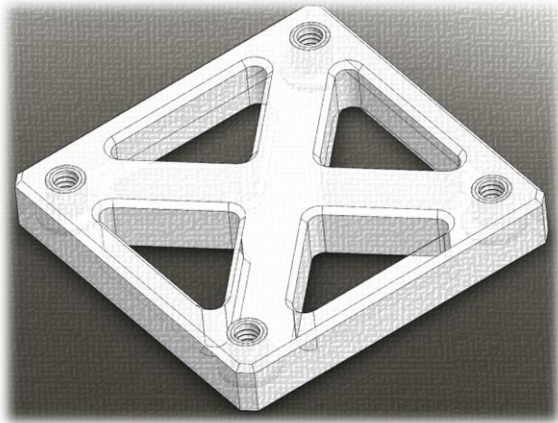


# TEAM ADD\_IN

12.17.2015



## Critical Design Review

### TEAM F

- Nikhil Baheti
- Daniel Berman
- Ihsane Debbache
- Astha Prasad

## Table of Contents

<b>1. Project Description .....</b>	<b>3</b>
<b>2. Use Case .....</b>	<b>3</b>
<b>3. System Level Requirements .....</b>	<b>6</b>
<b>3.1 Software Requirements.....</b>	<b>7</b>
<b>3.2 Hardware Requirements.....</b>	<b>8</b>
<b>4. Functional Architecture .....</b>	<b>10</b>
<b>4.1 Slicing Software .....</b>	<b>11</b>
<b>4.2 Printer Interface Software.....</b>	<b>11</b>
<b>4.3 Printer (Electromechanical System) .....</b>	<b>11</b>
<b>5. Cyberphysical Architecture .....</b>	<b>12</b>
Description .....	13
<b>6. Current System Status.....</b>	<b>13</b>
<b>Subsystem Requirements.....</b>	<b>13</b>
<b>6.1 Nozzle Subsystem.....</b>	<b>14</b>
Description .....	14
Status .....	15
Analysis.....	15
FVE Evaluation.....	16
Strengths and Weaknesses .....	17
<b>6.2 Rotary Joint Subsystem .....</b>	<b>18</b>
Description .....	18
Status .....	18
Analysis.....	19
FVE Evaluation.....	20
Strengths and Weaknesses .....	20
<b>6.3 Microcontroller Subsystem.....</b>	<b>20</b>
Description .....	20
Status .....	22
FVE Evaluation.....	23
Strengths and Weaknesses .....	23
<b>6.4 Software Subsystem.....</b>	<b>23</b>
Layer Selection Algorithm .....	23
Path Planning Algorithm.....	26
<b>7. Project Management.....</b>	<b>29</b>
<b>7.1 Work Breakdown Structure .....</b>	<b>29</b>
Nozzle Subsystem .....	29
Rotary Subsystem.....	29
Microcontroller Subsystem .....	29
Slicer Subsystem .....	29
System Level Tasks.....	<b>Error! Bookmark not defined.</b>
<b>7.2 Project Schedule .....</b>	<b>30</b>
<b>7.3 Spring Semester Test plan .....</b>	<b>31</b>

Spring Validation Experiments .....	31
<b>7.4 Budget.....</b>	<b>33</b>
<b>7.5 Risk Management.....</b>	<b>34</b>
Risk Management Process .....	34
Current 4DOF Risks.....	35
Changes since PDR.....	35
<b>8. Conclusions.....</b>	<b>36</b>
<b>Key Lessons Learned.....</b>	<b>36</b>
<b>Key Activities for Spring Semester .....</b>	<b>36</b>

## 1. Project Description

The ADD\_IN team aims to develop a 3D printer technology (named 4DOF) capable of producing printed parts which incorporate commercial off the shelf (COTS) components. These COTS items can include components such as threaded inserts, structural reinforcement (stiffeners), and electronics. Incorporating these components enables inclusion of geometrically precise features (such as threads), production of stronger parts (via the use of stiffeners), and reduced overall production time (since post-printing operations such as tapping and gluing can be eliminated). It will also open up new applications for 3D printing by making it possible to build complex parts that are hard or impossible to make otherwise.

The parts to be incorporated will be generally cylindrical or rectangular prism in shape, can optionally extend up to 1” above the top layer of the part, and must have sides which are orthogonal to the print surface. The 3D print material will be precisely deposited along the perimeter of the COTS item, thus encapsulating it within the 3D printed part.

Within the scope of this project, the installation of COTS parts during the printing process will be performed by a human operator. When the 3D printer reaches the print layer on which a part is to be installed, it will ‘pause’, move to a safe and accessible configuration for COTS item insertion, and await a command from the user that the part has been successfully installed before resuming printing. The system will be able to produce printed parts incorporating multiple COTS items at varying locations and orientations, provided that the orientations and clearances between the parts do not violate certain geometric restrictions.

## 2. Use Case

An RC plane engineering company is in a race to develop a new fixed-wing airplane. The general configuration of the airplane has been determined but continued iteration on the structure and aerodynamic surfaces is needed. Up until this point the airplane has been prototyped using individually cut pieces of foam, balsa wood, and aluminum. Even with the aid of a laser cutter and CNC hot-wire foam cutter this has been a time consuming process since many of the parts are made from vastly different materials and are difficult to join together. Because of this the engineering team is often using glues and composites to hold the parts together, which makes it near impossible to replace or modify parts thus making design iteration difficult.

As with their other fixed wing airplanes, and as is an industry norm, the company eventually aims to mass produce the design from aluminum and polystyrene. Aluminum provides the structural rigidity necessary to form a solid airframe, especially for enduring the high stresses present at the motor mounts. Polystyrene is a light-weight plastic which can be easily molded into the complex shapes necessary to produce the aerodynamic surfaces of the aircraft. Furthermore, polystyrene can be injection molded to contain thread inserts to provide attachment



Figure 1: Injection molded part with threaded insert. (<http://precision.bc.ca/photos/>)

points to the airframe and other internal hardware. This however, is only cost effective for large production runs.

The design team has previously tried using 3D printers to more rapidly produce and test structural components. While helpful in some instances, the team has generally found that directly replacing their wood-polystyrene-composite prototypes with 3D printed parts often leads to failure at the mechanical interfaces between parts. Efforts to work around this problem have been made, but usually involve designing a special ‘thickened’ version of the part for 3D printing, which must then be redesigned when transitioning to large volume manufacturing using polystyrene. This extra design step uses up valuable team member time and changes the weight, balance, and aeroelastic properties of the aircraft. The team however has remarked that 3D printing’s ability to easily incorporate more complex features, such as wire runs, electronics mounts, and access ports into their project is a great benefit.

Recently, as part of an experiment to improve their design process, the aircraft design team purchased an ADD\_IN 4DOF 3D printer. This printer contains an extra mechanical degree of freedom and a special slicing software to enable 3D printing of complex parts that include embedded COTS items. The cost of the printer was comparable to other FDM 3D printers, and thus was not a significant burden to the small company. Since the printer looked and operated in nearly the same manner as their other 3D printers, the employees were immediately able to start using the printer without any additional training.



Figure 2: Example of failed 3D printed motor mount and a less desirable ‘thickened’ replacement necessary to support the loads. (<http://www.wattflyer.com>)

The first project the 4DOF printer was applied to was development of the aircraft’s motor mount. The motor mount is a particularly challenging component since it must be strong enough to withstand large forces and torques imparted by the motor, propeller, and occasional crash landings, but also be light weight since it is the most forward part of the aircraft where weight is a premium. In these regards 3D printed parts had previously failed due to their poor material properties, especially with the motor mounting bolts loosening, stripping and pulling out from the plastic part. Using the ADD\_IN printer the team was able to produce a 3D printed part directly from the airplane’s CAD design, and include threaded inserts which could securely attach the motor and distribute the forces within the motor mount thus preventing material failure. The new motor mount was produced in half the time and with fewer resources than the previous aluminum-balsa wood and glue version, and thus enabled the team to conduct more frequent tests and design iterations.



Figure 4: Time intensive aluminum and balsa motor mount. (<http://www.rcgroups.com>)

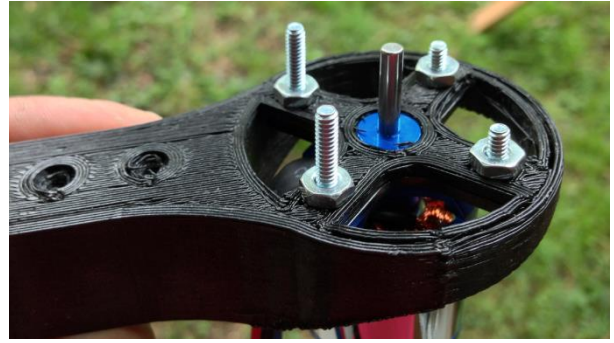


Figure 3: Example 3D printed motor mount with stripped threads (<https://backyardrc.wordpress.com/category/quadcopter/>)



Figure 5: Motor mount made with ADD\_IN printer (<http://www.anzel360.com/category/rc/fixed-wing/>)

In another instance, the engineering team decided to experiment with using the 4DOF printer to produce a wing rib. This particular wing rib had been causing difficulty since it established both the aerodynamic profile of the wing, but also was the structural mount for the mechanism that actuated the ailerons. Because of this it needed to be particularly rigid, and provide mounting for numerous bearings, sensors, and the aileron motor, and have a precise complex outer geometry to define the airfoil shape of the wing. All of these requirements had made prototyping in balsa-wood particularly difficult, and often the team had to pay large sums to have the part CNC machined.

Immediately, the 4DOF printer was able to overcome these issues. Incorporation of screw inserts, aluminum stiffeners, and a special aluminum socket for connection with the wing spar provided the structural rigidity needed. Because 4DOF printers can produce complex 3-dimensional parts, the bearing and motor mounts needed were not only incorporated into the wing rib, but actually replaced with integrated features in the rib which directly held the bearings and motor and reduced the overall part count. Finally, since the 4DOF printer can print around complex shapes, even the sensors used in the aileron actuation mechanism could be incorporated directly into the wing rib, thus greatly simplifying the whole assembly. Again, producing the wing rib required only a fraction of the time as needed for former methods, and the result achieved was



much higher than previously achievable. Rapid design iteration perfected the part, and helped to significantly reduce the time-to-market of the RC airplane engineering company’s newest airplane.

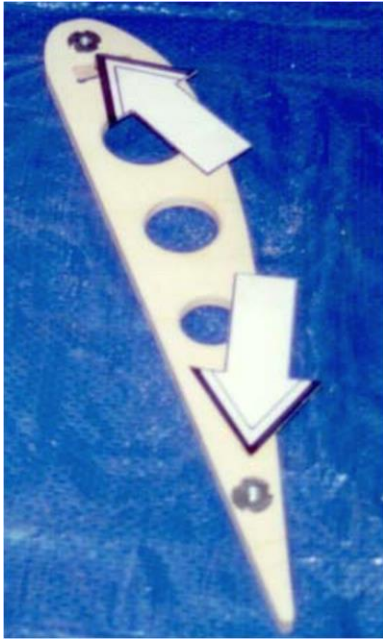


Figure 6: Plastic wing rib with added screw inserts. 4DOF printed ribs can have both more complex geometries and integrated screw inserts (<https://sidewalkfliersag.wordpress.com/>)

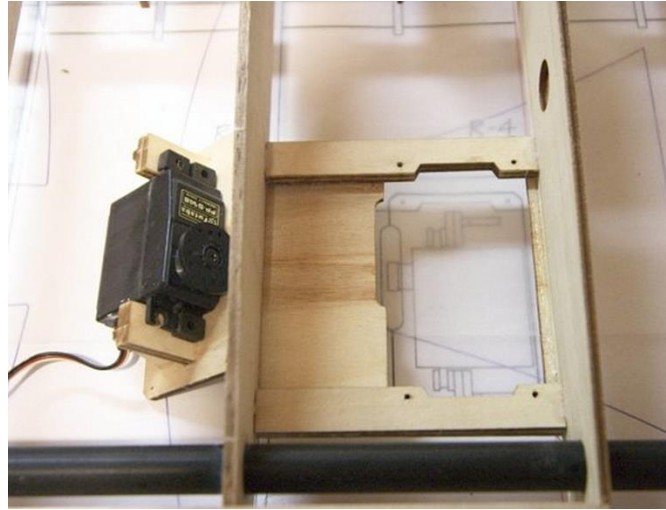


Figure 7: Balsa wood wing rib with complex aileron-servo mount

The case of the fixed-wing drone manufacturer is not unique. Thousands of companies today extensively rely on 3D printing to rapidly prototype parts and iterate designs. Improving the efficiency of this process even a small amount can have multiplicative effects on the quality of a resulting product. Strong, lightweight 3D printed parts and parts with integrated sensors and electronics represent a significant new advance in 3D printing technology that will help to provide rapid prototyping, mass customization, and lower cost parts to a variety of both businesses and consumers.

### 3 System Level Requirements

The system is divided into two primary focuses, namely the ‘Software’ and ‘Hardware’. System requirements are mapped to either of these sections. Requirements that have been introduced since the CODR are indicated in orange. Within Software and Hardware, the requirements are grouped under the categories listed in Table 1.

Table 1 : Requirements Labeling Legend

ID	Category
M	Mandatory (Threshold requirement)
D	Desirable (Stretch Goal)
F	Functional

<b>N</b>	Non Functional
<b>P</b>	Performance

### 3.1 Software Requirements

**The Software System shall –**

*Table 2: Software Requirements*

ID	Requirement	Subsystem
<b>Software Mandatory Functional Requirements</b>		
M.F.1	<b>- Receive 3D files</b>	Slicer
	The execution shall begin once the 3D file of the assembly (part to be printed + COTS item(s)) has been fed to the Slicer program.	
M.F.2	<b>- Prompt user for insertion layer</b>	Slicer
	Slicer shall request the user to input the decided insertion layer. Upon reaching this layer, the printer will pause and move to a safe position and allow the COTS item to be inserted.	
M.F.3	<b>- Create collision free path</b>	Slicer
	With knowledge of the dimensions and position of the COTS item, the Slicer program shall create a collision free path around the same.	
M.F.4	<b>- Generate 4DOF G-Code</b>	Slicer
	The G-Code is comprised of movement commands for each of the 4 DOFs The printer interface software will load this G-Code before printing.	
<b>Software Mandatory Nonfunctional Requirements</b>		
M.N.1	<b>- Work with standard 3D print files</b>	Slicer
	The slicing program shall be compatible with various standard 3D files Examples include .stl, .obj and .amf	
M.N.2	<b>- Slicing, Insertion Layer Selection, and Path Planning to implemented in single software application</b>	Slicer
	The slicing program shall be integrated to a single software application to allow the user to use it easily.	
<b>Software Desirable Functional Requirement</b>		
D.F.1	<b>- Assign insertion layer</b>	Slicer
	A desirable feature of the slicing program would be the capability to assign insertion layers autonomously based on the dimensions and orientation of the COTS item. This would greatly reduce the program's dependence on humans.	



Software Desirable Non Functional		
D.N.1	<b>- Easy-to-use interface</b>	Slicer
	Since the system requires user interaction, it is necessary to have an intuitive graphical user interface.	

Software Desirable Performance Requirements		
D.P.1	<b>- Print time not to increase more than 5% compared to Makergear M2</b>	Slicer
	The total print time for the part should not increase more than 5% when compared to the current print time. This does not include time to add the COTs part.	
D.P.2	<b>- Print between multiple COTS parts not less than 2 inches apart</b>	Slicer
	The path planning algorithm must be able to path plan for multiple COTs part to be added	

### 3.2 Hardware Requirements

**The Hardware System shall –**

*Table 3: Hardware Requirements*

Hardware Mandatory Functional Requirements		
M.F.1	<b>- Print layers of material</b>	Nozzle
	The primary function of the 3D printer, upon which our system is developed, must be upheld post alterations to the manipulator and other system components.	
M.F.2	<b>- Print locating features for COTS items</b>	Firmware
	Accurate positioning of an item on a partial print is achieved by printing locating features tailored to the base dimensions of the said item.	
M.F.3	<b>- Go to safe configuration during insertion</b>	Firmware
	Since the addition of the COTS part involves a human placing the part on a partial print, measures shall be taken to ensure that the hot nozzle is out of the way and that the partially printed part is easily accessible. The above shall include: <ul style="list-style-type: none"> <li>i. Moving the nozzle to safe location (away from printed part)</li> <li>ii. Fully lowering the bed to the insertion position</li> </ul>	
M.F.4	<b>- Enclose COTS item with print material</b>	Rotary Stage
	The printer shall deposit print material close to the added part such that there is no wiggle room around the piece for it to move, allowing for a snug fit.	
M.F.5	<b>- Avoid Collisions</b>	Rotary Stage
	Since the added COTS part will be protruding above the top surface of the part, the movement of the printer nozzle will be controlled to avoid collisions with it.	
M.F.6	<b>- Rotate nozzle Using G-code</b>	Firmware
	The nozzle must be rotated to avoid the cots item which will be taken care by the RAMBo board in planning the trajectory based on the path planners output.	
M.F.7	<b>- Avoid Kinks in Filament</b>	Rotary Stage
	The rotary stage must not kink the filament while rotating.	

Hardware Mandatory Non Functional Requirements		
M.N.1	<b>- Provide user feedback during printing</b>	LCD screen
	The user will be provided with a real time display of the progression of the print as the material gets deposited along the coordinates provided by the G-Code.	
M.N.2	<b>- Maintain accurate temperature control</b>	

	The slip ring adds noise to the analog data. Thus there must be a system that makes the data through the slip ring immune to noise.	<b>PCB</b>
--	-------------------------------------------------------------------------------------------------------------------------------------	------------

<b>Hardware Desirable Functional Requirements</b>		
<b>D.F.1</b>	<b>- Print between parts</b>	<b>Nozzle</b>
	The nozzle should be capable of maneuvering between two added COTS items. The items shall not be more than <b>2 inches apart</b> .	
<b>D.F.2</b>	<b>- Print close to the COTS item</b>	<b>Rotary Stage</b>
	The printer should print within <b>0.1 mm</b> of the COTS part	

<b>Hardware Desirable Non Functional Requirements</b>		
<b>D.N.1</b>	<b>- Maintain print speed</b>	<b>Rotary Stage</b>
	The 4DOF printer should not have a print speed lower than that of the unmodified printer.	

The performance requirements are as follows:

<b>Hardware Mandatory Performance Requirements</b>		
<b>M.P.1</b>	<b>- Incorporate COTS parts that are orthogonal to print surface</b>	<b>Nozzle</b>
	The hardware must be able to print against parts that are orthogonal to the print plane. Examples: Cylindrical and rectangular prism shapes	
<b>M.P.2</b>	<b>- Incorporate COTS parts that have a maximum height of one inch above the print plane</b>	<b>Nozzle</b>
	The COTS part must have a maximum height of 1 inch because beyond that we need to move along negative z to avoid collision	
<b>M.P.3</b>	<b>- Print volume of 3x3x3 inches</b>	<b>Nozzle</b>
	Since the nozzle is being modified, this should not reduce the print volume of the printer significantly	
Hardware Mandatory Performance Requirements		
<b>M.P.4</b>	<b>- Be able to infinitely rotate nozzle</b>	<b>Rotary Stage</b>
	The wires moving from through the rotary joint should not hinder the rotation and thus the rotation must be done infinitely	

<b>Hardware Desired Performance Requirements</b>		
<b>D.P.1</b>	<b>- Position nozzle within 0.1mm of COTS part</b>	<b>Rotary Stage</b>
	This is the general accuracy of a 3D printer and thus the printer must be able to print within 0.1mm of the COTS part	

#### 4. Functional Architecture

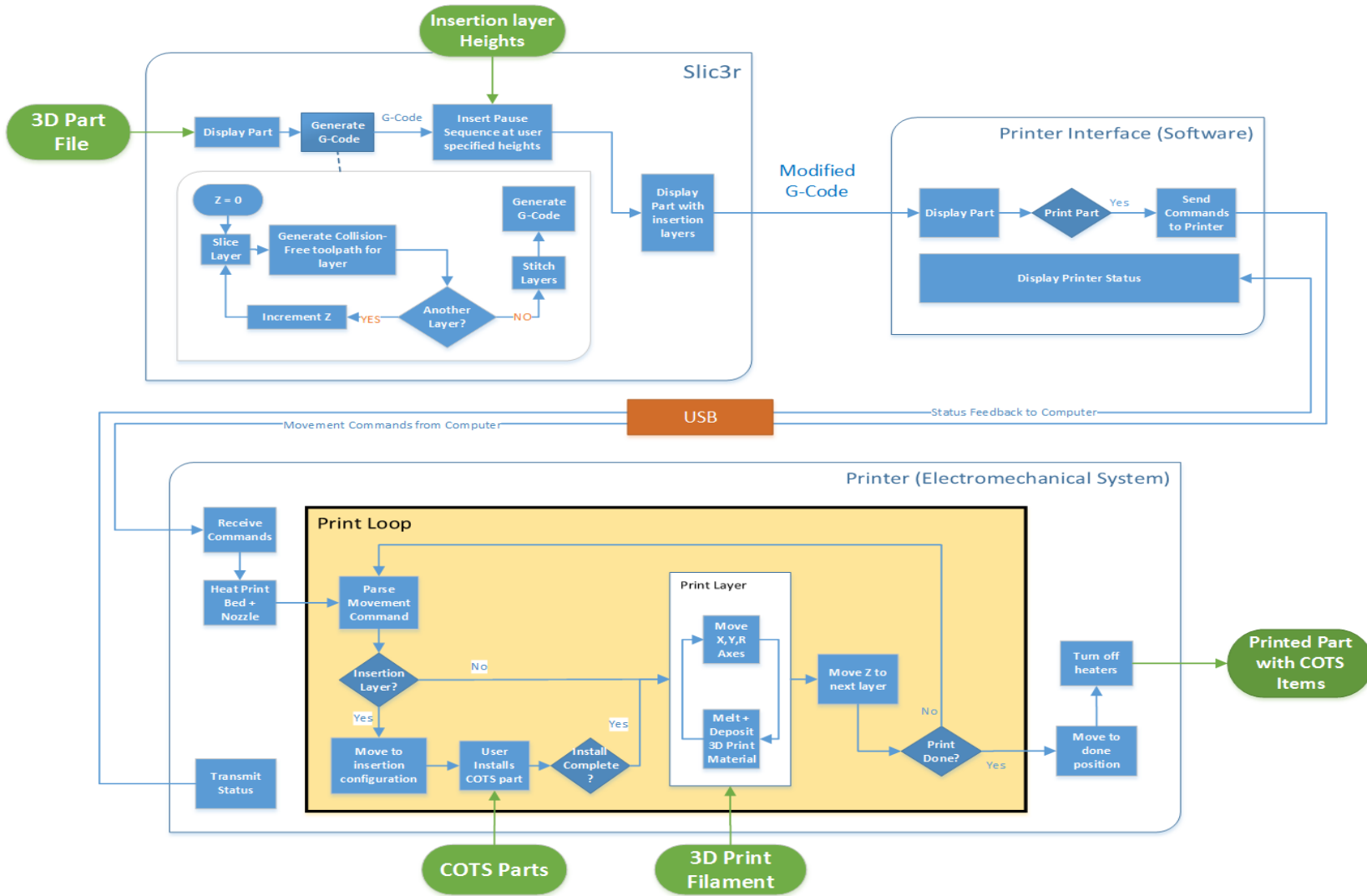


Figure 8: 4DOF functional architecture

## 4.1 Slicing Software

This subsystem is responsible for generating the G-Code that is fed to the printer. G-Code consists of a list of coordinates that define a path the printer nozzle must navigate while depositing the print material. A collision avoidance algorithm will be put in place so as to generate a path around the obstructing COTS items.

The following is an example of a G-Code command. It includes a positioning command ‘G00’ followed by coordinate locations for the XYZ axes.

```
G0 X0 Y0 Z0.25
```

This subsystem receives an STL file of the 3D part to be printed. It also prompts the user to enter the layer heights at which the printer must pause to enable insertion of the COTS item. The part is then plotted in three dimensions with insertion layers shown. The program then proceeds to slice the model layer by layer iteratively from bottom up, generating a layer wise collision free path for the nozzle. We then traverse through all the layer heights and locate those closest to the entered insertion height. A pause sequence is then inserted at this location which will cause the nozzle to go into insertion configuration. In this configuration, the bed is rolled out completely and the hot nozzle is sent to the back of the printer to avoid burning the operator during insertion. Once the pause sequences have been added, the G-code for the part is ready to be sent to the printer interface.

The program then waits for the user to insert the COTS component and hit ‘resume’, after which the printer will continue printing.

## 4.2 Printer Interface Software

The printer interface loads the G-Code generated by the slicing software and displays the 3D sliced model on the computer screen. It communicates with the microcontroller mounted on the printer over USB. The microcontroller receives instructions to initiate the print when the user hits ‘Print’. While printing the interface streams G-Code commands to the printer, and the status is transmitted back from the printer to the software. The print status is displayed on the screen, giving the user real time feedback on the print progress.

## 4.3 Printer (Electromechanical System)

The printer is an electromechanical system that is responsible for physically producing the required 3D assembly. As soon as the microcontroller on board receives commands to initiate the print, it first begins to heat the bed and nozzle head to temperatures defined in the G-Code. The printer then begins printing the base structure layer by layer. If and when it encounters an insertion layer, the hot nozzle moves to the safe configuration and alerts the user to place the COTS item in its feature. The user then inserts the item and hits ‘resume’ which causes the nozzle to go back to its printing temperature and continue printing. Before it does so, it is programmed to deposit some filament in a corner of the print bed and clean its nozzle by dragging the tip along the bed. This ensures a smooth continuation of the print. Aided by the bent nozzle, extra degree of freedom and collision avoidance algorithm, the nozzle is able to smoothly enclose the added part in print material. Once the print had been completed, the heaters are turned off and the bed is rolled out to display the finished 3D part.

## 5. Cyberphysical Architecture

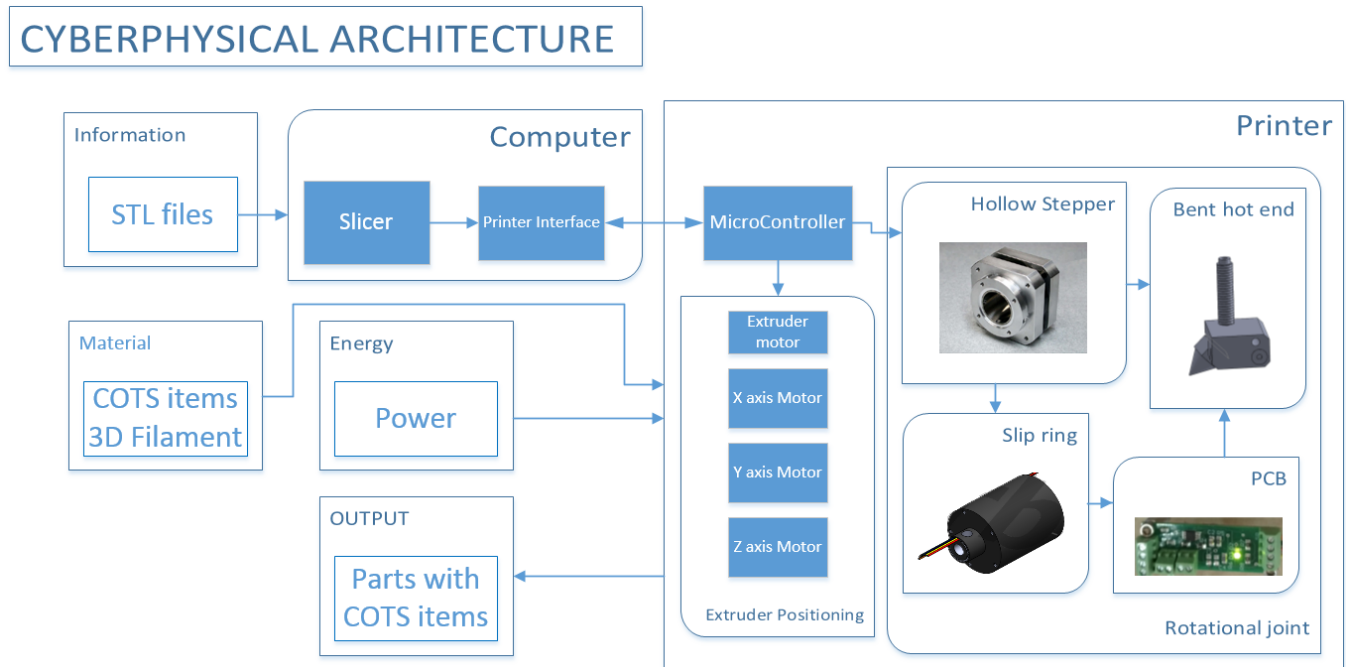


Figure 9: Cyberphysical Architecture

The cyberphysical architecture of the 4DOF system, shown in Figure 9, is inspired by the functional architecture and standard 3D printers. It comprises a software component, namely the slicer and printer interface, and hardware component which is the 4DOF printer.

The 4DOF system is based on the MakerGear M2 commercial 3D printer that was provided by our sponsor and already contains many of the necessary system components. The remaining portions of the cyberphysical architecture which are being developed by the ADD\_IN team are:

1. **Slicer software:** An open source 3D printer slicing software will be extended to include instructions necessary for an additional axis to avoid collision with COTS items.
2. **The printer controller:** Embedded system that must be modified to enable control of all degrees of freedom.
3. **R axis joint:** To avoid collisions while printing around the perimeter of COTS items the extruder nozzle must be infinitely rotatable. To enable infinite rotation a slip ring transfers heater power and temperature signals across the joint.
4. **Angled Nozzle:** A custom nozzle capable of printing along the surface of the COTS item
5. **Thermistor Digitizing PCB:** For the printer controller to reliably control the nozzle temperature a custom PCB will digitize the nozzle temperature signal prior to passing it through the slip ring.

These components are explained in more details in the subsystems section of this report.

## 5.1 Description

Figure 10 provides a graphical illustration of the 4DOF printer. The components on the left describes how the nozzle and rotation joint will be physically mounted on the 3D printer.

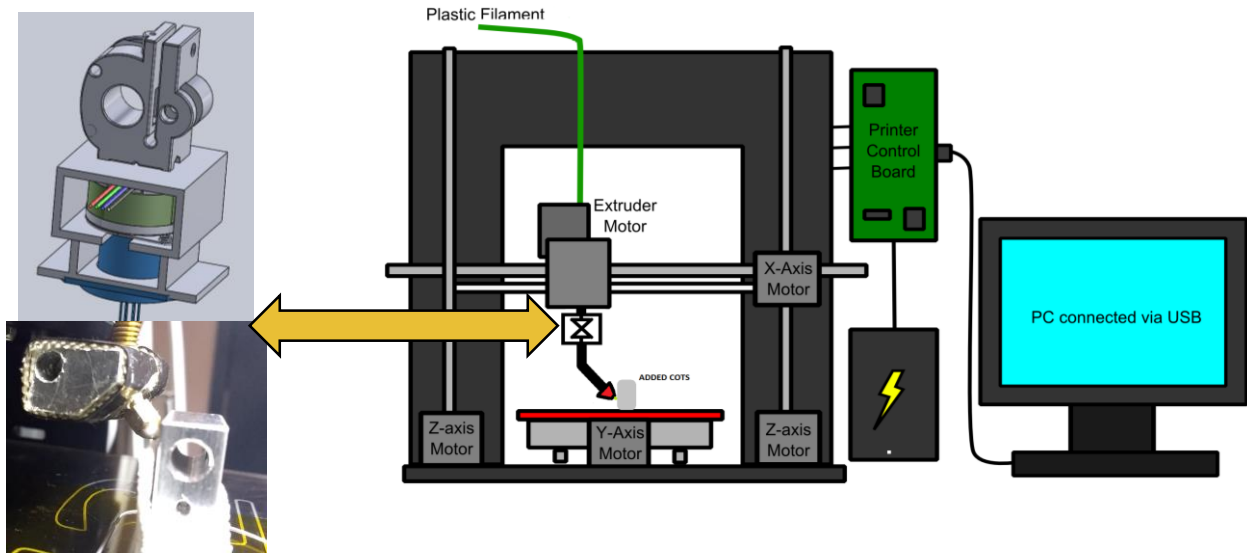


Figure 10: Graphical Depiction of 4DOF cyberphysical architecture

Many of the components in the cyberphysical architecture are from the original unmodified 3D printer. The main modification is highlighted in Figure 10 where it is shown how the hollow shaft stepper motor and the slip ring will be mounted on a custom mounting bracket. The upper part of the mount is the standard filament drive of the MakerGear M2 3D printer, which we extended to hold the stepper and slip ring. As with the original filament drive, it will be 3D printed in one piece.

## 6. Current System Status

### 6.1 Subsystem Requirements

By the spring validation experiment all requirements listed in section 0 will be met. During the fall semester the ADD\_IN team targeted and developed subsystems to achieve the requirements listed in Table 4

Table 4: Requirements Targeted in Fall Semester

ID	Type	Requirement	Nozzle	Insertion Layer	PCB
M.F.1	Functional	Receive standard 3D part files	N	Y	N
M.F.2	Functional	Prompt user for insertion layer	N	Y	N
M.F.4	Functional	Print layers of material	Y	Y	N
M.F.5	Functional	Print locating features for COTS items	Y	N	N
M.F.6	Functional	Go to safe configuration during insertion	N	Y	N



M.F.7	Functional	Enclose COTS item within print material	Y	N	N
M.F.10	Functional	Avoid Kinks in Filament	Y	N	N
M.N.3	Non - Functional	Provide user feedback during printing	N	Y	N

To achieve these requirements, ADD\_IN first identified the subsystems they belong to and started working towards the targeted subsystems. Since, there is no firmware associated requirement we did not focus on the firmware system. The first 5 requirements listed in **Error! Reference source not found.** belong to the insertion layer subsystem. This was developed by invoking slic3r from MATLAB and slowly progress was made through the semester to achieve the corresponding goals. M.F.7 was a part of FVE Encore and we worked on it by designing a part that can be enclosed at least from one side. This was done by printing a rectangular structure against one face of a section of aluminum extrusion. Kinks in the filament were avoided by bending the nozzle where the filament is in molten state and also by conducting quick experiments by 3D printed couplers and manual rotation. In the FVE we demonstrated successful filament extrusion without inconsistencies. Accurate temperature has been maintained by incorporating fiberglass insulation into the hot end end designing a PCB which can digitize the data of the thermistor and send it over the slip ring, thus making it immune to noise.

## 6.2 Nozzle Subsystem

### Description

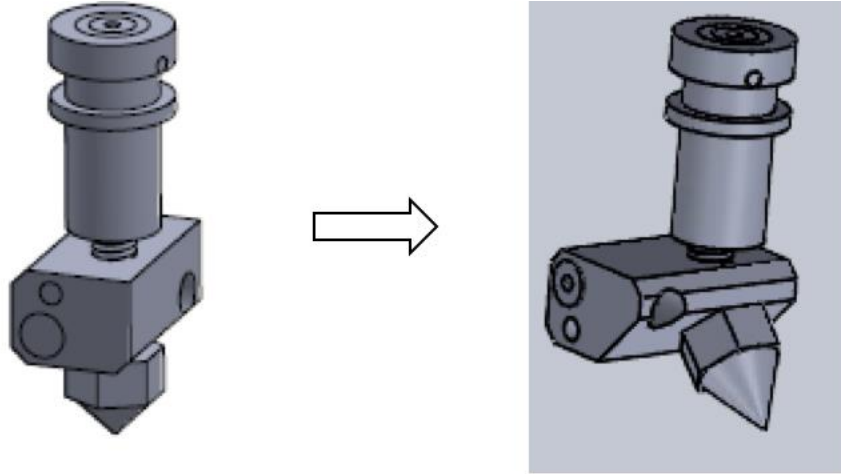
The original nozzle on the MakerGear M2 can only print parallel to the print bed. To print along the surface of COTS items a custom nozzle which can extrude filament at an angle is required. The nozzle subsystem describes the entire assembly which is required to melt and deposit filament including a mounting interface, heater, heat block, thermal insulator, thermistor, and small diameter brass nozzle.



Figure 11: Original MakerGear M2 Nozzle

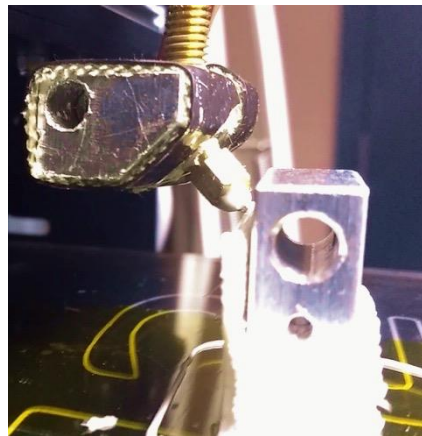
### Status

A custom nozzle was designed and manufactured to meet the system requirements. Multiple iterations and adjustments were performed to achieve the correct thermal and mechanical properties to reliably deposit print material. Operation of the nozzle was demonstrated in the FVE, and thus the subsystem is currently on schedule. A representation of the original and modified nozzle system is shown in Figure 12



*Figure 12: Changes from original 3D print nozzle to custom 4DOF nozzle*

The modified nozzle features a forty-five degree bend, thus giving it the clearance to print on surfaces that are orthogonal to the print bed. Figure 13 below shows the actual nozzle that was machined as per the nozzle design.



*Figure 13: Custom 4DOF nozzle*

### Analysis

The nozzle was tested extensively to determine the correct operating parameters (temperature, print speed, nozzle diameter, nozzle angle, fan setting) that produces the best print quality. It was shown that when printing in certain directions the nozzle tends to ‘smudge’ the extrusion filament producing non-uniform parts. This was anticipated and should not affect the performance of the

completed 4DOF printer since the rotation joint can be positioned to always print in an optimal direction. The smudging can be seen in Figure 14.

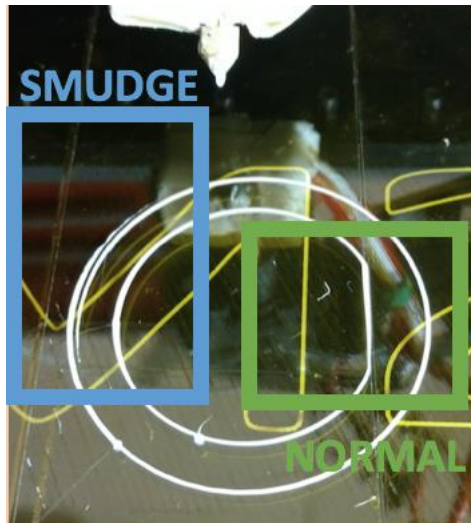


Figure 14: Nozzle Smudging

#### FVE Evaluation

For the FVE, the following tests were conducted to verify the nozzle's performance:

##### Test 1: Verification of nozzle extrusion consistency

To test the nozzle's capability to extrude consistent filament a simple test pattern consisting of parallel lines was developed and printed. The test pattern and the result of the nozzle are shown in Figure 15.

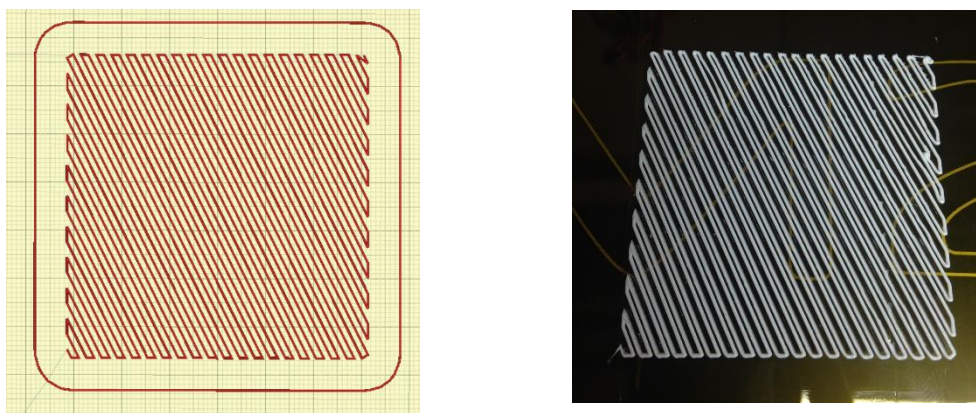


Figure 15: Extrusion test pattern and resulting print

The team was very pleased with the result of the test print. It was observed that the quality of the print was slightly lower in one direction due to smudging, however, this will be rectified by the introduction of the rotational axis.

### Test 2: Printing along the surface of a COTS item

To simulate enclosing a COTS item without requiring a rotational axis the team developed a method to print along the surface of an aluminum extrusion. For this test, the nozzle was allowed to print several layers of a rectangular prism, after which the nozzle went into insertion configuration. At this point, a section of aluminum extrusion was clamped against the rectangular prism and the print was continued. An image taken during this test is shown in Figure 16.



Figure 16: Printing along one surface of a COTS item

This test is the first demonstration of a capability that could not have been achieved using any other standard 3D printer. With the introduction of the 4<sup>th</sup> degree of freedom, this capability will be extended to printing against all surfaces of a COTS item.

### Strengths and Weaknesses

The strengths of the nozzle subsystem include:

1. Consistent filament extrusion along preferred directions
2. Good temperature regulation (with insulation)
3. Modularity (Identical mounting interface as the original nozzle)

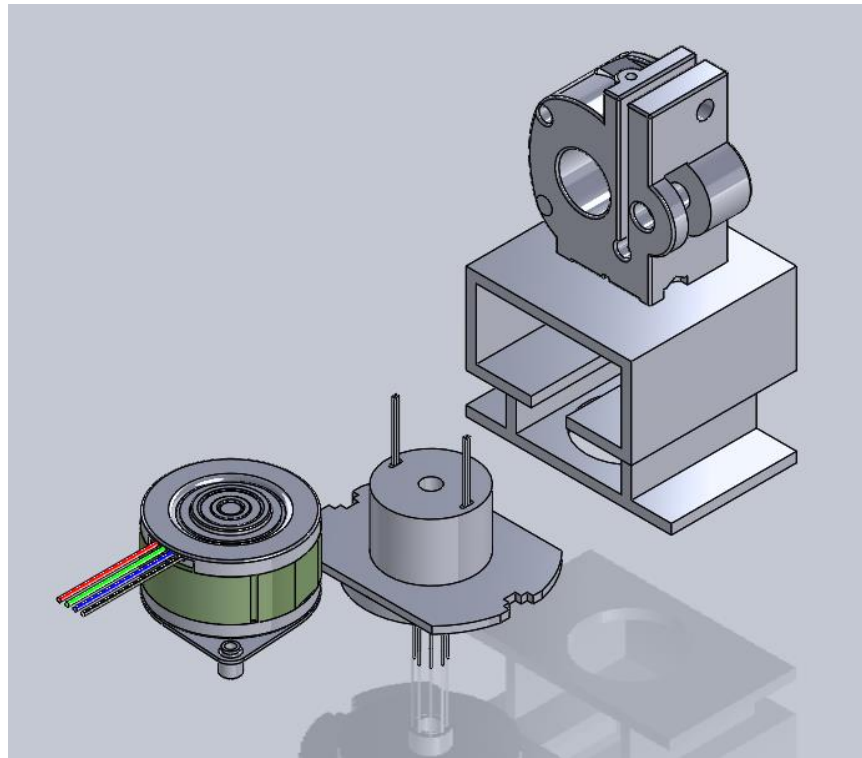
The weaknesses include:

1. Larger body than original nozzle (can be remedied)
2. Smears when printing in certain directions

## 6.3 Rotary Joint Subsystem

### Description

The rotational joint subsystem is the mechanical portion of our system that provides an additional degree of freedom to the 3D printer necessary to avoid collisions with COTS items. The components of this subsystem are the stepper motor, the slip ring and the mount. Figure 17 shows a rendering of these components.



*Figure 17: Rotational joint assembly.*

*From bottom left to top right: Hollow shaft stepper motor, slip ring, custom filament drive*

The main design objectives for this subsystem were to minimize weight, limit backlash, maintain a small form factor (especially the height as it will result in reduced print volume), and to satisfy our performance requirements for speed and precision. These are the main aspects that guided the selection of the components.

### Status

The implementation of the Rotary joint subsystem is planned for spring, so it was not tested during the fall validation experiment. The current status of this subsystem is that the slip ring and stepper motor have been specified and ordered, and are expected to be delivered by mid-January and the mount has been designed.



## Analysis

### *Stepper motor*

Calculations were conducted to determine the required specifications for our motor to satisfy the requirements, and we opted to get the 3709 stepper motor from Lin Engineering custom made to have a hollow shaft. The specifications of importance for our system are as follows:

Repeatability: 1.5 % of a step =  $\pm 0.0135^\circ$

Holding Torque: 0.04 N-m

Weight: 80 grams

Steps per revolution: 400

Hollow shaft ID: 3 mm

Also the stepper is compatible with the A4988 stepper driver that is on our control board.

### *Slip ring*

For the slip ring, we also needed a compact and lightweight component with at least 5 power lines, low added noise and hollow shaft. We selected the 504-0800 from Orbex, which satisfies these requirements. Figure 18 shows the stepper motor and slip ring that have been ordered.



*Figure 18: Stepper motor (hollow shaft not shown) and slip ring*

### *Mount*

The mount was designed as an extension of the filament drive part in the standard Makergear M2 extruder, which will secure the stepper motor and slip ring in a way that all hollow shafts are concentric with the filament drive. The Figure 19 shows the initial filament drive (the grey part), and the mount design, which will be printed as one piece for more rigidity.



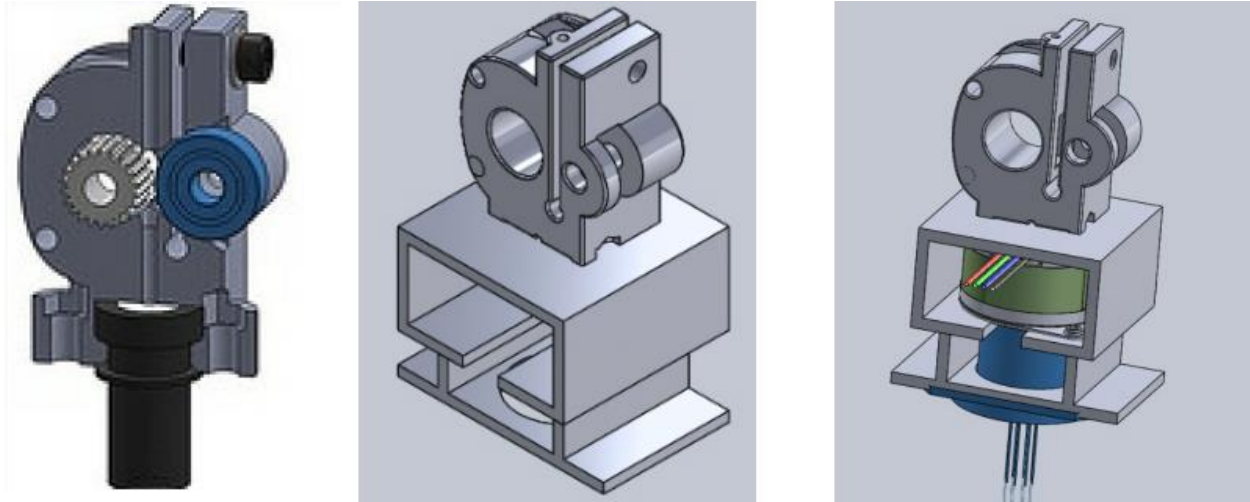


Figure 19: Original (Left) and Custom (Center, Right) Filament Drive

### FVE Evaluation

The joint subsystem will not be fully implemented until spring and thus was not demonstrated within the Fall Validation Experiment.

### Strengths and Weaknesses

The main strengths of the rotary joint subsystem are:

1. High Repeatability - 3 times the stated requirement.
2. Small form factor - / Light weight: Total added weight is less than 150 grams.
3. Simplicity - Having a concentric hollow shaft stepper motor enables a simple implementation, when compared to rotary stage, or standard stepper with gears.

The main weaknesses of the rotary joint subsystem include:

1. Long lead time - The current system relies on a custom component, which introduces the risk on long delays to get a new model in case the initial specifications are incorrect.

This subsystem includes the control of the nozzle and the rotary stage. It is also responsible for the temperature control which includes the PCB design.

## 6.4 Microcontroller Subsystem

### Description

The microcontroller board that powers the Makergear M2 is the RAMBo 1.1B Controller Board. Rambo stands for RepRap Arduino-Mega Board. As shown in Figure 20 the Rambo board is an all-in-one electronics board that communicates with the printer interface software via USB and is used to control all components of the 3D printer.

### Electrical Connections

Conveniently, the RAMBo board can control five stepper motors is typically used only in printers with dual extruders for control the second extruder stepper motor. ADD\_IN will use this port in to control the R joint necessary to rotate the nozzle. To implement the endstops for that axis we will use currently unassigned pins from the expansion port. To implement the temperature control I2C expansion ports will be used to communicate with the ADC on our custom PCB to get the digital reading of the thermistors.

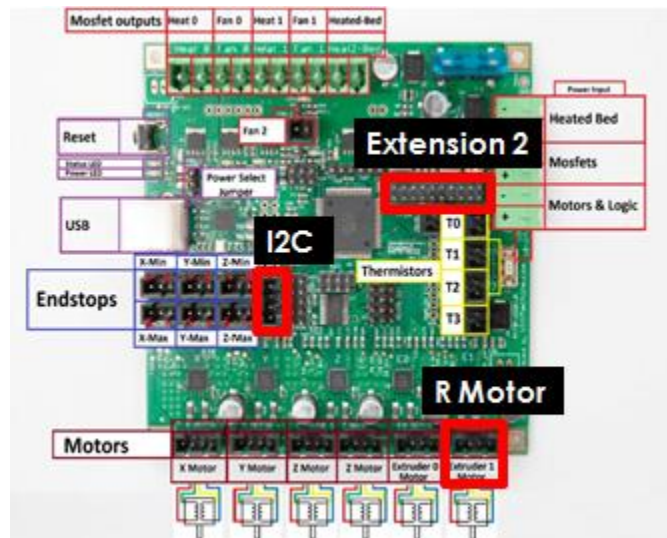


Figure 20: RAMBo 1.1B Control Board

### Firmware

The board on our Makergear 2 is loaded with the Marlin Firmware, which is Arduino based and entirely open-source. The role of the Firmware is to interpret the G-code commands and control the printer's heaters and motions accordingly.

We also successfully modified the code to send control signals from the fifth motor axis thereby confirming the board's ability to control an additional stepper motor. The objectives of this subsystem are to extend the code to control the R axis and enable parsing of G-Code commands for all four axes. Since the G-code commands received by the microcontroller only describe discrete point, line, and circular arc movements, the firmware must use motion planning algorithms to synchronize the movement of all axes.

### PCB

An ADC is used to digitize the thermistors readings before they are sent across the slip ring. This is done to ensure that the data through the slip ring is immune to noise. To achieve this, a PCB was designed with a 4 channel ADC with a PGA such that it can accurately convert the readings of the thermistors. This will be mounted near the nozzle. The parts used are robust to heat changes to avoid any damage due to nozzle heat.

## Status

### *Firmware*

All the files used in the firmware have been mapped to a flow diagram. Only the files that must be modified will be used. Currently, the firmware has been modified with a G-code “G5 Rx” which can control the R-axis stepper motor by “x” specified degrees by the user as shown in Figure 21.



*Figure 21: R-axis stepper motor controlled using G5 command from the pronterface software*

### *PCB*

The PCB has been designed, assembled and interfaced with the RAMBo board. It can successfully communicate with the PCB and the temperature can be measured. The PCB being tested during the FVE is shown in Figure 22.



*Figure 22: Custom PCB and RAMBo board under testing*

## FVE Evaluation

### *Firmware*

The modified firmware has been successfully demonstrated during ILR sessions throughout the semester. Since custom firmware was not required for any of the tests conducted in the FVE there was no firmware component demonstrated.

### *PCB*

The PCB implementation was demonstrated in the FVE. It successfully communicated with the RAMBo board and the temperature readings were shown on the serial monitor displayed on a laptop.

## Strengths and Weaknesses

The strengths of the RAMBo board are as follows:

1. **Motor Synchronization:** The RAMBo board can control up to 5 stepper motors. Thus it is easy to synchronize the X, Y, Z, R and E axes. If another board were used then care must be taken to communicate between the two boards which would result in latencies affecting the printing sequence.
2. **Open source firmware:** The RAMBo uses open source firmware which has enough documentation and thus is easy to modify.
3. **PCB communication:** It is easy to communicate with the PCB using I2C expansion ports on the RAMBo and no excess hardware needs to be installed.

The strengths of the PCB subsystem are as follows:

1. **Small form factor:** The PCB is only 1.5" x 0.5". Its compact design is such that it can be mounted at the nozzle easily.
2. **Minimum wires used for communication:** I2C is used for communication with the PCB and thus the numbers of lines passing through the slip ring have been minimized.
3. **High temperature robustness:** The PCB components are robust to temperature changes because they must be placed near a heated nozzle.

The weakness of the RAMBo board and the PCB is the large overhead required to modify these firmware which has nearly 50 files.

## 6.4 Software Subsystem

The software system is currently on schedule with a demonstrated successful implementation of the layer selection algorithm and initial proof of concept implementation of the path planning algorithm. A summary of these programs, the results achieved, and FVE demonstration is as follows.

### Layer Selection Algorithm

#### *Description*

To install COTS parts into a 3D printed part the printing process must be 'paused' at a certain print layer and the printer put into a configuration that allows an operator to install the COTS part.

Following the COTS part installation, the printer must then resume printing without creating defects in the part.

The layer selection algorithm is invoked as part of the slicing script written by the ADD\_IN team. The process flow for the script is shown as follows.

1. Prompt user for *.stl* file
2. Prompt user to enter insertion layer(s) [defined in millimeters above the print bed] (Figure 24)
3. Display 3D model of part with selected insertion planes [shown in red] (Figure 23)

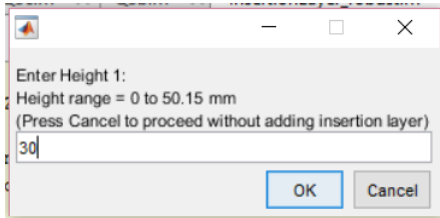


Figure 24 : Insertion Layer Selection GUI

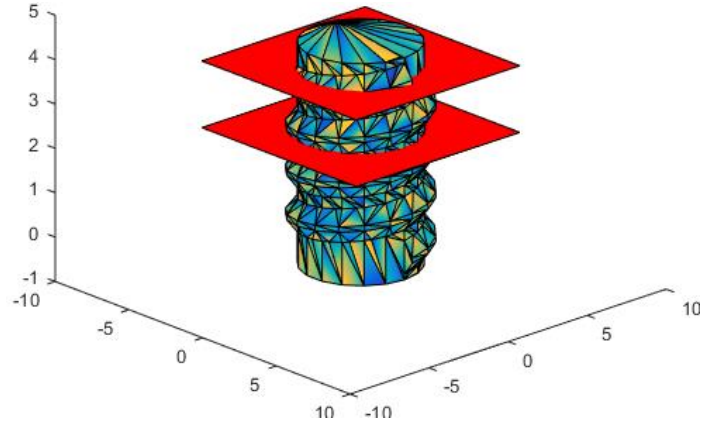


Figure 23 : Rendering of *.stl* file (multicolored) showing the insertion planes (red)

4. Invoke Slic3r (open source 3D printer slicing software) with 4DOF specific configuration data to generate G-Code file for printer control
5. Process G-Code file to insert 'pause sequence' at appropriate location



Figure 25: Printer in insertion configuration

A high level functional overview of this sequence is shown in Figure 26.

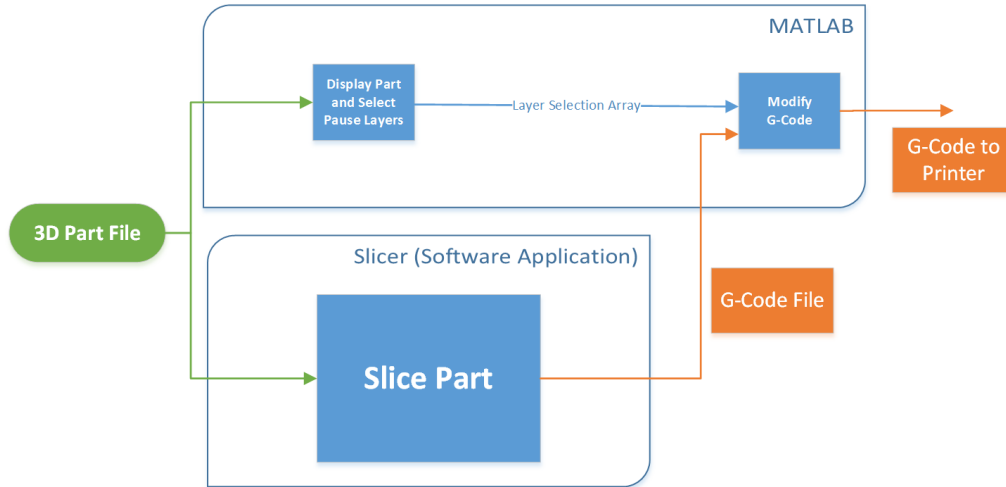


Figure 26: Layer Selection overview

### Status

The layer selection algorithm has been successfully implemented and demonstrated on numerous occasions. Initial testing identified minor bugs (such as robustness to invalid layer entries) which were quickly resolved. Later experiments revealed a consistent issue with maintaining print quality due to a trail of extruded filament being created between the print nozzle and part as the printer enters the insertion configuration (shown in Figure 27). Additional work was required to resolve this issue but it was eventually determined that a combination of fan operation (cooling the filament after it leaves the extrusion nozzle) and a short, rapid acceleration along the X axis when entering the insertion configuration would reliably prevent the creation of filament trails.

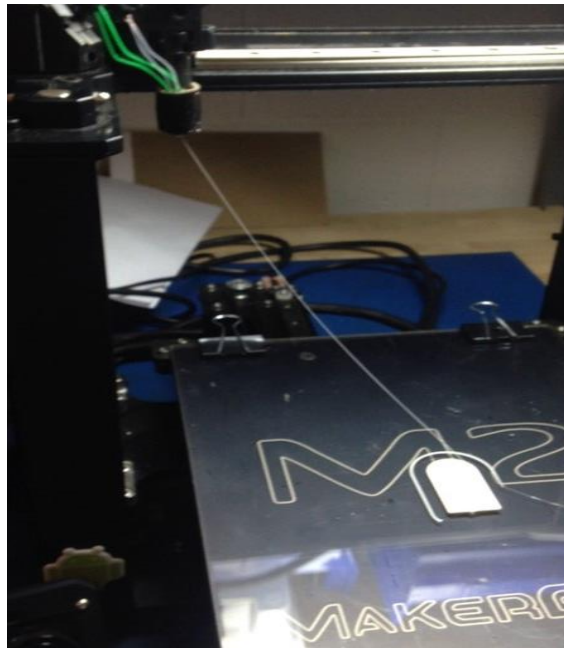


Figure 27: Filament trail created as the printer moves to the insertion configuration



The software team was able to fix the filament trail issue within a week and thus not affect the project schedule.

### *FVE Evaluation*

A specific test was included as part of the FVE to ensure correct operation of the layer selection algorithm. The test consisted of printing a two centimeter cube using the printer's original nozzle with and without an insertion layer. The test case (with insertion layer) was shown to have no noticeable difference in print quality when compared to the control (no insertion layer) part. The experiment was repeated multiple times and captured on video.

### Path Planning Algorithm

#### *Description:*

Risk analysis showed that whether or not a path planning algorithm could be successfully implemented was unknown and could potentially adversely affect our project. Although not scheduled to be implemented until spring semester, as part of a risk mitigation effort ADD\_IN decided to implement a proof of concept of the path planning algorithm during fall.

The basic steps for the algorithm are as follows:

1. Orient the printer nozzle to be normal (or at some specified angle) to the velocity trajectory of the nozzle at all times (this yields two possible solutions for nozzle orientation).
2. Always orient the nozzle to be pointing towards the nearest COTS item (this selects between the two solutions in (1)).

In this design the path planning algorithm simplifies each COTS item to a 'keep out' zone centered at a specific point location and with a specified height. The assumption is made that the G-Code file for the printed part is designed to accurately represent the exterior profile of the COTS item, and thus no other information is needed about the COTS item's external geometry. Not relying on additional geometry information implies that the part designer avoided creating a design which violate the geometry constraints of the printer (i.e. parts are too close together, or have concave profiles that the nozzle cannot access), since these potential collisions will not be detected by the path planning algorithm.

The path planning algorithm was implemented in MATLAB and is applied to a given G-Code file. The inputs to the algorithm are a G-Code file produced by Slic3r which encodes the XYZ locations the print nozzle must move to, and a list of COTS item locations and heights. Using this information, the algorithm computes the necessary angle of the rotation axis for each G-Code movement command, and produces a modified G-Code file. It was discovered that Slic3r only produces straight line commands (G00 and G01) which greatly simplifies the implementation algorithm because rotation along an arc doesn't need to be considered.

The actual implementation of the algorithm is as follows:

#### ***For each G00 or G01 (straight line movement) command:***

1. Compute the vector ( $\vec{n}$ ) in the XY plane from the current position to the center point of each COTS item ( $\vec{c}_i$ ) which is at or below the current Z position

$$\vec{n}_i = \vec{c}_i - \vec{p}_1$$

- Determine the COTS item which is nearest to the current position.

$$\vec{n} = \min_i(\|\vec{n}_i\|)$$

- Compute the vector for the current movement command

$$\vec{m} = \vec{c}_i - \vec{p}_2, \text{ where,}$$

$\vec{p}_2$  is the desired (x,y) position of the extruder at the end of the movement command

- Determine which side of the print path the COTS item is on

$$side = \det([\vec{m}, \vec{n}]) \quad [1]$$

where for

$side > 0$ : COTS item is to the left of  $\vec{m}$

$side = 0$ : COTS item is on  $\vec{m}$

$side < 0$ : COTS item is to the right of  $\vec{m}$

- Compute the velocity of the nozzle in the coordinates of the print bed

$$\vec{v} = \frac{\vec{p}_2 - \vec{p}_1}{\|\vec{p}_2\| \|\vec{p}_1\|}$$

- Compute the angle between the velocity vector and the x-axis

$$\alpha = \text{atan2}(v_y, v_x), \text{ where } v_y, v_x \text{ are the } x, y \text{ components of } \vec{v}$$

- Based on which side of  $\vec{m}$  the COTS item is, rotate the nozzle to be normal to the velocity vector and pointing towards the COTS item.

$$side > 0: r = \alpha + \frac{\pi}{2}$$

$side = 0$ : Invalid – Collision will occur

$$side < 0: r = \alpha - \frac{\pi}{2}$$

### Status

To evaluate the path planning algorithm MATLAB scripts were written both to implement the algorithm and to plot the resulting G-Code file. A simple part (cylindrical tube) was designed and sliced. The algorithm was then applied on the resulting G-Code file with a COTS item positioned in the center of the cylinder and the result was visualized as shown in Figure 30 and Figure 29. An example of the modified G-Code is shown in Figure 28 : G-Code modified to include additional 'R' axis commands

```
g1 x119.200 y133.151 e1238.61271 r18.4349
g1 x118.973 y133.665 e1238.63294 r90
```

Figure 28 : G-Code modified to include additional 'R' axis commands

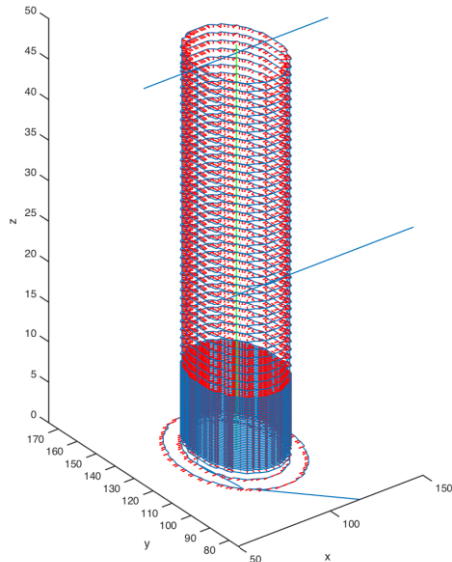


Figure 30 : Plot of modified G-Code including R-axis commands. A COTS item (represented by the green line) was specified to be at the center of the cylinder.

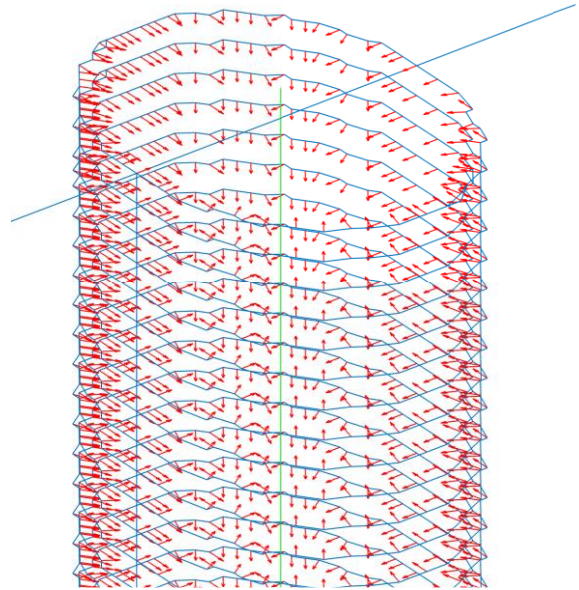


Figure 29 : Close up of G-Code path. Notice the nozzle is always oriented normal to the path and towards the COTS item.

The results clearly indicate that the algorithm is successfully controlling the nozzle orientation to avoid collision with the COTS item. This proof of concept validates the path planning algorithm, but requires significant improvements in spring semester before the printer can reliably avoid collisions. The required improvements are as follows.

1. Incorporate 'last position awareness' into path planning algorithm. The algorithm currently processes each line of G-Code individually with no consideration of the previous position of the nozzle. This can lead to collisions for certain geometries where both the new position and direction of rotation from the previous position are critical.
2. Incorporate error checking to ensure that the part geometry is physically printable based on the limitations of the printer. For example, COTS items can not be so close together that the nozzle can not fit between them.
3. Special geometries such as large aspect ratio rectangles may require special consideration to operate time efficiently (print time is only specified as a stretch requirement and thus this is non-mandatory).

### FVE Evaluation

As previously described, the path planning algorithm was implemented purely as a proof of concept for risk mitigation, and thus was not included as part of the FVE. However, the algorithm had been successfully demonstrated to the TA's during a progress review presentation.

## 7. Project Management

### 7.1 Work Breakdown Structure

ADD\_IN's work breakdown structure is shown in Figure 31.

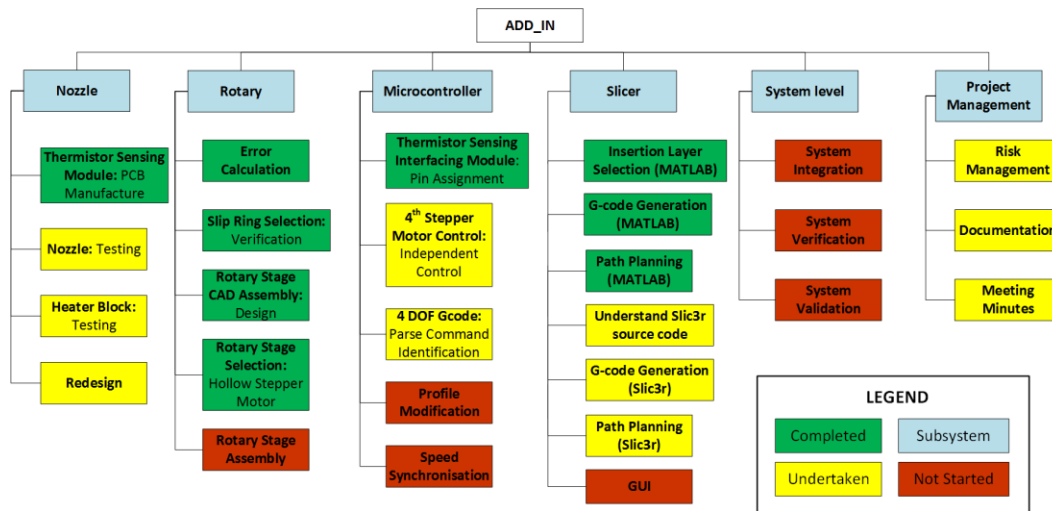


Figure 31: Work Breakdown Structure

The WBS consists of three tiers:

- Tier 1 – Project name
- Tier 2 – Major Subsystems
- Tier 3 - Subsystem level tasks.

#### Nozzle Subsystem

We plan to have several nozzle designs over the year. Currently, we have finished the first nozzle iteration which was demonstrated during our FVE. Further iterations will follow in the spring semester which is why this remains to be an ongoing task. Meanwhile, accurate reading of thermistor data was achieved using our custom PCB which was used to digitize the output. This will also ensure that our system remains immune to noise generated by the slip ring.

#### Rotary Subsystem

After arriving at the necessary specifications and requirements, the selection of the hollow stepper motor and slip ring has been completed. The parts have been ordered and are expected to arrive by the beginning of spring semester. A mount has been designed to position these parts. They will be assembled early next semester.

#### Microcontroller Subsystem

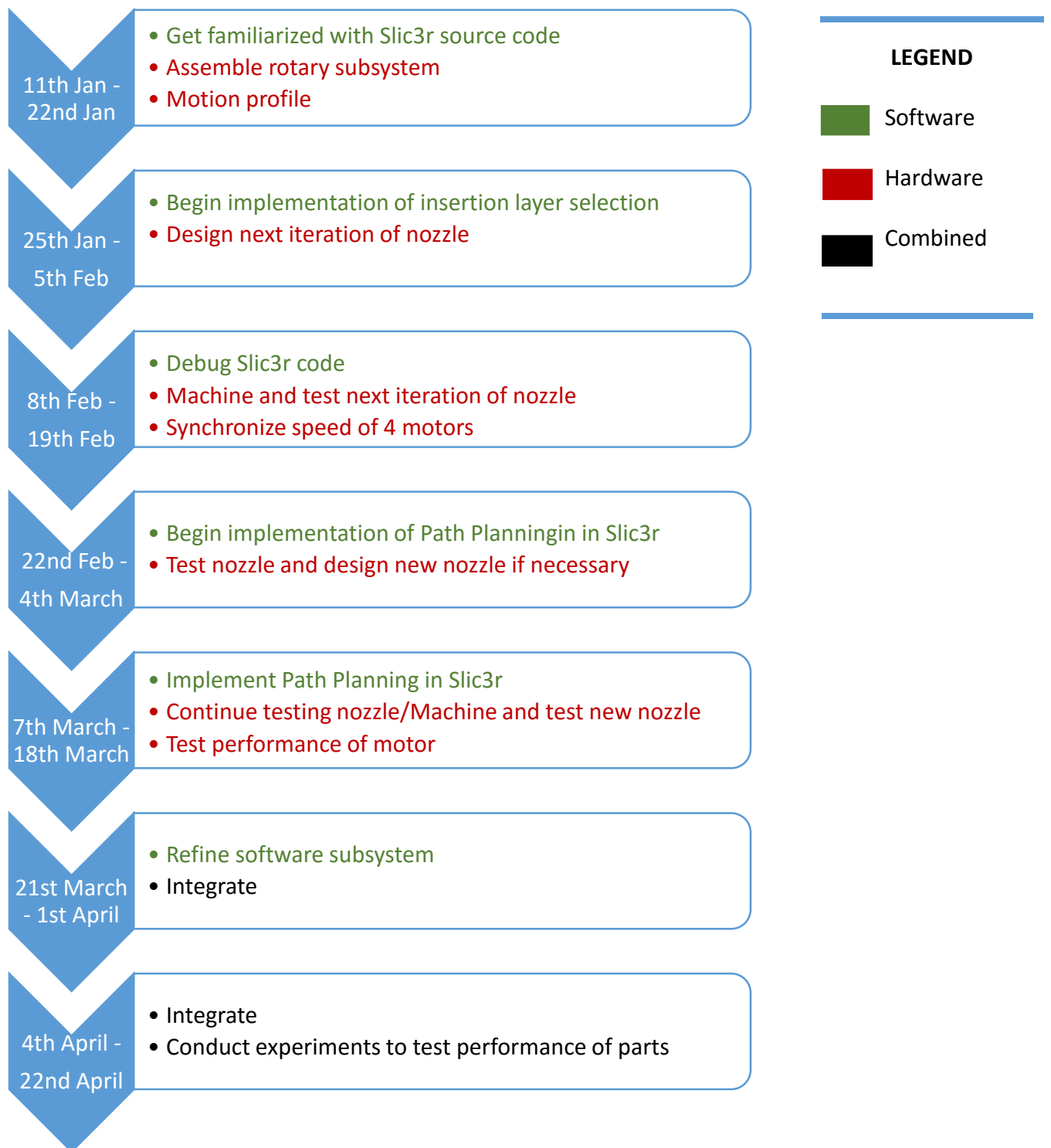
We have started work on controlling the 4<sup>th</sup> stepper motor using G-Code, however there still remains a lot to be done in this subsystem.

#### Slicer Subsystem

Insertion layer selection and 4 DOF G-Code generation has been achieved using MATLAB, however the same remains to be implemented using the Slic3r software. We have also implemented path planning in MATLAB to serve as a proof of concept for future work.

## 7.2 Project Schedule

### *Biweekly tasks for spring*



## Milestones

We are currently on schedule since we were able to complete all our fall semester tasks in a timely manner. The major milestones for spring are summarized in Table 5.

Table 5 : Spring Milestones

Date	Milestone
19 <sup>th</sup> February	Insertion Layer implementation
18 <sup>th</sup> March	Path Planning and 4DOF G-Code generation
18 <sup>th</sup> March	Finalize nozzle design
15 <sup>th</sup> April	Verification and validation of printed parts with COTS items

## 7.3 Spring Semester Test plan

Table 6 highlights the capability milestones for each progress review in the spring semester.

Table 6 : Spring Semester Milestones

Milestones	Date	Capability
PR 7	Late January	Install Slicer from source. Mount hollow stepper and slip ring on the extruder
PR 8	Mid-February	Test next iteration of nozzle
PR 9	Late February	Implement insertion layer selection on Slicer. Synchronously control the 4 degrees of freedom
PR 10	Mid-March	Implement path planning
PR 11	Early April	Drive motors using 4DOF G-code
PR 12	Mid-April	Print parts with enclosed COTS items

## Spring Validation Experiments

The team's spring validation experiments will be composed of 3 parts, first we will test the functionality of the software. Second we will test the 4DOF printer. And finally we will verify that the integration of the system gives the results it was designed for, i.e., printed parts with well incorporated COTS Items that satisfy our performance requirements.

### Objective

To verify the following:

1. The ability of the software to generate four axis G-Code commands with obstacle avoidance and insertion layer selection capability
2. The ability of the printer to correctly print by executing those commands.
3. To validate that the parts printed with ADD\_IN have superior qualities than standard parts.

### Location

NSH B506 – Team F Test Bench



## Equipment

1. Laptop computer equipped with the following:
  - a. Slicer software
  - b. 3D visualization software
  - c. Printer interface software
  - d. *.stl* files of
    - P1: Part with stiffener rod
    - P2: Part with threaded insert
    - P3: Part with sensor/electronics
2. Reference 3D printed parts:
  - a. R1: Similar to P1 but without stiffener rod
  - b. R2: Similar to P2 with glued screw insert
3. ADD\_IN printer configured with the following:
  - a. Custom nozzle
  - b. PLA filament roll
4. COTS items
5. Test bench
6. Test setup for each printed part
  - a. Bending stiffness tester
  - b. Pull out strength tester
  - c. Electronics operation tester
  - d.

## Test Procedure:

For parts P1-P3, perform the following:

1. Use our Slicer software to load the *.stl* file
2. Select insertion layer and print configuration
3. Generate the *.gcode* file
4. Visualize resulting G-Code and ensure accuracy
5. Send *.gcode* file to printer
6. Start print
7. Insert COTS item when printer pauses at the insertion layer
8. Resume and finish printing
9. Conduct the following tests
  - a. Bending stiffness test for P1 and R1
  - b. Pullout strength test for P2 and R2
  - c. Electronics/sensor functionality check for P3

## Success criteria

1. Generated G-code visualization should show the rotation axis being positioned to avoid collisions with the COTS item
2. Printer should successfully finish the print
3. Stiffness of P1 > Stiffness of R1
4. Pullout strength of P2 > Pullout strength of R2
5. Electronics/sensor in P3 should be functional

## 7.4 Budget

The team is currently considering buying a new 3D printer in spring in order to speed up the testing process. That would add another estimated \$2000 to our budget. Table 7 below shows the complete list of parts for our project, with the big-ticket items being the top 3 at the list and the optional 3D printer.

Table 7 : Project Budget

Item	Quantity	Cost per unit	Total Price	Details
Hollow shaft stepper motor	1	\$169	\$169	Custom made motor from Linengineering
Slip ring	1	\$167	\$167	504-0800 Hollow shaft slip ring from Orbex.
Rambo backup board	1	\$180	\$180	Rambo v1.3 board
Makerfaire tickets	4	\$20	\$80	Tickets for the Makerfaire to do market research
Backup extruder motor	1	\$80	\$80	Kysan geared stepper motor with drive gear
V3b hot end kit	1	\$65	\$65	V3b hot end for testing and backup
Hot end parts	N/A	\$135	\$135	Various parts for machining the hot end and backups: Barrels, nozzles, heat blocks, thermistors, heaters and aluminum bars.
Flexible filament	1	\$50	\$50	Ninja Flex filament for testing
Extruder wiring harness	1	\$30	\$30	Wiring harness for the added motor
Print bed	1	\$24	\$24	Backup glass print surface
COTS items	N/A	\$20	\$20	Standard COTS items from McMaster
Estimated shipping	N/A	N/A	\$50	Overall estimated shipping costs
<b>Total Spent in fall</b>			<b>\$1050</b>	91% of the expected total budget
Spring additional spares and filament	N/A	N/A	\$100	Expected spare parts, specific cots items and testing filament
<b>Total budget</b>			<b>\$1150</b>	
Second 3D printer (Optional)	1	\$2000	\$2000	Second 3D printer to speed up the testing process
<b>Total budget with 2nd 3D printer</b>			<b>\$2900</b>	

Received
  Ordered
  Not ordered
  Big-ticket items

## 7.5 Risk Management

Recognizing and mitigating risks is essential to avoid unexpected failures, changes, and delays to the project. The ADD\_IN team has adopted a systematic method for tracking and managing risks.

### Risk Management Process

#### *Risk Identification*

Risks of all magnitudes can be identified by all team members at all times. Tracking additional risks incurs almost no burden and in many cases risks which were initially considered to be minor can grow to become significant. Risk brainstorming sessions are occasionally held, but most risks are identified on a day-to-day basis while working on developing the system.

#### *Risk Level Assessment*

To consistently assess the magnitude of a risk, we developed rubrics that define a scale for the likelihood and severity of each risk. These rubrics are depicted in **Error! Reference source not found.** and Table 9.

*Table 8 : Risk Likelihood Legend*

<b>Legend - Likelihood</b>	
1	Highly Unlikely - Almost impossible
2	Unlikely but possible to occur
3	Likely to occur sometime during the life of the project
4	Expected to occur during the life of the project
5	Estimated to occur

*Table 9 : Risk Severity Legend*

<b>Legend-Severity</b>	
1	Inconsequential hardware loss, Time delay < 1 day
2	Minor Hardware loss. Time delay ~1 day
3	Hardware loss. Time delay 1-5 days. Inability to meet non-critical requirement
4	Inability to meet critical requirement (significant reduction in printer's capability)
5	Failure of project/All major requirements

After assigning an appropriate likelihood and severity level to each risk, the risk is added to a risk tracking excel template. The template calculates and ranks all risks according to their total risk level (product of likelihood and severity). The results are broadly classified into low, medium, and high risks based on **Error! Reference source not found.** Mitigation strategies are then immediately implemented for all high and medium level risks.

*Table 10 : Total Risk Level Definitions*

<b>Legend - Total</b>	
0-6	Low Risk
6-12	Medium Risk
12+	High Risk

### Current 4DOF Risks

The ADD\_IN team is currently tracking twenty risks, ranging in total risk level from two through six. The top four risks are shown in Table 11 and indicated on a risk matrix in Figure 32.

Table 11 : Top 4 risks being tracked

ID	Type	Description	Likelihood	Severity	Total	Owner
2	HW	Rotary joint not sufficiently precise	2	3	6	Ihsane
13	HW	Unforeseen challenges in rotary joint mounting	3	2	6	Nikhil
20	HW	Custom stepper motor wrong specifications (Est lead time ~6 weeks)	2	3	6	Dan
5	SW	Slic3r cannot be modified	2	3	6	Astha

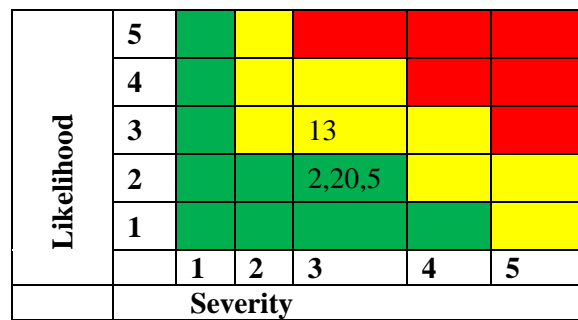


Figure 32: Risk Matrix

### Changes since PDR

Significant risk mitigation has been performed since the PDR. The top four risks at the time of the PDR (5/11/15) and the mitigation strategies implemented are shown in Table 12.

Table 12 : Risk mitigations from PDR to CDRR

ID	Type	Description	Mitigation	Likelihood Change	Severity Change	Total Change
1	HW	Non-uniform extrusion from nozzle	Construction + Testing of custom nozzle has demonstrated uniform extrusion	2->1	4->4	8->4
2	HW	Rotary joint not sufficiently precise	Numerical calculations based on final nozzle design have shown that the selected hollow shaft stepper provides the precision specified in performance requirement #600.	3->2	3->3	9->6
3	HW	Print quality to reduce by additional weight on X axis	Test conducted with additional weight added to X axis did not detectably affect print quality	4->1	2->2	8->2
4*	SW	Challenges in path planning algorithm implementation	Proof of concept implantation of path planning algorithm has demonstrated path planning feasibility, and also identified potential challenges in implementation	2->3	3->1	6->3

\*Note that for risk #4 the likelihood has increased. The mitigation strategy (implementing a proof of concept version) identified new challenges in the algorithm implementation, but the cumulative severity of all challenges identified is low and deemed surmountable.

## 8. Conclusions

### 8.1 Key Lessons Learned

There are several lessons we learned this semester, some related to our project and some about team work. The key lessons learned include:

1. *Start testing early next semester*  
Printing is a slow process and requires many runs in order to arrive at the correct parameter settings. We learned this lesson when we were very close to the FVE and as a consequence, had to spend several late nights in the lab waiting for prints to complete. So next semester, we plan to start printing parts early.
2. *The printer is fragile so treat it with care*  
Over the course of the semester, we ended up breaking several parts of the printer and came extremely close to breaking many others. Since we have only one printer, it would be wise to treat it with care.
3. *Stock up on spare parts*  
Having spare parts ready saved us a lot of time this semester.
4. *Ensure that all team members have the same version of software*  
Software compatibility caused some problems this semester. At one point of time, all the team members were using different versions of Slic3r. This caused quite a bit of confusion until we all got back on the same page.

### 8.2 Key Activities for Spring Semester

The following are the key activities that will be taken up in spring:

1. Purchase a new printer: As mentioned in the above sections we plan to purchase a new printer so that the testing time may be accelerated.
2. Rotary Stage Assembly: The rotary stage must be assembled and its integration must be tested.
3. Path Planning Algorithm: The entire path planning algorithm must be completed such that it avoids the COTS item.
4. System Integration: All the subsystems must be integrated and tested

## 9. References

- [1] Slic3r software: <http://slic3r.org/>
- [2] Makergear M2 3D printer: <http://www.makergear.com/products/m-series-3d-printers>
- [3] Cura software: <https://ultimaker.com/en/products/cura-software>
- [4] Tang, Tran Duc, "Algorithms for collision detection and avoidance for five-axis NC machining: A state of the art review" J. Computer Aided-Design pp.1-16, 2014.
- [5] Rambo Development Board: [http://reprap.org/wiki/Rambo\\_v1.1](http://reprap.org/wiki/Rambo_v1.1)