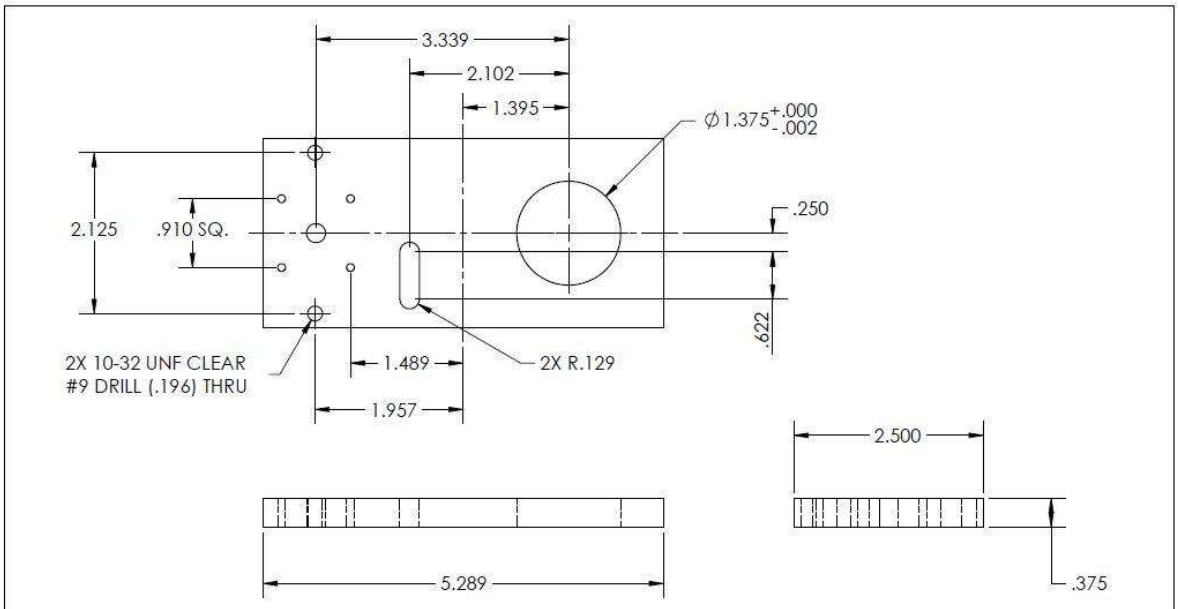
The WILLIAM STATES LEE COLLEGE of ENGINEERING
UNC CHARLOTTE

The University of North Carolina at Charlotte

HEADASSEMBLY_MOUNTINGBRACKET

Proprietary and Confidential	Tolerances UOS			Initials	Date				
The information contained in this drawing is the property of the University of North Carolina at Charlotte (the "University"). All rights are reserved by the University. No right to reproduce this document in part or in whole is granted except by separate written agreement with the University.	2-place decimal	±.01	Drawn by:	CAW	12/06/2017				
	3-place decimal	±.005	Checked by:		mm/dd/yyyy				
	Angular--bent	±2°	Approved by:		mm/dd/yyyy		Scale: 1:2	Sheet 1 of 1	Rev:-- A
	Angular--machined	±.5°							
	Do not scale drawing								

5 4 3 2 1



The WILLIAM STATES LEE COLLEGE of ENGINEERING
UNC CHARLOTTE

The University of North Carolina at Charlotte

X-ROTATIONAL_BEARING_BRACKET

			Initials	Date
Drawn by:	CAW	12/07/2017		
Checked by:		mm/dd/yyyy		
Approved by:		mm/dd/yyyy		

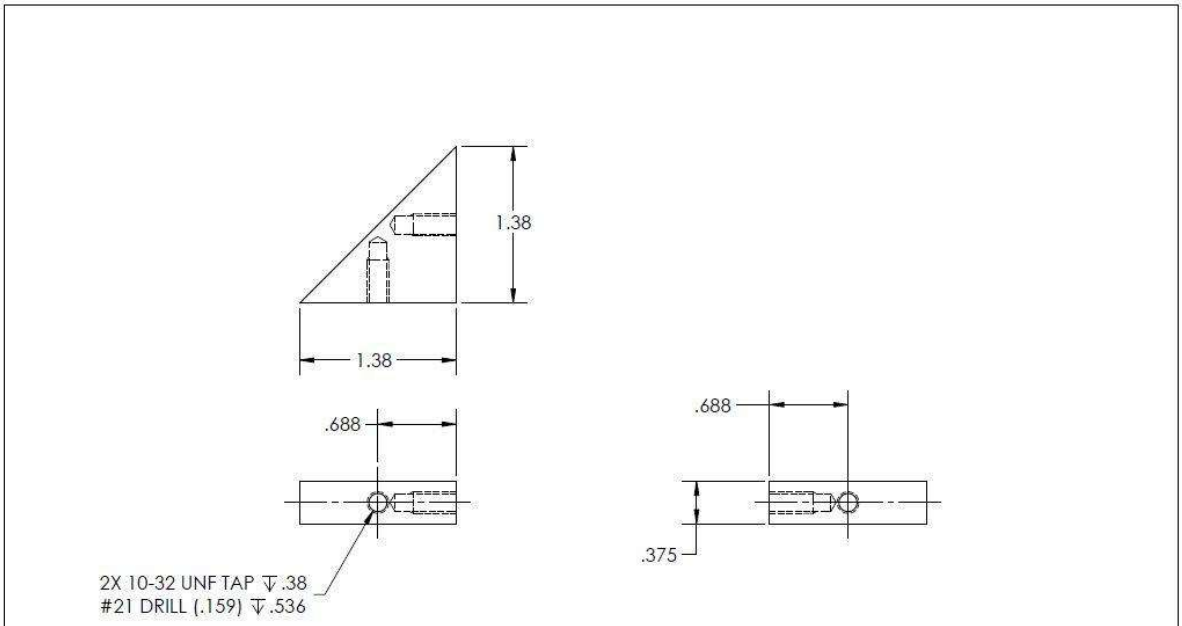
Scale: 2:3 | Sheet 1 of 1 | Rev:-- | A

Tolerances UOS	
2-place decimal	±.01
3-place decimal	±.005
Angular--bent	±2°
Angular--machined	±.5°
Do not scale drawing	

Proprietary and Confidential

The information contained in this drawing is the property of the University of North Carolina at Charlotte (the "University"). All rights are reserved by the University. No right to reproduce this document in part or in whole is granted except by separate written agreement with the University.

5 4 3 2 1



The WILLIAM STATES LEE COLLEGE of ENGINEERING
UNC CHARLOTTE

The University of North Carolina at Charlotte

X-ROTATION_MOUNTING_BRACKET_SUPPORT

Proprietary and Confidential	Tolerances UOS	
The information contained in this drawing is the property of the University of North Carolina at Charlotte (the "University"). All rights are reserved by the University. No right to reproduce this document in part or in whole is granted except by separate written agreement with the University.	2-place decimal	±.01
	3-place decimal	±.005
	Angular--bent	±2°
	Angular--machined	±.5°
	Do not scale drawing	

	Initials	Date
Drawn by:	CAW	12/08/2017
Checked by:		mm/dd/yyyy
Approved by:		mm/dd/yyyy

Scale: 1:1	Sheet 1 of 1	Rev:--	A

5 4 3 2 1

Appendix B: LinuxCNC Configuration Code

5AXIS_TEST_V4.ini

```
#-----------------------------------------------------------------------
#
#Author: Matthew Osborne
#Date: April 26,2018
#Version: LinuxCNC 2.5 Dev
#License: GPL Version 2
#
#-----------------------------------------------------------------------



########################################################################
# General Section
########################################################################

# General section ------------------------------------------------------
[EMC]

# Version of this INI file
VERSION =              $Revision: 1.0$

# Name of machine, for use with display, etc.
MACHINE =              5AXIS_TEST_V4

# Debug level, 0 means no messages. See src/emc/nml_int/emcglb.h for others
DEBUG =                0



########################################################################
# Sections for Display Options
########################################################################
[DISPLAY]

# Name of display program, e.g., xemc
DISPLAY = axis

# The coordinate system (RELATIVE or MACHINE) to show when the user interface starts. The
RELATIVE coordinate system reflects the G92 and G5x coordinate offsets currently in effect.
POSITION_OFFSET =          RELATIVE

# The coordinate value (COMMANDED or ACTUAL) to show when the user interface starts.
The COMMANDED position is the ideal position requested by LinuxCNC. The ACTUAL
position is the feedback position of the motors.
```

```
POSITION_FEEDBACK =    ACTUAL

# The maximum feed override the user may select. 1.2 means 120% of the programmed feed
rate.
MAX_FEED_OVERRIDE = 2.000000

# The maximum spindle override the user may select. 1.0 means 100% of the programmed
spindle speed.
MAX_SPINDLE_OVERRIDE =       1.000000

# The minimum spindle override the user may select. 0.5 means 50% of the programmed spindle
speed. (This is useful as it's dangerous to run a program with a too low spindle speed).
MIN_SPINDLE_OVERRIDE =       0.500000

# The image shown on the splash screen.
INTRO_GRAPHIC =          linuxcnc.gif

# The maximum time to show the splash screen, in seconds.
INTRO_TIME =             5

# The default location for g-code files and the location for user-defined M-codes. This location is
searched for the file name before the subroutine path and user M path if specified in the
[RS274NGC] section.
PROGRAM_PREFIX =         /home/mammoth/linuxcnc/nc_files

# Defines the increments available for incremental jogs. The INCREMENTS can be used to
override the default. The values can be decimal numbers (e.g., 0.1000) or fractional numbers
(e.g., 1/16), optionally followed by a unit (cm, mm, um, inch, in or mil). If a unit is not specified
the machine unit is assumed. Metric and imperial distances may be mixed: INCREMENTS = 1
inch, 1 mil, 1 cm, 1 mm, 1 um is a valid entry.
INCREMENTS =             .5in .1in .05in .01in .005in .001in .0005in .0001in

# The default velocity for linear jogs, in , machine units per second.
DEFAULT_LINEAR_VELOCITY = 0.250000

# The maximum velocity for linear jogs, in machine units per second.
MAX_LINEAR_VELOCITY =        2.000000

# The approximate lowest value the jog slider.
MIN_LINEAR_VELOCITY =        0.010000

# The default velocity for angular jogs, in machine units per second.
DEFAULT_ANGULAR_VELOCITY = 1.666667

# The maximum velocity for angular jogs, in machine units per second.
MAX_ANGULAR_VELOCITY = 16.666667
```

```
# The approximate lowest value the angular jog slider.
MIN_ANGULAR_VELOCITY = 0.010000

# The editor to use when selecting File > Edit to edit the G code from the AXIS menu. This must
be configured for this menu item to work. Another valid entry is gnome-terminal -e vim.
EDITOR = gedit

# Controls the preview and backplot of rotary motion. This item consists of a sequence of axis
letters, optionally preceded by a "-" sign. Only axes defined in [TRAJ]AXES should be used.
This sequence specifies the order in which the effect of each axis is applied, with a "-" inverting
the sense of the rotation. The proper GEOMETRY string depends on the machine configuration
and the kinematics used to control it. The example string GEOMETRY=XYZBCUVW is for a 5-
axis machine where kinematics causes UVW to move in the coordinate system of the tool and
XYZ to move in the coordinate system of the material. The order of the letters is important,
because it expresses the order in which the different transformations are applied. For example
rotating around C then B is different than rotating around B then C. Geometry has no effect
without a rotary axis.
GEOMETRY = XYZBC



############################################################################
# Filter Section
############################################################################
[FILTER]

PROGRAM_EXTENSION = .png,.gif,.jpg Greyscale Depth Image
PROGRAM_EXTENSION = .py Python Script
png = image-to-gcode
gif = image-to-gcode
jpg = image-to-gcode
py = python



############################################################################
# Task Controller Section
############################################################################
[TASK]

# Name of task controller program, e.g., milltask
TASK =                  milltask

# Cycle time, in seconds, that task controller will sleep between polls
CYCLE_TIME =            0.010
```

```
# Part Program Interpreter Section
[RS274NGC]

# The file located in the same directory as the ini file which contains the parameters used by the
interpreter (saved between runs)
PARAMETER_FILE =        linuxcnc.var


# Motion Control Section
[EMCMOT]

# Motion controller name
EMCMOT =            motmod

#- Timeout for comm to emcmot, in seconds
COMM_TIMEOUT =          1.0

#- Interval between tries to emcmot, in seconds
COMM_WAIT =             0.010

#- Servo task period, in nanoseconds
SERVO_PERIOD =          1000000


# Hardware Abstraction Layer section
[HAL]

# Adds the HAL user interface pins
HALUI = halui

# Execute the file example.hal at start up. If HALFILE is specified multiple times, the files are
executed in the order they appear in the ini file. Almost all configurations will have at least one
HALFILE, and stepper systems typically have two such files, one which specifies the generic
stepper configuration (core_stepper.hal) and one which specifies the machine pin out
(xxx_pinout.hal)
```

```
HALFILE = 5AXIS_TEST_V4.hal
HALFILE = custom.hal

# (Only with the TOUCHY and AXIS GUI) Execute example2.hal after the GUI has created its
HAL pins
POSTGUI_HALFILE = postgui_call_list.hal

# Execute the file shutdown.hal when LinuxCNC is exiting. Depending on the hardware drivers
used, this may make it possible to set outputs to defined values when LinuxCNC is exited
normally. However, because there is no guarantee this file will be executed (for instance, in the
case of a computer crash) it is not a replacement for a proper physical e-stop chain or other
protections against software failure.
SHUTDOWN = shutdown.hal



#############################################################################
# Trajectory Planner Section
#############################################################################
[TRAJ]

# One more than the number of the highest joint number in the system. For an XYZ machine, the
joints are numbered 0, 1 and 2; in this case AXES should be 3. For an XYUV machine using
trivial kinematics, the V joint is numbered 7 and therefore AXES should be 8. For a machine
with nontrivial kinematics (e.g., scarakins) this will generally be the number of controlled joints.
AXES = 6

# The names of the axes being controlled. Only X, Y, Z, A, B, C, U, V, W are valid. Only axes
named in COORDINATES are accepted in g-code. This has no effect on the mapping from G-
code axis names (X- Y- Z-) to joint numbers—for trivial kinematics, X is always joint 0, A is
always joint 3, and U is always joint 6, and so on. It is permitted to write an axis name twice
(e.g., X Y Y Z for a gantry machine) but this has no effect.
COORDINATES = X Y Z B C

# The maximum angular velocity for any axis or coordinated move, in machine units per second.
MAX_ANGULAR_VELOCITY = 16.67

# The initial rate for jogs of angular axes, in machine units per second. The value shown in Axis
equals machine units per minute.
DEFAULT_ANGULAR_VELOCITY = 1.67

# Specifies the machine units for linear axes. Possible choices are (in, inch, imperial, metric,
mm). This does not affect the linear units in NC code (the G20 and G21 words do this).
LINEAR_UNITS = inch
```

# Specifies the machine units for rotational axes. Possible choices are deg, degree (360 per circle), rad, radian (2pi per circle), grad, or gon (400 per circle). This does not affect the angular units of NC code. In RS274NGC, A-, B- and C- words are always expressed in degrees.
ANGULAR_UNITS = degree

CYCLE_TIME = 0.010

# The initial rate for jogs of linear axes, in machine units per second. The value shown in Axis equals machine units per minute.
DEFAULT_VELOCITY = 0.17

# The maximum linear velocity for any axis or coordinated move, in machine units per second.
MAX_LINEAR_VELOCITY = 1.67

```
#****************************************************************************
# Axes Section
#****************************************************************************
```

TYPE - The type of axes, either LINEAR or ANGULAR

HOME - The position that the joint will go to upon completion of the homing sequence.

HOME_OFFSET - The axis position of the home switch or index pulse, in machine units. When the home point is found during the homing process, this is the position that is assigned to that point. When sharing home and limit switches and using a home sequence that will leave the home/limit switch in the toggled state the home offset can be used define the home switch position to be other than 0 if your HOME position is desired to be 0.

HOME_SEARCH_VEL - nitial homing velocity in machine units per second. Sign denotes direction of travel. A value of zero means assume that the current location is the home position for the machine. If your machine has no home switches you will want to leave this value at zero.

HOME_LATCH_VEL - Homing velocity in machine units per second to the home switch latch position. Sign denotes direction of travel.

HOME_USE_INDEX - If the encoder used for this axis has an index pulse, and the motion card has provision for this signal you may set it to yes. When it is yes, it will affect the kind of home pattern used. Currently, you can’t home to index with steppers unless you’re using stepgen in velocity mode and PID.

HOME_IGNORE_LIMITS - When you use the limit switch as a home switch and the limit switch this should be set to YES. When set to YES the limit switch for this axis is ignored when homing. You must configure your homing so that at the end of your home move the home/limit switch is not in the toggled state you will get a limit switch error after the home move.

VOLATILE_HOME - When enabled (set to 1) this joint will be unhomed if the Machine Power is off or if E-Stop is on. This is useful if your machine has home switches and does not have position feedback such as a step and direction driven machine.

FERROR - FERROR is the maximum allowable following error, in machine units. If the difference between commanded and sensed position exceeds this amount, the controller disables servo calculations, sets all the outputs to 0.0, and disables the amplifiers. If MIN_FERROR is present in the .ini file, velocity-proportional following errors are used. Here, the maximum allowable following error is proportional to the speed, with FERROR applying to the rapid rate set by [TRAJ]MAX_VELOCITY, and proportionally smaller following errors for slower speeds. The maximum allowable following error will always be greater than MIN_FERROR. This prevents small following errors for stationary axes from inadvertently aborting motion. Small following errors will always be present due to vibration, etc. The following polarity values determine how inputs are interpreted and how outputs are applied. They can usually be set via trial-and-error since there are only two possibilities. The LinuxCNC Servo Axis Calibration utility program (in the AXIS interface menu Machine/Calibration and in TkLinuxCNC it is under Setting/Calibration) can be used to set these and more interactively and verify their results so that the proper values can be put in the INI file with a minimum of trouble.

MIN_FERROR - This is the value in machine units by which the axis is permitted to deviate from commanded position at very low speeds. If MIN_FERROR is smaller than FERROR, the two produce a ramp of error trip points. You could think of this as a graph where one dimension is speed and the other is permitted following error. As speed increases the amount of following error also increases toward the FERROR value.

MAX_VELOCITY - Maximum velocity for this axis in machine units per second.

MAX_ACCELERATION - Maximum acceleration for this axis in machine units per second squared.

DIRSETUP - Stepper motor driver direction setup timing in nanoseconds.

DIRHOLD - Stepper motor driver direction hold timing in nanoseconds.

STEPLEN - Stepper motor driver step time in nanoseconds.

STEPSPACE - Stepper motor driver step space time in nanoseconds.

STEP_SCALE - The number of pulses that corresponds to a move of one machine unit as set in the [TRAJ] section. For stepper systems, this is the number of step pulses issued per machine unit. For a linear axis one machine unit will be equal to the setting of LINEAR_UNITS. For an angular axis one unit is equal to the setting in ANGULAR_UNITS. For servo systems, this is the number of feedback pulses per machine unit. A second number, if specified, is ignored.

MIN_LIMIT - The minimum limit (soft limit) for axis motion, in machine units. When this limit is exceeded, the controller aborts axis motion.

MAX_LIMIT - The maximum limit (soft limit) for axis motion, in machine units. When this limit is exceeded, the controller aborts axis motion.

```
#---------------------------------------------------------------------------
# Axis x
#---------------------------------------------------------------------------
[AXIS_0]

TYPE =                      LINEAR
HOME =                       0.000
HOME_OFFSET =                5.25
HOME_SEARCH_VEL =             0.5
HOME_LATCH_VEL =             0.08
HOME_USE_INDEX =             NO
HOME_IGNORE_LIMITS =          YES
VOLATILE_HOME =                          1
FERROR =                    0.005
MIN_FERROR =                         0.0005
MAX_VELOCITY =                       0.5
MAX_ACCELERATION =                   2.0
DIRSETUP   =                         5000
DIRHOLD    =                         20000
STEPLEN    =                         7500
STEPSPACE  =                         7500
STEP_SCALE =                         1270.0
MIN_LIMIT =                          -10.0
MAX_LIMIT =                          10.0


#---------------------------------------------------------------------------
# Axis Y
#---------------------------------------------------------------------------
[AXIS_1]

TYPE =                      LINEAR
HOME =                       0.000
HOME_OFFSET =                11
HOME_SEARCH_VEL =             1.0
HOME_LATCH_VEL =             0.08
HOME_USE_INDEX =             NO
HOME_IGNORE_LIMITS =          YES
VOLATILE_HOME =                          1
FERROR =                    0.005
MIN_FERROR =                         0.0005
MAX_VELOCITY =                       1.667
```

```
MAX_ACCELERATION =              2.0
DIRSETUP   =                    5000
DIRHOLD    =                    20000
STEPLEN    =                    7500
STEPSPACE  =                    7500
STEP_SCALE =                    2539.9999
MIN_LIMIT =                     -10.0
MAX_LIMIT =                     10.0

#-----------------------------------------------------------------------
# Axis Z
#-----------------------------------------------------------------------
[AXIS_2]
TYPE =                      LINEAR
HOME =                       0.000
HOME_OFFSET =                4.6
HOME_SEARCH_VEL =             0.2
HOME_LATCH_VEL =             0.05
HOME_USE_INDEX =             NO
HOME_IGNORE_LIMITS =          YES
VOLATILE_HOME =                        1
FERROR =                    0.005
MIN_FERROR =                    0.0005
MAX_VELOCITY =                  0.666
MAX_ACCELERATION =              2.0
DIRSETUP   =                    5000
DIRHOLD    =                    20000
STEPLEN    =                    7500
STEPSPACE  =                    7500
STEP_SCALE =                    3200.0
MIN_LIMIT =                     -4
MAX_LIMIT =                     4

#-----------------------------------------------------------------------
# Axis B
#-----------------------------------------------------------------------
[AXIS_4]

TYPE =                      ANGULAR
HOME =                       0.000
HOME_OFFSET =                149
HOME_SEARCH_VEL =             5
HOME_LATCH_VEL =             1
HOME_USE_INDEX =             NO
HOME_IGNORE_LIMITS =          YES
VOLATILE_HOME =                        1
```

```
FERROR =                    0.005
MIN_FERROR =                0.0005
MAX_VELOCITY =              16.66666667
MAX_ACCELERATION =          40.0
DIRSETUP   =                2500
DIRHOLD    =                10000
STEPLEN    =                5000
STEPSPACE  =                5000
STEP_SCALE =                13.3333
MIN_LIMIT =                 0
MAX_LIMIT =                 150

#-----------------------------------------------------------------------
# Axis C
#-----------------------------------------------------------------------
[AXIS_5]

TYPE =                  ANGULAR
HOME =                   0.000
HOME_OFFSET =              65
HOME_SEARCH_VEL =          20.0
HOME_LATCH_VEL =           5.0
HOME_USE_INDEX =           NO
HOME_IGNORE_LIMITS =       YES
VOLATILE_HOME =                 1
FERROR =                    0.005
MIN_FERROR =                0.0005
MAX_VELOCITY =              16.66667
MAX_ACCELERATION =          40
DIRSETUP   =                2500
DIRHOLD    =                10000
STEPLEN    =                5000
STEPSPACE  =                5000
STEP_SCALE =                17.7778
MIN_LIMIT =                 -180
MAX_LIMIT =                 180
```

5AXIS_TEST_V4.hal

```
#--------------------------------------------------------------------------
#
#Author: Matthew Osborne
#Date: April 26,2018
#Version: LinuxCNC 2.5 Dev
#License: GPL Version 2
#
#--------------------------------------------------------------------------



#############################################################################################
# General Section
#############################################################################################
loadrt XYZBCkins_V1
loadrt [EMCMOT]EMCMOT servo_period_nsec=[EMCMOT]SERVO_PERIOD
num_joints=[TRAJ]AXES
loadrt probe_parport
loadrt hostmot2
loadrt hm2_pci config=" num_encoders=2 num_pwmgens=0 num_3pwmgens=0
num_stepgens=5 sserial_port_0=0xxxxxxx "
setp    hm2_5i25.0.watchdog.timeout_ns 10000000
loadrt abs names=abs.spindle
loadrt lowpass names=lowpass.spindle


addf hm2_5i25.0.read servo-thread
addf motion-command-handler servo-thread
addf motion-controller servo-thread
addf abs.spindle servo-thread
addf lowpass.spindle servo-thread
addf hm2_5i25.0.write        servo-thread
addf hm2_5i25.0.pet_watchdog  servo-thread



#############################################################################################
# External Input Signals
#############################################################################################
# --- MAX-HOME-X ---
net max-home-x    <=  hm2_5i25.0.7i76.0.0.input-00

# --- MIN-X ---
net min-x    <=  hm2_5i25.0.7i76.0.0.input-01
```

```
# --- MAX-HOME-Y ---
net max-home-y    <=  hm2_5i25.0.7i76.0.0.input-03

# --- MIN-Y ---
net min-y    <=  hm2_5i25.0.7i76.0.0.input-02

# --- MAX-HOME-Z ---
net max-home-z    <=  hm2_5i25.0.7i76.0.0.input-05

# --- MIN-Z ---
net min-z    <=  hm2_5i25.0.7i76.0.0.input-04

# --- MAX-HOME-B ---
net max-home-b    <=  hm2_5i25.0.7i76.0.0.input-07

# --- HOME-C ---
net home-c    <=  hm2_5i25.0.7i76.0.0.input-06



#********************************************************************************
# Axis X
#********************************************************************************
# Step Gen signals/setup
setp   hm2_5i25.0.stepgen.00.dirsetup        [AXIS_0]DIRSETUP
setp   hm2_5i25.0.stepgen.00.dirhold         [AXIS_0]DIRHOLD
setp   hm2_5i25.0.stepgen.00.steplen         [AXIS_0]STEPLEN
setp   hm2_5i25.0.stepgen.00.stepspace       [AXIS_0]STEPSPACE
setp   hm2_5i25.0.stepgen.00.position-scale  [AXIS_0]STEP_SCALE
setp   hm2_5i25.0.stepgen.00.step_type       0
setp   hm2_5i25.0.stepgen.00.control-type    0
setp   hm2_5i25.0.stepgen.00.maxaccel        2.5
setp   hm2_5i25.0.stepgen.00.maxvel          0.6

net x-pos-fb     axis.0.motor-pos-fb  <=  hm2_5i25.0.stepgen.00.position-fb
net x-pos-cmd    axis.0.motor-pos-cmd  =>  hm2_5i25.0.stepgen.00.position-cmd
net x-enable     axis.0.amp-enable-out =>  hm2_5i25.0.stepgen.00.enable

# ---setup home / limit switch signals---
net max-home-x    =>  axis.0.home-sw-in
net min-x          =>  axis.0.neg-lim-sw-in
net max-home-x    =>  axis.0.pos-lim-sw-in



#********************************************************************************
```

```
# Axis Y
#################################################################################
# Step Gen signals/setup
setp   hm2_5i25.0.stepgen.01.dirsetup       [AXIS_1]DIRSETUP
setp   hm2_5i25.0.stepgen.01.dirhold        [AXIS_1]DIRHOLD
setp   hm2_5i25.0.stepgen.01.steplen        [AXIS_1]STEPLEN
setp   hm2_5i25.0.stepgen.01.stepspace      [AXIS_1]STEPSPACE
setp   hm2_5i25.0.stepgen.01.position-scale  [AXIS_1]STEP_SCALE
setp   hm2_5i25.0.stepgen.01.step_type       0
setp   hm2_5i25.0.stepgen.01.control-type    0
setp   hm2_5i25.0.stepgen.01.maxaccel        2.5
setp   hm2_5i25.0.stepgen.01.maxvel          2.1

net y-pos-fb     axis.1.motor-pos-fb  <=  hm2_5i25.0.stepgen.01.position-fb
net y-pos-cmd    axis.1.motor-pos-cmd  =>  hm2_5i25.0.stepgen.01.position-cmd
net y-enable     axis.1.amp-enable-out =>  hm2_5i25.0.stepgen.01.enable

# ---setup home / limit switch signals---
net max-home-y     =>  axis.1.home-sw-in
net min-y          =>  axis.1.neg-lim-sw-in
net max-home-y     =>  axis.1.pos-lim-sw-in



#################################################################################
# Axis Z
#################################################################################
# Step Gen signals/setup
setp   hm2_5i25.0.stepgen.02.dirsetup       [AXIS_2]DIRSETUP
setp   hm2_5i25.0.stepgen.02.dirhold        [AXIS_2]DIRHOLD
setp   hm2_5i25.0.stepgen.02.steplen        [AXIS_2]STEPLEN
setp   hm2_5i25.0.stepgen.02.stepspace      [AXIS_2]STEPSPACE
setp   hm2_5i25.0.stepgen.02.position-scale  [AXIS_2]STEP_SCALE
setp   hm2_5i25.0.stepgen.02.step_type       0
setp   hm2_5i25.0.stepgen.02.control-type    0
setp   hm2_5i25.0.stepgen.02.maxaccel        2.5
setp   hm2_5i25.0.stepgen.02.maxvel          2.1

net z-pos-fb     axis.2.motor-pos-fb  <=  hm2_5i25.0.stepgen.02.position-fb
net z-pos-cmd    axis.2.motor-pos-cmd  =>  hm2_5i25.0.stepgen.02.position-cmd
net z-enable     axis.2.amp-enable-out =>  hm2_5i25.0.stepgen.02.enable

# ---setup home / limit switch signals---
net max-home-z     =>  axis.2.home-sw-in
net min-z          =>  axis.2.neg-lim-sw-in
net max-home-z     =>  axis.2.pos-lim-sw-in
```

```
#*******************************************************************************
# Axis B
#*******************************************************************************
# Step Gen signals/setup
setp   hm2_5i25.0.stepgen.04.dirsetup       [AXIS_4]DIRSETUP
setp   hm2_5i25.0.stepgen.04.dirhold        [AXIS_4]DIRHOLD
setp   hm2_5i25.0.stepgen.04.steplen        [AXIS_4]STEPLEN
setp   hm2_5i25.0.stepgen.04.stepspace      [AXIS_4]STEPSPACE
setp   hm2_5i25.0.stepgen.04.position-scale  [AXIS_4]STEP_SCALE
setp   hm2_5i25.0.stepgen.04.step_type       0
setp   hm2_5i25.0.stepgen.04.control-type    0
setp   hm2_5i25.0.stepgen.04.maxaccel        50.0
setp   hm2_5i25.0.stepgen.04.maxvel          41.7

net b-pos-fb     axis.4.motor-pos-fb  <=  hm2_5i25.0.stepgen.04.position-fb
net b-pos-cmd    axis.4.motor-pos-cmd  =>  hm2_5i25.0.stepgen.04.position-cmd
net b-enable     axis.4.amp-enable-out =>  hm2_5i25.0.stepgen.04.enable

# ---setup home / limit switch signals---
net max-home-b      =>  axis.4.home-sw-in
net b-pos-limit     =>  axis.4.pos-lim-sw-in
net max-home-b      =>  axis.4.pos-lim-sw-in



#*******************************************************************************
# Axis C
#*******************************************************************************
# Step Gen signals/setup
setp   hm2_5i25.0.stepgen.03.dirsetup       [AXIS_5]DIRSETUP
setp   hm2_5i25.0.stepgen.03.dirhold        [AXIS_5]DIRHOLD
setp   hm2_5i25.0.stepgen.03.steplen        [AXIS_5]STEPLEN
setp   hm2_5i25.0.stepgen.03.stepspace      [AXIS_5]STEPSPACE
setp   hm2_5i25.0.stepgen.03.position-scale  [AXIS_5]STEP_SCALE
setp   hm2_5i25.0.stepgen.03.step_type       0
setp   hm2_5i25.0.stepgen.03.control-type    0
setp   hm2_5i25.0.stepgen.03.maxaccel        125.0
setp   hm2_5i25.0.stepgen.03.maxvel          20.8

net c-pos-fb     axis.5.motor-pos-fb  <=  hm2_5i25.0.stepgen.03.position-fb
net c-pos-cmd    axis.5.motor-pos-cmd  =>  hm2_5i25.0.stepgen.03.position-cmd
net c-enable     axis.5.amp-enable-out =>  hm2_5i25.0.stepgen.03.enable
```

```
# ---setup home / limit switch signals---
net c-home-sw       =>  axis.5.home-sw-in
net c-neg-limit     =>  axis.5.neg-lim-sw-in
net c-pos-limit     =>  axis.5.pos-lim-sw-in


#############################################################################
# Spindle
#############################################################################
# ---setup spindle control signals---
net spindle-vel-cmd-rps    <=  motion.spindle-speed-out-rps
net spindle-vel-cmd        <=  motion.spindle-speed-out
net spindle-enable         <=  motion.spindle-on
net spindle-cw             <=  motion.spindle-forward
net spindle-ccw            <=  motion.spindle-reverse
net spindle-brake          <=  motion.spindle-brake
net spindle-revs           =>  motion.spindle-revs
net spindle-at-speed       =>  motion.spindle-at-speed
net spindle-vel-fb         =>  motion.spindle-speed-in
net spindle-index-enable  <=>  motion.spindle-index-enable

# ---Setup spindle at speed signals---
sets spindle-at-speed true
#  Use COMMANDED spindle velocity from LinuxCNC because no spindle encoder was
specified
#  COMMANDED velocity is signed so we use absolute component to remove sign

net spindle-vel-cmd         =>    abs.spindle.in
net absolute-spindle-vel    <=    abs.spindle.out


#############################################################################
# Connect Miscellaneous Signals
#############################################################################
#  ---HALUI signals---
net joint-select-a        halui.joint.0.select
net x-is-homed            halui.joint.0.is-homed
net jog-x-pos             halui.jog.0.plus
net jog-x-neg             halui.jog.0.minus
net jog-x-analog          halui.jog.0.analog
net joint-select-b        halui.joint.1.select
net y-is-homed            halui.joint.1.is-homed
net jog-y-pos             halui.jog.1.plus
net jog-y-neg             halui.jog.1.minus
```

```
net jog-y-analog          halui.jog.1.analog
net joint-select-c        halui.joint.2.select
net z-is-homed            halui.joint.2.is-homed
net jog-z-pos             halui.jog.2.plus
net jog-z-neg             halui.jog.2.minus
net jog-z-analog          halui.jog.2.analog
net joint-select-d        halui.joint.4.select
net b-is-homed            halui.joint.4.is-homed
net jog-b-pos             halui.jog.4.plus
net jog-b-neg             halui.jog.4.minus
net jog-b-analog          halui.jog.4.analog
net joint-select-e        halui.joint.5.select
net c-is-homed            halui.joint.5.is-homed
net jog-c-pos             halui.jog.5.plus
net jog-c-neg             halui.jog.5.minus
net jog-c-analog          halui.jog.5.analog
net jog-selected-pos      halui.jog.selected.plus
net jog-selected-neg      halui.jog.selected.minus
net spindle-manual-cw     halui.spindle.forward
net spindle-manual-ccw    halui.spindle.reverse
net spindle-manual-stop   halui.spindle.stop
net machine-is-on         halui.machine.is-on
net jog-speed             halui.jog-speed
net MDI-mode              halui.mode.is-mdi

#  ---coolant signals---
net coolant-mist      <=  iocontrol.0.coolant-mist
net coolant-flood     <=  iocontrol.0.coolant-flood

#  ---probe signal---
net probe-in     =>  motion.probe-input

#  ---motion control signals---
net in-position              <=  motion.in-position
net machine-is-enabled       <=  motion.motion-enabled

#  ---estop signals---
net estop-out     <=  iocontrol.0.user-enable-out
net estop-out     =>  iocontrol.0.emc-enable-in

#  ---manual tool change signals---
loadusr -W hal_manualtoolchange
net tool-change-request     iocontrol.0.tool-change       =>  hal_manualtoolchange.change
net tool-change-confirmed   iocontrol.0.tool-changed      <=  hal_manualtoolchange.changed
net tool-number             iocontrol.0.tool-prep-number  =>  hal_manualtoolchange.number
net tool-prepare-loopback   iocontrol.0.tool-prepare      =>  iocontrol.0.tool-prepared
```

```
#*****************************************************************************
# Kinematics
#*****************************************************************************
# ---Extras for XYZBC 5 axis Configuration
setp XYZBCkins.tool-length 2.61134
setp XYZBCkins_V1.X3 13.25
setp XYZBCkins_V1.X7 1.07865
setp XYZBCkins_V1.Y2 15.75
setp XYZBCkins_V1.Y7 0.5187
setp XYZBCkins_V1.Z1 1.68898
setp XYZBCkins_V1.Z2 0.78740
setp XYZBCkins_V1.Z3 24.75
setp XYZBCkins_V1.Z5 0.63435
setp XYZBCkins_V1.Z6 4.43812
setp XYZBCkins_V1.Z7 1.84730
```

XYZBCkins_V1.*c*

```
/********************************************************************
*
* Description: XYZBCkins_V1.c
* Mahcine Geometry:XYZBC
* Kinematics for 5 axis Spindle Tilting Rotating Router
* Author: Matthew Osborne
* License: GPL Version 2
* System: LinuxCNC 2.5 Dev
*
********************************************************************/
#include "kinematics.h" /* these decls */
#include "posemath.h"
#include "hal.h"
#include "rtapi_math.h"

#define d2r(d) ((d)*PM_PI/180.0)
#define r2d(r) ((r)*180.0/PM_PI)

struct haldata {
  hal_float_t *tool_length;
  hal_float_t *X3;
  hal_float_t *X7;
  hal_float_t *Y2;
  hal_float_t *Y7;
  hal_float_t *Z1;
  hal_float_t *Z2;
  hal_float_t *Z3;
  hal_float_t *Z5;
  hal_float_t *Z6;
  hal_float_t *Z7;
} *haldata;

int kinematicsForward(const double *joints,
  EmcPose * pos,
  const KINEMATICS_FORWARD_FLAGS * fflags,
  KINEMATICS_INVERSE_FLAGS * iflags)
{
double thB;
double thC;
double TL;
double X3;
double X7;
double Y2;
double Y7;
double Z1;
```

```
double Z2;
double Z3;
double Z5;
double Z6;
double Z7;

thB = d2r(joints[4]);
thC = d2r(joints[5]);
TL = *(haldata->tool_length);
X3 = *(haldata->X7);
X7 = *(haldata->X7);
Y2 = *(haldata->Y2);
Y7 = *(haldata->Y7);
Z1 = *(haldata->Z1);
Z2 = *(haldata->Z2);
Z3 = *(haldata->Z3);
Z5 = *(haldata->Z5);
Z6 = *(haldata->Z6);
Z7 = *(haldata->Z7);

  pos->tran.x = cos(thC)*cos(thB)*X7-sin(thC)*Y7-(Z7+TL)*cos(thC)*sin(thB)+joints[0]-X3;
  pos->tran.y = cos(thC)*cos(thB)*X7-cos(thC)*Y7-(Z7+TL)*sin(thC)*sin(thB)+Y2-joints[1];
  pos->tran.z = Z3-Z1-Z2-joints[2]-Z5-Z6-sin(thB)*X7-(Z7+TL)*cos(thB);
  pos->b = joints[4];
  pos->c = joints[5];


  return 0;
}

int kinematicsInverse(const EmcPose * pos,
  double *joints,
  const KINEMATICS_INVERSE_FLAGS * iflags,
  KINEMATICS_FORWARD_FLAGS * fflags)
{

  double thB;
  double thC;
  double TL;
  double X3;
  double X7;
  double Y2;
  double Y7;
  double Z1;
  double Z2;
  double Z3;
```

```
  double Z5;
  double Z6;
  double Z7;

  thB = d2r(joints[4]);
  thC = d2r(joints[5]);
  TL = *(haldata->tool_length);
  X3 = *(haldata->X3);
  X7 = *(haldata->X7);
  Y2 = *(haldata->Y2);
  Y7 = *(haldata->Y7);
  Z1 = *(haldata->Z1);
  Z2 = *(haldata->Z2);
  Z3 = *(haldata->Z3);
  Z5 = *(haldata->Z5);
  Z6 = *(haldata->Z6);
  Z7 = *(haldata->Z7);

  joints[0] = -cos(thC)*cos(thB)*X7+sin(thC)*Y7+(Z7+TL)*cos(thC)*sin(thB)+(pos-
>tran.x)+X3;
  joints[1] = sin(thC)*cos(thB)*X7+cos(thC)*Y7-(Z7+TL)*sin(thC)*sin(thB)-Y2-(pos->tran.y);
  joints[2] = Z3-Z1-Z2-Z5-(pos->tran.z)-Z6-sin(thB)*X7-(Z7+TL)*cos(thB);
  joints[4] = pos->b;
  joints[5] = pos->c;


  return 0;
}

/* implemented for these kinematics as giving joints preference */
int kinematicsHome(EmcPose * world,
  double *joint,
  KINEMATICS_FORWARD_FLAGS * fflags,
  KINEMATICS_INVERSE_FLAGS * iflags)
{
  *fflags = 0;
  *iflags = 0;

  return kinematicsForward(joint, world, fflags, iflags);
}

KINEMATICS_TYPE kinematicsType()
{
  return KINEMATICS_BOTH;
}
```

```
#include "rtapi.h" /* RTAPI realtime OS API */
#include "rtapi_app.h" /* RTAPI realtime module decls */
#include "hal.h"

EXPORT_SYMBOL(kinematicsType);
EXPORT_SYMBOL(kinematicsForward);
EXPORT_SYMBOL(kinematicsInverse);
MODULE_LICENSE("GPL");

int comp_id;
int rtapi_app_main(void) {
  int result;
  comp_id = hal_init("XYZBCkins_V1");
  if(comp_id < 0) return comp_id;

  haldata = hal_malloc(sizeof(struct haldata));

  result = hal_pin_float_new("XYZBCkins_V1.tool-length", HAL_IN,
    &(haldata->tool_length), comp_id);
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.X3", HAL_IO,
    &(haldata->X3), comp_id);
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.X7", HAL_IO,
    &(haldata->X7), comp_id);
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.Y2", HAL_IO,
    &(haldata->Y2), comp_id);
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.Y7", HAL_IO,
    &(haldata->Y7), comp_id);
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.Z1", HAL_IO,
    &(haldata->Z1), comp_id);
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.Z2", HAL_IO,
    &(haldata->Z2), comp_id);
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.Z3", HAL_IO,
    &(haldata->Z3), comp_id);
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.Z5", HAL_IO,
    &(haldata->Z5), comp_id);
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.Z6", HAL_IO,
    &(haldata->Z6), comp_id);
```

```
  if(result < 0) goto error;
  result = hal_pin_float_new("XYZBCkins_V1.Z7", HAL_IO,
    &(haldata->Z7), comp_id);
  if(result < 0) goto error;


  hal_ready(comp_id);
  return 0;

error:
  hal_exit(comp_id);
  return result;
}

void rtapi_app_exit(void) { hal_exit(comp_id); }
```

VISMACH_5AXIS_SIMULATION.py

```
#-----------------------------------------------------------------------
#
# Description: VISMACH 5 axis rotating/tilting spindle router simulation
#Author: Matthew Osborne
#Date: March 13 ,2018
#Version: LinuxCNC 2.5 Dev
#License: GPL Version 2
#
#-----------------------------------------------------------------------

from vismach import *
import hal
import math
import sys

# give endpoint Z values and radii
# resulting cylinder is on the Z axis
class HalToolCylinder(CylinderZ):
   def __init__(self, comp, *args):
CylinderZ.__init__(self, *args)
self.comp = comp

   def coords(self):
      return self.comp.tool_length, 3, 0, 3

c = hal.component("sptiltgui")
c.newpin("joint0", hal.HAL_FLOAT, hal.HAL_IN)
c.newpin("joint1", hal.HAL_FLOAT, hal.HAL_IN)
c.newpin("joint2", hal.HAL_FLOAT, hal.HAL_IN)
c.newpin("joint3", hal.HAL_FLOAT, hal.HAL_IN)
c.newpin("joint4", hal.HAL_FLOAT, hal.HAL_IN)
c.newpin("joint5", hal.HAL_FLOAT, hal.HAL_IN)

c.newpin("tool_length", hal.HAL_FLOAT, hal.HAL_IN)
c.newpin("pivot_length", hal.HAL_FLOAT, hal.HAL_IN)
c.ready()

pivot_len = 50
tool_len = 40
Lt = pivot_len + tool_len

for setting in sys.argv[1:]: exec setting

tooltip = Capture()
tool = Collection([tooltip,
```

```
                HalToolCylinder(c,"tool_length"),
                CylinderZ(tool_len, 30, Lt+70, 30)
                ])

baxis = CylinderY(-65,10,65,10)
baxis = Translate([baxis],0,0,Lt)

baxis = Collection([tool,
                    baxis,
                    ])
baxis = Color([1,0,0.5,1],[baxis])
baxis = Translate([baxis],0,0,-Lt)
baxis = HalRotate([baxis],c,"joint4",1,0,1,0)
baxis = Translate([baxis],-130,0,Lt)

yoke = Collection([Box(-30,-32,0, 30,-52,-90),
                Box(-30, 32,0, 30, 52,-90),
                CylinderZ(0, 56,-15,56)
                ])
yoke = Translate([yoke],-130,0,Lt+90)
yoke = Color([1,0.5,0,1],[yoke])

caxis = Collection([baxis,
                    yoke,
                    ])
caxis = Translate([caxis],130,0,0)
caxis = HalRotate([caxis],c,"joint5",1,0,0,1)
caxis = Translate([caxis],-130,0,0)

bmotor = CylinderZ(0,25,60,25)
bmotor = Translate([bmotor],20,30,0)
cmotor = CylinderZ(0,25,60,25)
cmotor = Translate([cmotor],20,-30,0)

zaxis = Collection([cmotor,bmotor,
                    CylinderZ(15,60,0,60),
                    Box(50,-60,-50, 80,60,60),
                    Box(0,-60,15, 50,60,0)
                    ])

zaxis = Translate([zaxis],-130,0,Lt+90)
zaxis = Color([0,1,1,1],[zaxis])
zaxis = Collection([caxis,zaxis,
                   ])

zaxis = Translate([zaxis], 0,0,-150)
```

```
zaxis = HalTranslate([zaxis],c,"joint2",0,0,1)

zmotor = CylinderZ(150,25,90,25)
zmotor = Translate([zmotor],10,0,0)
saddle = Collection([zaxis,zmotor,
                Box(-10,-50,-100, -50,50,150)
                ])

saddle = HalTranslate([saddle],c,"joint1",0,1,0)

ybridge = Collection([
            Box(-60,255,-350, 60,285,0),
            Box(-60,-255,-350, 60,-285,0),
            Box(-30,-280,-100, 30,280,0),
            Box(-70,255,-350, 70,295,-320),
            Box(-70,-255,-350, 70,-295,-320),
            ])

ybridge = Color([0,1,0,1],[ybridge])
ybridge = Collection([saddle,ybridge])

ybridge = HalTranslate([ybridge],c,"joint0",1,0,0)

xaxes = Collection([
                Box(-500,-300,-400, 500, -250,-350),
                Box(-500, 300,-400, 500, 250,-350)
                ])

xaxes = Color([0,0,1,1],[xaxes])
router = Collection([xaxes,ybridge])

work = Capture()
table = Collection([
work,
Box(-500,-250,-320, 500,250,-365)
      ])

model = Collection([router, table])

main(model, tooltip, work, 1500)
```

Appendix C: Arduino Spindle Speed Control Code

```
#-------------------------------------------------------------------------------
#
#Author: Talal Alatiah
#Date: April 26,2018
#Version: Arduino 1.86
#License: GPL Version 2
#
#-------------------------------------------------------------------------------
float temp;

void setup() {
  Serial.begin(9600);

  //Setup Channel A
  pinMode(12, OUTPUT); //Initiates Motor Channel A pin
  pinMode(9, OUTPUT); //Initiates Brake Channel A pin
  pinMode(A1, INPUT);
  Serial.println("-------");

  for(int i = 0; i<10; i++){
  delay(1000);
  temp = analogRead(A1) * (255 / 1023.0);
  }
}

void loop(){
  float voltage = analogRead(A1) * (255 / 1023.0);

  digitalWrite(12, HIGH); //Establishes forward direction of Channel A
  digitalWrite(9, LOW);   //Disengage the Brake for Channel A

  //Serial.println(voltage);

  if(voltage > temp){
  analogWrite(3,255);
  } else {
    analogWrite(3,0);
    digitalWrite(12, LOW);
    digitalWrite(9, HIGH);
  }
  delay(100);
}
```

Appendix D: Stippling Mechanism: Bearing Selection

The max force (F_{MAX}) is 2.528 lbf while the minimum force (F_{MIN}) is -2.528lbf when the soldering iron deforms the plastic.

$$F_{mean} = \frac{F_{MAX} + F_{MIN}}{2} = 0 \quad (1)$$

$$F_{alt} = \frac{F_{MAX} - F_{MIN}}{2} = 2.526 \text{ lbf} \quad (2)$$

Based off these values the fatigue limit for the pin can be found where the formula to find the fatigue limit is as follows:

$$S_n = S_n' C_L C_G C_S C_T C_R \quad (3)$$

Where S_n is the fatigue limit of the material,

S_n' is the R.R. Moore Endurance limit,

C_L is the load correction factor,

C_G is the gradient factor,

C_G is the surface factor,

C_S is the surface factor,

C_T is the temperature factor,

And C_R is the reliability factor.

The endurance limit can be found by using the following formula

$$S_n' = .5S_u$$

Where S_u is the ultimate tensile strength of the material

The fatigue limit at 99% reliability is compared for the materials listed below as well as including the differing surface and ultimate tensile strengths. The factors are also listed below in

Table 11

Material	S_u (Ksi)	C_S	S_n (KSI)
Stainless hot rolled steel	73.2	0.65	19.37
Stainless cold drawn steel	73.2	0.77	22.94
carbon hot rolled steel	42	0.73	12.48
carbon cold drawn steel	42	0.8	13.68

Table 12

C_L	1
C_G	1
C_T	1
C_R	0.814

Based off the results hot rolled carbon steel will be used for further analysis due to the low endurance limit. Next the required diameter for the bearing was selected by analyzing the loading on the pin the bearing was going to be pressed on. The loading is shown in Figure below.

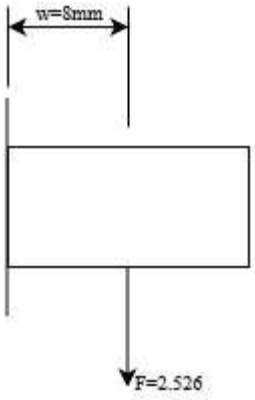


Figure 19 Bearing Pin Force Diagram

$$\mathrm{M} = \mathrm{F}\frac{w}{2}$$

$$\sigma_{bend} = \frac{32\ M}{\pi\ d^3}$$

$$\mathrm{SF} = \frac{S_n}{\sigma_{bend}}$$

$$\mathrm{SF} = \frac{\pi\ S_n\ d^3}{32\ M} \therefore \mathrm{d} = \sqrt[3]{\frac{SF\ 32\ M}{\pi\ S_n}}$$

Using these values a bearing size was determined, then next the bearing life was found using the following formula.

$$L = K_r L_R \left(\frac{C}{F_e\ K_\alpha}\right)^{3.33}$$

Where L is the life,

K_r is the reliability factor,

L_R is the life corresponding to rated capacity

C is the dynamic loading capacity,

F_e is the equivalent loading,

K_α is the application factor.

An R3 bearing and acetal plastic bearing were compared using the previous formula to get the number of cycles before failure. Then the the lifespan was found if the stippling mechanism was ran at frequency of 20 Hz. The results are shown in the table below that show that the R3 bearing would be most suitable for the task.

Table 13

	C (lbf)	K_r	K_α	L_R (cycles	F_e (lbf)	L (cycles)	Life Span (hrs)
R3	290	0.21	1.5	10^6	2.586	3.94×10^{11}	5.97×10^6
Acetal	15	0.21	1.2	10^6	2.586	1×10^6	599