School of Mechanical and Aeronautical Engineering

Diploma in Mechanical Engineering

Sensor integration into industrial metal 3D printing and powder spreader equipment

Final Report

MAE-22705

TOH EN RUI CLEMENT 2034135

Year of 2023

FYP Supervisor: Dr. Chew Choon Seng

Co-examiner: Mr. Cheung Kim Kwong

Academic Year 2022/3

Acknowledgements

I would like to express my sincere gratitude to Dr. Zheng Jie Tan, my company supervisor in the Additive Manufacturing Industrialisation department in the Advanced Remanufacturing and Technology Centre (ARTC). He has provided unwavering support throughout my 22-weeks internship equivalent - FYP project. His guidance and mentorship have been invaluable in the diverse mechanical, materials, software, and system engineering fields. His expertise and experience have been instrumental in the success of my projects. I am grateful for the opportunity to have performed under his supervision and honoured to be part of a dynamic and innovative organization.

Abstract

Integrating sensors into metal 3D printing and powder spreading equipment can significantly improve the efficiency and precision of these processes. By incorporating sensors into the equipment, real-time monitoring of critical parameters such as temperature, pressure, and material flow can be achieved. This allows real-time adjustments to be made to the equipment, resulting in improved accuracy and reduced waste. Additionally, the use of sensors can also provide valuable data for process optimization and quality control. This paper presents the design and implementation of a sensor integration system for a metal 3D printer and powder spreader equipment. The 4 key projects are the Feasibility Study of Integrating Sensors to 3D Printer for Conditional Monitoring, M5Stack FIRE IoT Development, Webcam Streaming in EOS M 290 Chamber, and Spectrometer Integration and Development.

Table of Contents

List of Figures

List of Tables

1. Introduction

Sensor integration into industrial metal 3D printing and powder spreader equipment

1.1 Background

Additive Manufacturing (AM) processes, such as Laser powder bed fusion (LPBF), have grown in popularity in recent years. The metal additive process provided quality production parts for the medical and aerospace industries [1,3]. The quality of parts generated by the metal additive process has improved dramatically but has reached a limit using the trial and error
approach. One challenge in AM is the lack of real-time feedback on the fabrication process and the quality of the part being made. This is especially critical given the relatively long periods of time that complex parts can require to be constructed [2]. Hidden flaws in a part, which excessive residual stresses can cause, can result in the final part being unusable and wasting valuable time, resources, and AM machine life.

Real-time monitoring also allows for possible feedback control to compensate for or correct detected errors or abort the continued construction of defective parts. In situ measurements will add to the knowledge about the physics behind additive processes and expedite process development that leads to improved material and structural properties [4]. This study aims to address this challenge by proposing the integration of sensors into industrial metal 3D printing and powder spreader equipment for real-time documentation and data acquisition to optimise prints.

1.2 Project Objectives and Scope

The proposed system involves the implementation of real-time and accessible documentation of the print job by monitoring the powder bed using an integrated webcam and various sensors. These sensors live-stream the build area to show any irregularities, such as an incomplete powder layer, grooves or cavities, revealing any uneven or incomplete layering steps. If any problems are detected, the build process can quickly be stopped. This sensor integration can be a beneficial tool for many LPBF implementations by providing real-time acquisition data of the print job. The study presents a custom-designed sensor integration system and its potential impact on improving the build process.

2. Feasibility Study of Integrating Sensors to 3D Printer for Conditional Monitoring

Additive Manufacturing (AM) recently gained significant traction, particularly in prototyping. Using AM techniques reduces the time and cost associated with designing and developing tools and fixtures. One of the most promising AM technologies in industrial applications is Laser Powder Bed Fusion (LPBF), which boasts low post-processing requirements, high accuracy, and exceptional component quality. However, given the long print times associated with LPBF, a single defect can result in significant economic losses. Therefore, developing technologies for process monitoring and self-regulation of print jobs are crucial for improving LPBF and have been identified as a critical area of research focus to make LPBF more sustainable and industrially viable.

The early detection of faults in the layering process of metal 3D printing is crucial in implementing preventive measures to mitigate the progression of such faults into subsequent layers. Such measures may include altering process parameters or using troubleshooting techniques to identify defects in the powder quality. Integrating sensors and data collection and analysis of the printing process is essential for achieving sustainable outcomes, such as increased material and energy efficiency and substantially reduced waste output. Implementing proactive maintenance methods is a foundation for advancing predictive and in-line feedback process control techniques in the metal 3D printing process monitoring field.

This study aims to discover what hurdles exist in attempting to transfer some of the previous hardware and analysis algorithms from the bespoke powder bed device onto an actual printer. Two key points will be evaluated on the effectiveness of sensors when transferred for use on an existing printer and the feasibility of adding sensors in a non-intrusive method and without modifying a commercial printer in any permanent way.

2.1 Experiment Set-up

Two types of sensors were selected to be integrated under the baseplate: three Rosette strain gauges of type FRAB-1-350-11 and a Kistler 8152C Acoustic sensor with a range of 100KHz to 900 kHz.

A new base plate was designed and modified to integrate these sensors into the base plate. The purpose of the strain gauges is to measure the baseplate's warpage due to the build part's shrinkage solidifying. Acoustic sensors can detect any stress waves generated within the building part during the printing process. Most commonly, the stress wave is generated from a part delaminating from the build plate due to the excessive stress build-up. A stress build-up is typical with laser powder-bed fusion processes due to the large temperature gradients.

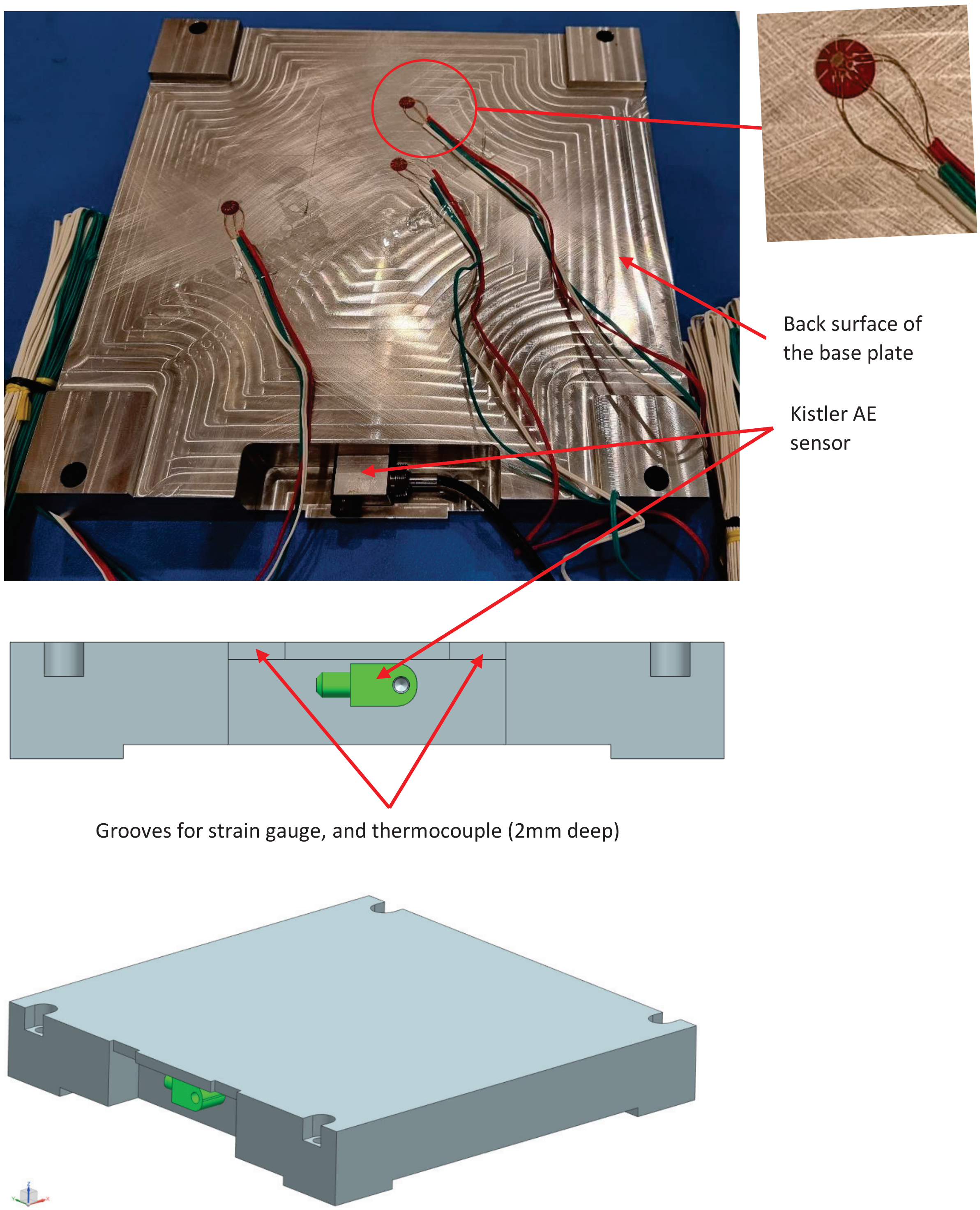


Figure 1: 3 Strain gauges mounted beneath the modified steel base plate with the Kistler acoustic emission sensor placed at the side

2.2 Approach for Incorporating Sensors

The sensors chosen have been decided by ARTC and are best mountable under the build plate. The strain gauge will experience the highest strain on the build plate and will not affect other areas of the printer. Due to delamination or melt pool signals occurring on the build plate, placing the acoustic emission sensors away from the build plate may result in weaker or multiple false signals due to increased echo generation over a longer distance.

To incorporate these sensors under a build plate, the team designed a modified build plate with a sufficiently large cavity to include these sensors.

The sensors are connected through the chamber using a custom-fabricated port cover on an unused EOS M 290. The electrical feedthroughs allow power to be fed into the chamber and data to be piped out. This approach is more practical during the development phase of such sensors as there will be little feedback, data bandwidth or battery life if we use battery-powered sensors for this study.

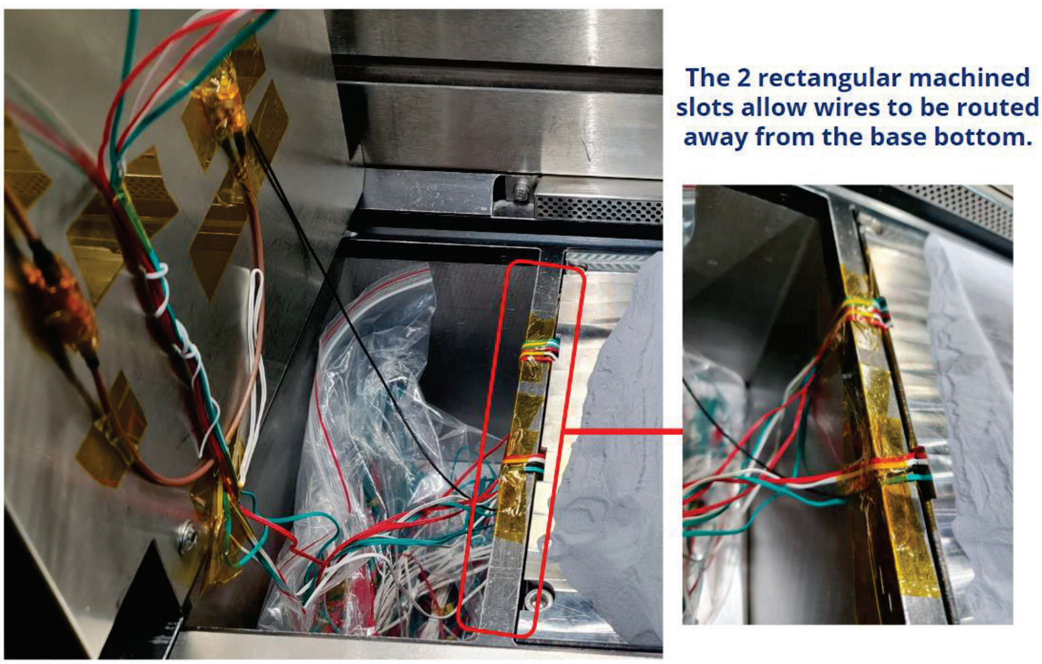


Figure 2: Set-up of the feasibility study where wires are routed from the slots, down to the powder collector and up to the PCB connection

To ensure the safety of the electronics within the chamber, a 200mA fast-acting fuse was used on the +5V voltage power supply, preventing the risk of sparks. The airtightness of the chamber was verified by checking if it could maintain a low oxygen concentration of 0.1%, as required for printing AlSi10Mg. If the chamber is not airtight, the printer will not operate due to the manufacturer's built-in safety interlocks.

2.3 Designing of the secure holder for PCB with Separator board

The wires routed out of the baseplate were connected to the PCB board attached to the modified port cover. This is with the octagonal shape PCB mounted on top of the port cover. I was tasked to designing the secure holders for PCB with a Separator board that aims to solve 2 main problems. First, the 2 wire-connected PCB must be securely fastened to prevent both from dropping into the collector bin of the printer. Secondly, the back pins of the 2 PCBs must not touch one another, which may cause short-circuiting.

To capitalise on the FDM printer for fast prototyping, I designed it with 6 holes of 6.4mm to fasten the part onto the port cover. The 2 semi-circle opening allows the PCBs to be attached to the existing PCB with right-angled pin headers. To prevent the pin from touching each other, the straightforward solution is to have a divider between the 2 PCBs. The rectangular slot fastens the 2 PCB with zip ties. The simple yet effective features solved the main challenges with a fast design-to-implementation solution.

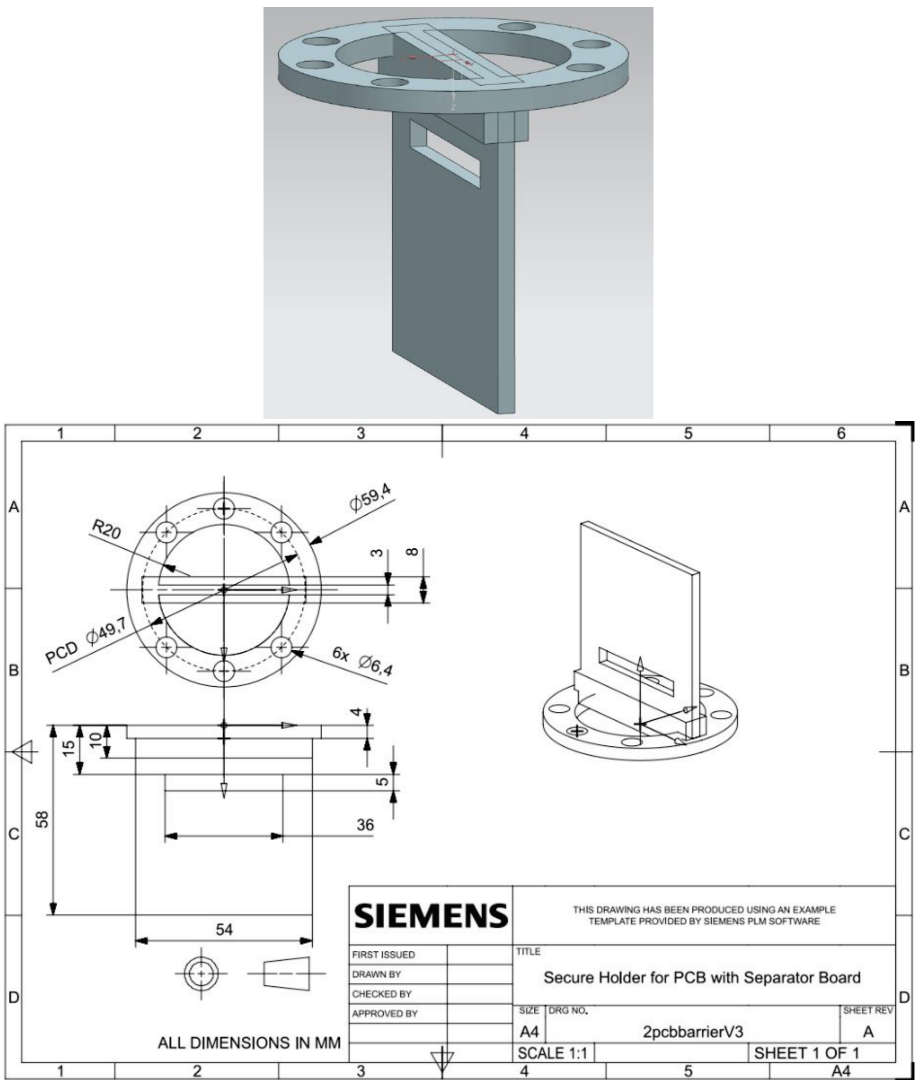


Figure 3: 3D model and drawing of the secure holder for PCB with separator board

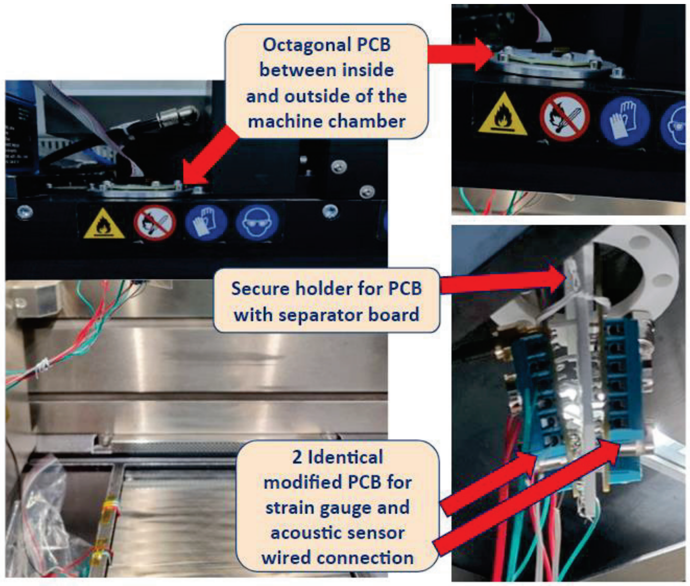


Figure 4: Wired connection for the acoustic emission and strain gauge wires routed out from the base plate to the adaptor PCB then to the octagonal PCB

2.4 Outcome of feasibility study

The team and I learned how to properly install the sensors on the build plate and design any necessary cavity for them through the feasibility study. However, the print did not work due to errors in the instructions for modifying the build plate.

Figure 5: Aftermath of the print of AlSi10Mg powder on steel base plate at the height of about 0.48mm

The print attempt was unsuccessful at a very early stage, approximately 0.48mm from the start, due to the build plate being made of mild steel, which is unsuitable for the AlSi10Mg printer. This caused poor adhesion and delamination of the first few layers of the printed parts. The modified steel plate's curvature and rough surface also impacted the print. It had an inverted bowl shape with a centre that was taller by an estimated 50µm than the corners, which resulted in the first layer of powder being thicker than usual, increasing the likelihood of a delamination.

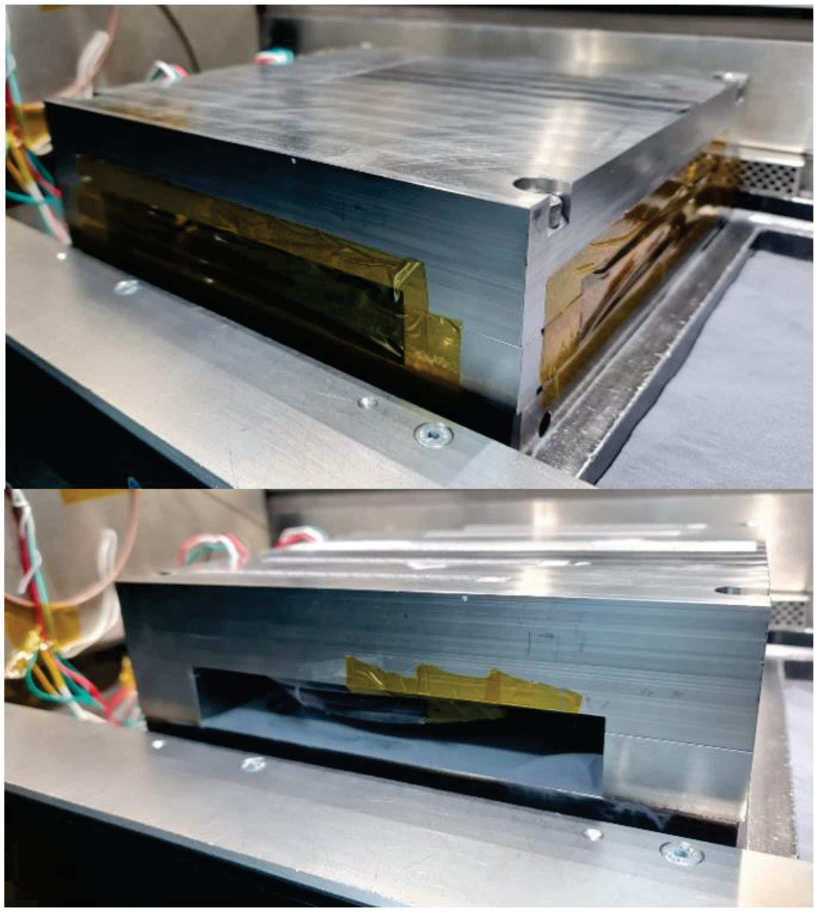


Figure 6: (Top) Kapton polyimide thermal insulation tape was used to cover the sides to prevent the AlSi10Mg powder from flowing via the rectangular gap. (Bottom) After removing the tape, a small heap of AlSi10Mg powder is visible underneath the base plate

Integrating the sensors into the chamber revealed several essential considerations for the future sensorisation of AM printers. One issue identified was the design of the cavity in the build plate. The cavity had numerous openings on the sides, which caused the powder to enter. Two potential scenarios may occur that result in negative consequences. Firstly, the strain gauge contacts could have been shorted out, rendering the collected data meaningless.

Secondly, powder coating over the build volume for each print layer may be insufficient due to powder continuously falling into the cavity through the gap between the build plate and the surrounding chamber wall. One solution is to modify the build plate by taping it to prevent powder from entering the cavity, however, future build plates should be designed not to have large openings on the side walls.

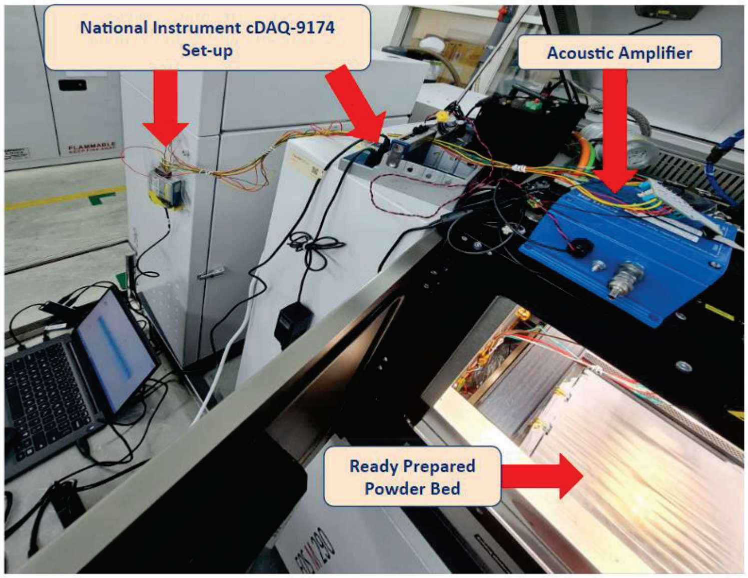


Figure 7: Routing multiple wires required careful consideration of spatial and safety factors while keeping the printer chassis and processes unaltered.

2.5 Impracticality of routing all the strain gauge contacts

Another important lesson learned during the process was the impracticality of routing all the strain gauge contacts from the bottom of the modified build plate to outside the chamber. The implementation of multiple sets of strain gauges on the bottom of the build plate resulted in 12 wires leading to the 2 strain gauges connected to an NI 9236 quarter-bridge strain gauge data acquisition module outside the chamber. The wires were taped securely to prevent interference, but this caused issues with excess wires getting caught in the way of the recoater blade when the building platform moved upwards.

To address this, in the future, it would be more practical to read out the strain gauges during printing under the build plate with a custom high-temperature-resistant electronics board. This way, only power and data would need to be fed through the chamber.

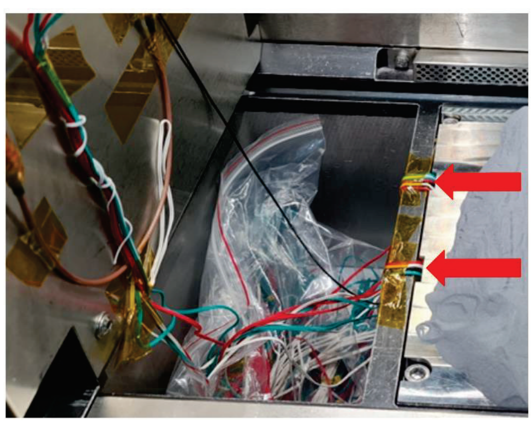


Figure 8: 2 rectangular machined slots allow wires to be routed away from the base bottom. This is found to be poor practicality of wiring out all the strain gauge contacts.

3. M5Stack FIRE IoT Development

The objective of using the M5 stack FIRE device is to monitor and view the signal tower status of the EOS M 290 and DMG MORI SLM 30 remotely. The M5 stack is an industrial internet of things (IIoT) device that can collect data from a 3D printer, such as temperature, humidity, or in our scenario, the machine status, and send it to a remote location for analysis. The M5 stack FIRE device is a modular, stackable, scalable, and portable device powered by an ESP-32 core. It makes it open source, low cost, full function, and easy for developers to handle new product development. The steps will cover the connection of the M5 stack to an EOS M 290 metal 3D printer. It performs functions from collecting data via an InfluxDB time series database and developing code for the M5 stack using C++ on the Platform.io IDE. The full implementation of the M5 stack has been successfully done on the DMG MORI SLM 30 by Dr. Tan.

My role in this sensor development will be to transfer the existing solution to the EOS M 290 printer to expand the monitoring process. Dr. Tan developed all the work specified in the M5 Stack, and my involvement will be noted explicitly in the work explanation.

Figure 9: M5Stack FIRE IoT Development Kit

3.1 Monitoring and Viewing Signal Tower Status

One of the main reasons to remotely monitor the signal tower status of a 3D printer is to ensure that the printer is running correctly and to detect and address any issues that may arise quickly. For example, if the signal strength is low, it may indicate a problem with the network connection, which can lead to print failures or other issues. Additionally, it is helpful if the operator is away from the printer or working on other tasks. The status can also use to keep an eye on a running print job and check if there are any issues with the print job, such as signal strength and error messages.

3.2 Connecting to the IIoT Device

The first step in integrating the M5 stack for data collection and presentation is to connect the device to the EOS M 290 metal 3D printer. The device can then collect data from the printer and send it to a remote location for analysis.

3.3 Collecting Data via InfluxDB

Once the IIoT device is connected to the printer, it can collect data and send it to a time series database such as InfluxDB. This database is optimized for time-stamped data and can handle high write and query loads. The data can be stored and analysed over time to monitor the printer's performance and detect any issues.

The code was written in C programming language, which defines the constants used to connect to an InfluxDB database. These constants, described below, are used in the code to connect to the InfluxDB server, authenticate using the token, and write data to the specified bucket in the selected organization.

3.4 IoT Development using C++ and Platform.io IDE

The next step is to use Platform.io IDE, an extension for Visual Studio Code, to develop the code for the IIoT device. C++ is used as the programming language to collect and process the data from the printer and send it to InfluxDB. I spent most of my time in this area to understand and break down Dr. Tan's code to write the script for the EOS M 290 from the DMG MORI SLM 30.

The M5Stack device, an ESP32-based development kit, utilizes several libraries, including Arduino.h, M5Stack.h, WiFiMulti.h, InfluxDbClient.h, and Adafruit_NeoPixel.h, to perform functions such as connecting to WiFi, sending data to an InfluxDB database, and controlling the device's built-in LED lights. Overall, this code is designed to read data from an InfluxDB database and use that data to control the device's LED lights. The code also includes logic for connecting to a WiFi network and reconnecting if necessary. The code explanation will be of specific areas of the code I edited.

3.4.1 Setting of Previous and Incoming States

To let the M5Stack digest the data for display, the code will need to call in the state of the printer via the InfluxDB's database (from sub-point 3.3 Collecting Data via InfluxDB). The state of the printer refers to the printer's statuses such as being in fine operating condition (green light) or calling for intervention (red light). Therefore, the code sets up variables to track the state of two lights, a red light and a green light.

Initially, the old code had the blue light to accommodate the operator mode, but this is not a feature for the EOS M 290. The change was to use the green light state instead of the blue light. The variables "was_red_light" and "was_green_light" are used to store the previous state of the lights. The variables "is_red_light" and "is_green_light" are used to store the incoming state of the lights.

```
int8_t was_red_light = -1; // Previous state of red light
int8_t was_green_light = -1; // Previous state of green light

int8_t is_red_light = -1; // Incoming state of red light
int8_t is_green_light = -1; // Incoming state of green light

//DMG_SLM30: blue light (operator mode), red light (error, requires attention)
//PHR_M290: green light (print in progress), red light (error,requires attention)
```

Figure 10: Green light states used for the EOS M 290, changed from previous blue light states

3.4.2 Creating a Flux Query to Access the Script Editor

The next step is to add in the code where the M5Stack refreshes the local states (updating the latest data) in the InfluxDB database. The specific code is gotten from InfluxDB itself, written in InfluxQL, which is a SQL-like language used to interact with the InfluxDB database.

The query data used in the code is accessible by creating a flux query from InfluxDB's flux builder by choosing the right equipment (eqpt), PHR_M290, shown in figure 11. The pins selected from the flux builder must also be identical to the IIoT hardware connection. Flux is a functional data scripting language designed for querying, analyzing, and acting on time series data.

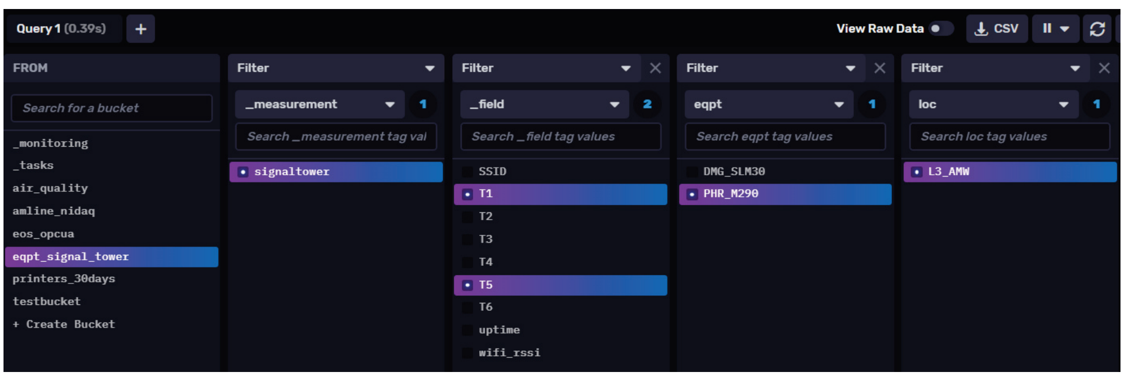


Figure 11: Flux builder panel to create a Flux query to retrieve data

After creating the flux query, the script is accessible to the script editor, shown in Figure 12. It is possible to type out the query manually, but it is readily given by InfluxDB's script editor. Next, I placed the script in the refresh_local_states function.

```
from(bucket: "eqpt_signal_tower")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "signaltower")
  |> filter(fn: (r) => r["_field"] == "T1" or r["_field"] == "T5")
  |> filter(fn: (r) => r["eqpt"] == "PHR_M290")
  |> filter(fn: (r) => r["loc"] == "L3_AMW")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

Figure 12: Script editor shown, after submitting the flux query from the flux builder, to be placed in the main code

After the flux query is placed the script in the refresh_local_states function, the string variable query begins by selecting data from a specific "bucket" in InfluxDB (constant "INFLUXDB_BUCKET.") Then, it filters the data to only show the last 30 minutes by using the "range" function. Next, it filters the data only to show the data related to specific equipment, defined by the constant "EQPT_NAME."

```
void refresh_local_states() {
  // This fetches the latest data from InfluxDB (but will not act on the data)

  bool serialprint_query = false;

  for (int query_number=0;query_number<=1;query_number++) {

    String query = "from(bucket: \"" INFLUXDB_BUCKET "\")";
    query += "|> range(start: -30m)";
    query += "|> filter(fn: (r) => r.eqpt == \"" EQPT_NAME "\")";
    //query += "|> filter(fn: (r) => r.SSID == \"webserver\")"; // For debugging use
    query += "|> filter(fn: (r) => r._measurement == \"signaltower\")";
    if (query_number==0) {
      query += "|> filter(fn: (r) => r._field == \"T1\")"; // On PHR_M290, T1 is GREEN, T5 is RED
    } else if (query_number==1) {
      query += "|> filter(fn: (r) => r._field == \"T5\")"; // On PHR_M290, T1 is GREEN, T5 is RED
    }
    query += "|> drop(columns: [\"_start\", \"_stop\"])";
    query += "|> group()";
    query += "|> sort(columns: [\"_time\"])";
    query += "|> tail(n: 1)";

    FluxQueryResult result = client.query(query);
```

Figure 13: Script from script editor is placed into the code to retrieve specific data from the InfluxDB database

Finally, the query data is used for display with the lit pixels on the M5 Stack shown in Figure 14. This is done by allocating the function called "digest_incoming_states," that is used to act on the incoming state of the red light. The function first checks if the incoming state of the red light is different from the previous state of the red light, and if it is, it proceeds to check the value of the incoming state of the red light.

If the incoming state of the red light is less than or equal to 0, the function interprets this as a green condition. The function uses a for loop to set the color of 10 pixels to display the green color on the pixels.

If the incoming state of the red light is equal to 1 or 2, the function interprets this as a red condition. The function then calls the "pixels.show()" function to display the red color on the pixels. I edited the code to fit the simple condition when the EOS M 290 is green and red. I changed the code that controls the blue light to comments to preserve the flow and history of the code written. This is for future reference if the next developer has a similar machine that uses the additional operation.

```
void digest_incoming_states() {
  if (is_red_light != was_red_light) { // Acts on the incoming red light
    if (is_red_light<=0) {
      // Then we now have a green condition
      for (int p=0;p<10;p++) {
        pixels.setPixelColor(p, pixels.Color(0,LED_BRIGHTNESS,0));//(green)

        /*if ((p==3)) {
          pixels.setPixelColor(p, pixels.Color(0,LED_BRIGHTNESS,0)); // (black) Handles pixel 3 specially for blue
        //} else if ((p==4)) {
         // pixels.setPixelColor(p, pixels.Color(0,0,LED_BRIGHTNESS)); // (blue) Handles pixel 4 specially for blue
        } else {
          pixels.setPixelColor(p, pixels.Color(0,LED_BRIGHTNESS,0));//(green)
        }*/
      }
      pixels.show();
      Serial.println("Changed into green LEDs.");

    } else if ((is_red_light==1) || (is_red_light==2)) { // Then we now have a red condition
      if (was_red_light<=0) {
        for (int p=0;p<10;p++) {
          pixels.setPixelColor(p, pixels.Color(LED_BRIGHTNESS,0,0));
          /*if ((p==3)) {
            pixels.setPixelColor(p, pixels.Color(LED_BRIGHTNESS,0,0)); // Handles pixel 3 specially for blue
          //} else if ((p==4)) {
          //  pixels.setPixelColor(p, pixels.Color(0,0,LED_BRIGHTNESS)); // Handles pixel 4 specially for blue
          } else {
            pixels.setPixelColor(p, pixels.Color(LED_BRIGHTNESS,0,0));
          }*/
        }
        pixels.show();
        Serial.println("Changed into red LEDs.");
```

Figure 14: Edited the code removing blue light and added green light for the different state of the printer

3.5 Uploading to the M5Stack FIRE IIoT Device

Once the code is developed and tested, it is uploaded to the M5 STACK FIRE for testing. In this example, a red light (error) is induced by opening the chamber door suddenly while moving the build platform. This simulates an error condition and allows the developer to test how the code handles such situations.

Figure 15: When the signal tower is not blinking, M5 Stack is lit green. When the signal tower flashes red light, the M5 Stack is lit red.

3.6 Summary of the M5 Stack Implementation

Overall, the above steps demonstrate how an IIoT device can remotely monitor and collect data from a 3D printer and make it available for analysis and visualization. This allows the operator to track the printer's performance, detect and troubleshoot issues, and improve the efficiency of the printing process. The M5 Stack, in this scenario, provides a convenient and efficient way to remotely monitor and control the printer while providing a means for data collection and analysis.

The code is then uploaded to GitLab (web-based Git repository manager). It provides source code management (SCM) and enables continuous integration. This allows easy review and merges of code changes for improvements or debugging.

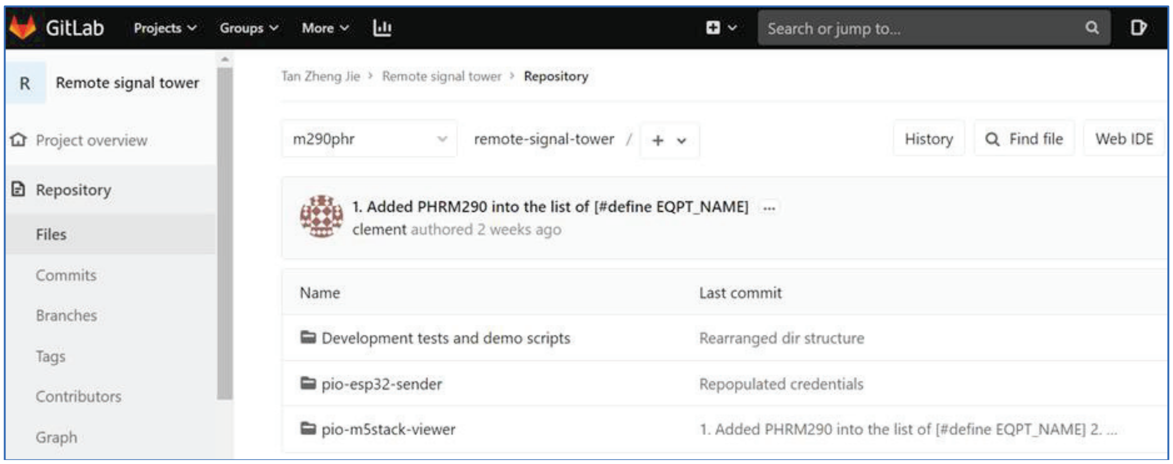


Figure 16: GitLab webpage where the new code is added to the new branch (m290phr), separated from the main branch

4. Webcam Streaming in EOS M 290 Chamber

4.1 Anker PowerConf C200 Webcam Integration into Print Chamber

The objective was to implement a real-time video streaming system for monitoring the laser powder bed fusion process in an industrial setting. Several considerations were considered to achieve this, including spatial, safety, network, hardware and software, and preservation of the printer chassis and processes.

The first step was to identify a suitable location for the camera to provide an unobstructed view of the printer's chamber. This involved installing the camera via a setscrew and positioning it at the optimal height and angle while ensuring that the camera location was in a safe and accessible area, away from any moving parts (recoater blade) or hazardous areas such as being in the way of the laser. The webcam must also avoid the reddish hue lens as it is for the thermal imaging camera.

Figure 17: Anker PowerConf C200 Webcam is attached via a set screw on a modified port cover. A 15cm USB-C is used to connect the webcam to the customised PCB.

Safety was also a significant concern, with the camera and its cables securely mounted and protected from any potential damage, such as heat or dust generated by the printer. The webcam's operating temperature is up to 50°C
Additionally, the camera was placed in a location that did not interfere with the printer's regular operation or impede any emergency stop or shut-off procedures.

The camera was connected to a network that allowed remote access through a wired or wireless connection. This enabled operators to view the video stream from a remote location, such as a control room or a mobile device while ensuring that the network was secure and protected from unauthorized access.

The video streaming server Yawcam was set up on a device with sufficient processing power and storage capacity to handle the video stream. The software used to stream the video was also compatible with the camera and the network and had the ability to adjust the video quality and resolution as needed.

Finally, it was essential to ensure that the printer chassis and processes were unaltered during the implementation of the video streaming system. The camera and its accessories were not allowed to affect the regular operation of the printer and were not allowed to cause any damage to the printer or its components.

In conclusion, by considering these factors and following best practices for camera installation and network security, the proposed video streaming system allows for remote monitoring of the laser powder bed fusion process while preserving the integrity of the printer chassis and processes. This system can optimize the laser powder bed fusion process and provide valuable data for process control and predictive maintenance.

4.2 Python and HTML Code Development

In addition to the hardware and network considerations, a crucial component of the process monitoring system is the software that controls video streaming and data analysis. To accomplish this, a python script was developed to handle the HTTP server and request handling for the video stream. The script utilizes the built-in http.server and socketserver modules to create a simple HTTP server that listens on port 8002. A custom request handler called MyHttpRequestHandler is defined in the script, which overrides the do_GET method always to serve the index.html file when a GET request is received. This HTML file serves as the primary interface for the live video stream of the EOS M290 3D printer, allowing operators to view the state of the print job in real-time remotely.

```python
import http.server
import socketserver

PORT = 8002
class MyHttpRequestHandler(http.server.SimpleHTTPRequestHandler):
    def do_GET(self):
        self.path = 'index.html'
        return http.server.SimpleHTTPRequestHandler.do_GET(self)

Handler = MyHttpRequestHandler

with socketserver.TCPServer(("192.168.192.103", PORT), Handler) as httpd:
    print("Http Server Serving at port", PORT)
    httpd.serve_forever()
```

Figure 18: Python script to host the webcam stream from Yawcam application

The second code is an example of an HTML file the above Python server can serve. The HTML file includes a <link> element in the stylesheet "style.css". The <title> element sets the page title to "PHR EOS M290 Print Job".

The HTML also includes a <div> element with class "center" which will center the contents of the div. Inside the div, there is a <p> element that has a <h1> tag inside it. The <h1> element displays text "PHR EOS M290 Print Job 1". The HTML file also includes an <img> tag that displays a video stream from the IP address "http://192.168.192.103:8081/video.mjpg?q=30&fps=33&id=0.2900691557912334&r=1667816092915" with a width of 940 and height of 528.75 and angle of 90 degrees. Overall, this HTML file is used as the index page, which will be served by the python server and display the live video stream of the print job of the EOS M 290 3D printer.

```html
<!DOCTYPE html>
<html lang="en">

    <head>
        <link rel="stylesheet" href="style.css">
        <meta charset="UTF-8">
        <title>PHR EOS M290 Print Job</title>
    </head>

<div class="center">

        <p><h1>PHR EOS M290 Print Job 1</h1></p>
        <img src="http://192.168.192.103:8081/video.mjpg?q=30&fps=33&id=0.2900691557912334&r=1667816092915"
            width="940" height="528.75" angle="90"/>
</div>

</html
```

Figure 19: HTML code for the webcam stream to be presented on a webpage

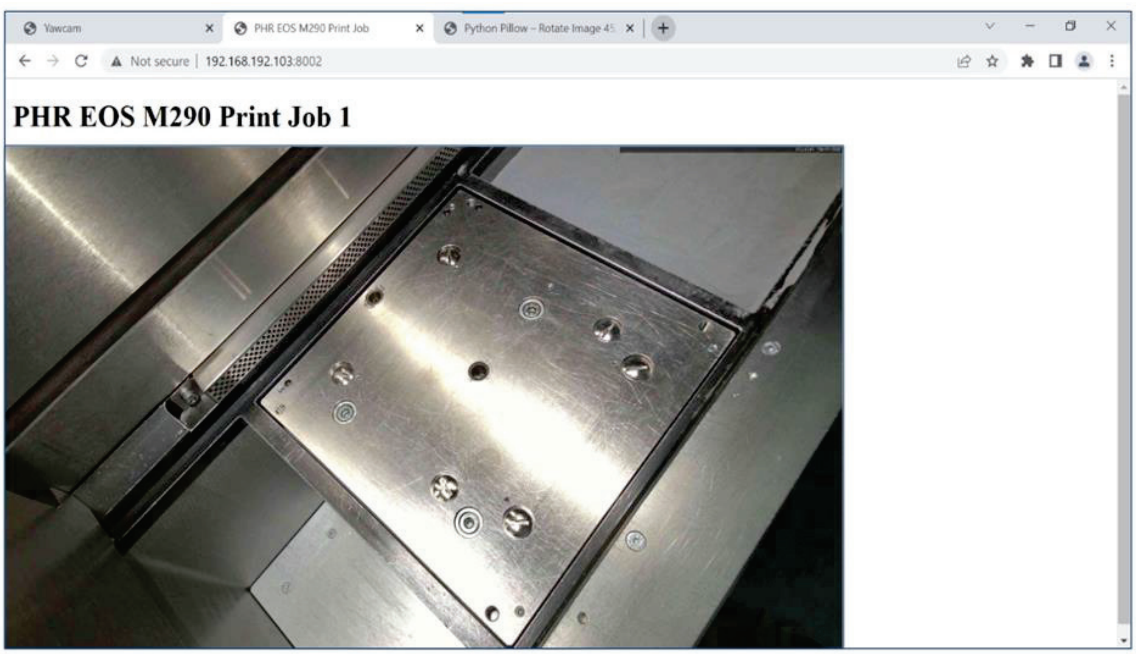


Figure 20: Safe and unobstructed view of the webcam stream showing a full coverage of the building platform

5. Spectrometer Integration and Development

5.1 Project Background

To support an upcoming program on powder reuse, this project will evaluate how the elemental composition of the powder can be determined by the optical spectrum from its plasma emission when the powder is excited by a laser. Plasma emission is governed by the set of discrete energy levels present in a chemical system and varies from material to material, and also ought to be influenced by slight changes in the chemical properties of the feedstock powder. This work involves design aspects to extract spectroscopic spectra in-situ from live prints [5,7].

5.2 Working Principle of Spectrometer

A spectrometer is an instrument used to measure properties of light, including its wavelength and intensity. It works by splitting light into its individual wavelength components, also known as a spectrum, and then measuring the intensity of each component. The process of splitting light into its individual wavelengths is called dispersion, and it can be achieved using a prism or a diffraction grating [6].

5.3 Spectroscopy of Fluorescent Light with Compact Spectrometer-Firefly4000

To test out the spectrometer to verify its working condition, I placed the spectrometer under a fluorescent light to get the spectrum of intensities with wavelengths from 350nm to 1000nm. The spectrometer receives a spectrum of light that the fluorescent light is emitting. Fluorescent lights emit light through a process called fluorescence, which occurs when certain materials absorb ultraviolet (UV) light and then re-emit it as visible light. I cross-checked with the spectrum with another research paper titled “Fluorescent Light Incites a Conserved Immune and Inflammatory Genetic Response within Vertebrate Organs.” The two graphs have an identical shaped of spectrum of intensities against various wavelengths

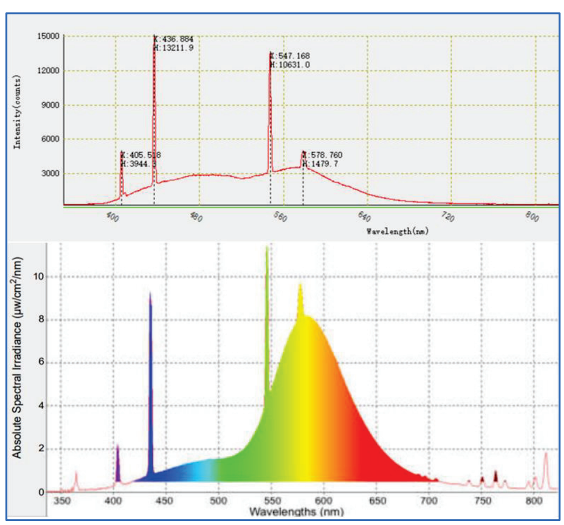


Figure 21: Comparing with the Compact Spectrometer-Firefly4000 (Top) and research paper (Bottom), both have identical shaped of spectrum of intensities against various wavelengths

5.4 Hardware Connections for the Spectrometer on EOS M 290

The spectrometer will be placed beneath the EOS M 290 hood, outside of the print chamber. The SMA905-SMA905 Fibre Optic Patchcord with a 1.0mm core allows light to pass through from one end to another. The cable enables the transmission of data over at high speeds.

A fibre collimator is attached at the opposite end of the Fibre Optic Patchcord. This device that collimates, or aligns, light from a fibre optic cable into a parallel beam. A focusing lens is used in conjunction with the collimator to focus the collimated light to a smaller spot size.

The purpose of a fibre collimator focusing lens is to reduce the divergence angle of the collimated beam and increase the energy density at the focus point. This allows for more efficient coupling of the light into the spectrometer for more precise control of the beam shape for a particular application. Precise alignment of optical fibres and focused light beams is crucial as the laser-produced plasma plume and expands away from the melting pool surface.

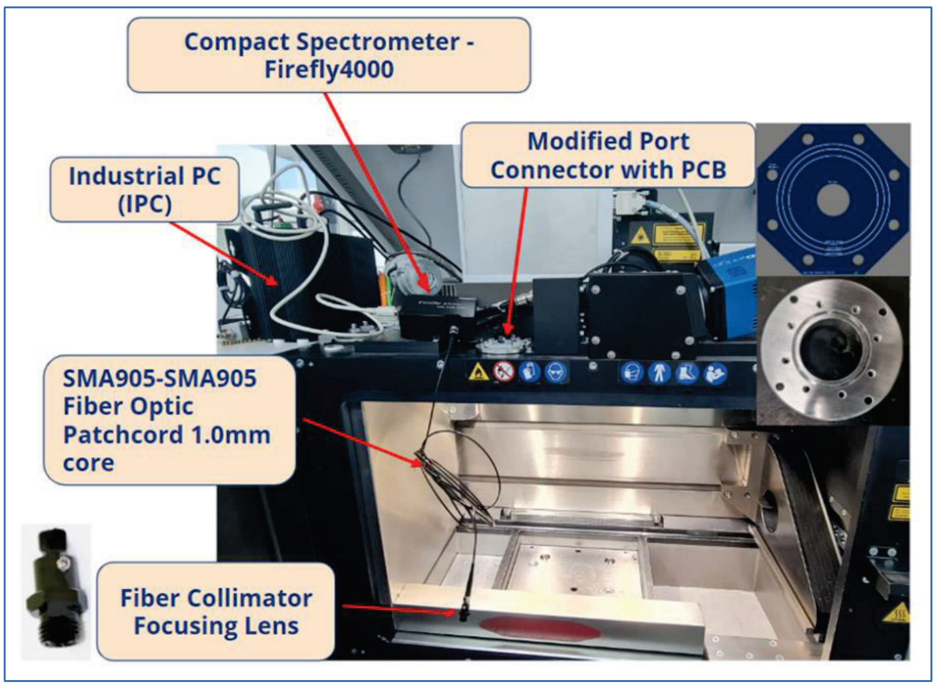


Figure 22: Visual representation of the spectrometers position with the fibre optic cable. The cable will pass through the 13mm hole.

5.5 Deciphering EOS M 290 Components' Communications through Wiretapping

The purpose is to understand the request and response data from communications made from the EOS M 290 hardware components, but not limited to, different grades of Industrial PCs (IPCs), servomotors, stepper motors. The goal is to match up the printer's various key parameters, such as the height of the building platform, the position of the dispenser and the recoater blade, at a particular time frame. With this information, the data from the emission acoustic sensor and the optical spectrometer can be analysed effectively by knowing the printer's state.

A scenario is using the spectrometer to get the optical spectrum from the plasma emission when the powder is excited by a laser. Hence, the printer must have a stationary building platform height, position of recoater blade and dispenser platform to allow the laser the powder. To retrieve the time frame of the process, it is by deciphering the parameters of the printer's internal components.

This principle works the same for other components and processes such as the recoater blade speed, building platform height, and dispenser's height. These parameters help to build a more complete environment of the print chamber to assist the spectrometer to collect useful data. For example, the recoater blade is activated by the printer to sweep a new layer of powder. The time request and response data can be taken from the communications of the recoater blade internal motor. With that particular time frame, analysis of the emission acoustic sensor will be of interest and easier to retrieve the emission acoustic graph. This will be time efficient as the specific time frame of the recoater blade will sweep the building platform periodically.

5.5.1 Accessing the communications made within the industrial components

The first step is to access the request and response data with a communication protocol between the IPC and the industrial components. This is done by Dr. Tan by hardware tapping of the IPC, to intersect and eavesdrop on the communication.

Once the communication protocol has been established, the IPC can send requests to the industrial components to retrieve data on the various key parameters. The industrial components will then respond with the requested data. This data can then be analysed to match up the key parameters at a particular time frame. Dr. Tan used Wireshark, a free and open-source packet analyser, to capture and analyse network traffic. It captured packets and displayed them in a detailed and easy-to-read format. I am able to filter packets by protocol, IP address, port, and message type to view the traffic that is relevant to my analysis.

5.5.2 Analysing Packets from a Completed Print Job

A titanium print job was used to collect the printer's network traffic over a period of 30.6 hours or 1 day, 6 hours, 35 minutes . The whole print job captured a total of 337 files with each file of 19 megabytes, containing 500,000 individual packets.

Dr Tan wrote 2 python scripts to read out the packet's data (hexstring) and retrieve the key information such as the source, destination, message type and time. My role was to identify which packets of different data could possibly hold similar or potential information which correlates to the printer's components movement and processes.

5.5.3 Reading and Converting the Powerlink Packets (Done by Dr. Tan)

This code is written in Python and it is used to extract information from a series of packet capture files (PCAPs) captured from Ethernet Powerlink (EPL) V2 network. EPL is a real-time communication protocol for industrial automation.

His code imports several modules such as os, ast, time, glob, pickle, and binascii. These modules are used for file management, parsing abstract syntax trees, working with time, finding files, serialising and deserializing objects, and converting between binary and ASCII representations. Then, it imports the scapy module, which is a packet manipulation tool for Python. It imports sniff which is used to capture the packets flowing across a computer network. His code uses the glob.glob() function to find all the pathnames of the PCAP files in a specified directory and store them in a list filenames. Then it opens a file named 'testingoutput.txt' for writing. Overall, this code is used to extract relevant information from EPLV2 network packets captured.

As a novice in the field of Python and software development, I acknowledge that I currently lack certain technical skills. However, I am highly motivated and eager to acquire new knowledge in this area. As a self-directed learner, I have employed a variety of techniques to enhance my understanding, including the use of the print() function to break down code and visually identify the purpose of each line. I am confident that with further study and practice, I will be able to develop my proficiency in this field.

5.5.4 Data Analysis of Packets with Matplotlib and InfluxDB graphs

The data has been presented in an excel file (Table 1) with individual hex string converting to its corresponding integer value from the packet capture and sorting them based on their timestamp and node id. This is my starting point for the analysis where the data is well accessible for manipulating and plotting via matplotlib through python.

Table 1: Excerpt of excel sheet sorted by the packets' nodes chronologically b79							
No. of Rows	**Epoch time**	**Node**	**Hexstring**	**b01**	**b02**	**bXX**	**b79**
1	1672990662	1	3832…..	31	31	…….	67
…….	…….	1	…….	…….	…….	…….	…….
…….	1672990662	2	1433…..	41	4	…….	53
…….	…….	2	…….	…….	…….	…….	…….
…….	1672990662	40	9239…..	33	56	…….	85
…….	…….	40	……..	…….	…….	…….	…….

5.5.5 Verification of Graph Quality with Reduced Sample Size

Since there are 67 thousand rows combined with 4 nodes, each node has an estimate of 17 thousand rows. This is a huge sum of rows for each node to be analysed which require more processing power when creating graphs with Python. Hence, I cut down every alternate row to save ample size.

I conducted a test to verify how cutting down the sample size affects the graph's shape and definition produced by matplotlib. I used the node 40, integer column b09, where the graph produces a distinct saw tooth function against time. The packets used are from the titanium print job hence prior set-up and post removal of part is not included. The sample sizes used are 16558 rows (original size), 8279 rows (alternate row) and 5519 rows (1 in every 3 rows).

The shape of the saw tooth function remains similar from all 3 different sample sizes. However, visible fluctuation and variation in the data points results in more distinct peaks and valleys as shown in Figure 23. The smaller the sample it is, the graph presents rougher. Based on these findings, reducing the sample size by every 3 rows presents more data loss. This is not optimal as the purpose of using the graphs is to look out for distinct shapes, peaks, troughs and patterns, and outliers. Therefore, the optimal reduced sample size is at 8279 rows (alternate rows) for time-efficiency in processing and little loss of quality in the graphs.

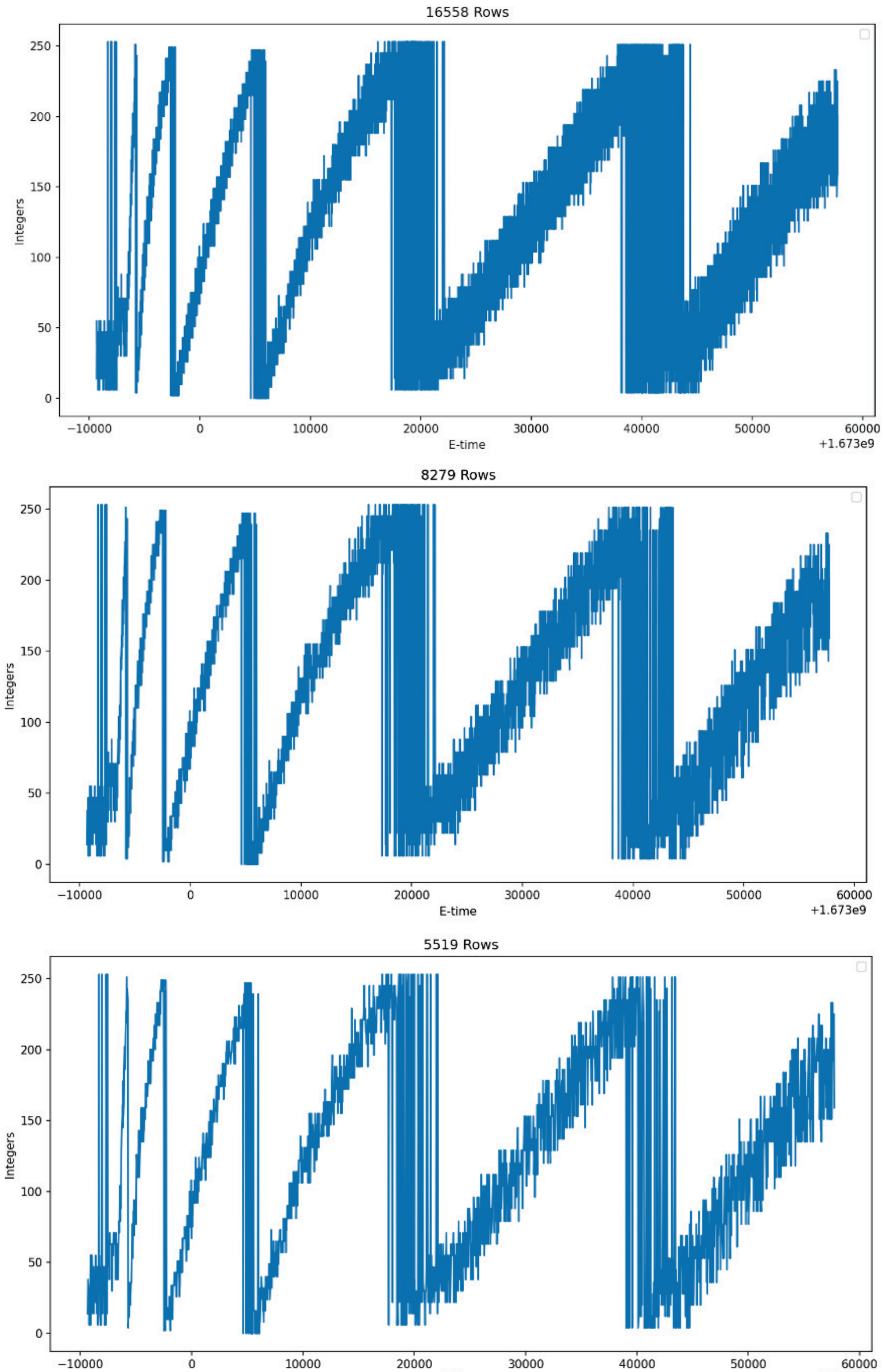


Figure 23: Sample sizes used are 16558 rows (original size), 8279 rows (alternate row) and 5519 rows (1 in every 3 rows).

5.5.6 Matplotlib for Graphs

The x-axis is the time frame from the start and end of print. The y-axis is the individual integers that were converted from hex string to integers. From the long string of data, there are up to 79 individual integer columns. The plan was to present each column against time to see if the data possessed any potential insights such as fluctuations and distinct patterns such as the saw tooth function shown earlier.

My code used the Python libraries such as pandas, openpyxl, matplotlib and numpy. The first section of commented code is for reading in data from an Excel file and formatting the date column, shown in figure 24. Then, the code reads in data from an Excel sheet (node1/ 2/ 40/ 41) in the file 'Tabulated.xlsx' and saves it to the variable old_df. It then creates a new variable df which is a subset of the old_df dataframe, selecting every 3rd row using the iloc method.

```
def format_date(data):
    date_time = datetime.datetime.fromtimestamp(data)
    return date_time

df = pd.read_excel('Tabulated.xlsx', sheet_name='node 40 actual print', usecols="A")
df['epochtime'] = df['epochtime'].map(format_date)
```

Figure 24: Function takes an epoch time (a unix timestamp) as input and returns a datetime object.

The code reads data from an excel file and plots 12 columns of the data in a 3x4 grid on a graph, using the Matplotlib library. It also includes labels for the x-axis and y-axis, and also a legend to display the plotted data. And it is taking every alternate row, instead of all the rows. The code then sets the x-axis label to "E-time" and the y-axis label to "Integers" using the plt.xlabel() and plt.ylabel() functions. The plt.legend() function to display a legend on the graph showing the labels of the plotted data. Finally, plt.show() function to display the graph on the screen.

```
#Create Graph with MATPLOTLIB
old_df = pd.read_excel('Tabulated.xlsx', sheet_name='node2 test')
df = old_df.iloc[::2] #Selects alternate rows

columns = ['b01', 'b02', 'b03', 'b04', 'b05', 'b06', 'b07', 'b08', 'b09', 'b10', 'b11', 'b12']

for i,col in enumerate(columns):
    plt.subplot(3, 4, i+1) #row,column,position
    plt.plot(df['epochtime'], df[col], label=col)
    plt.xlabel("E-time")
    plt.ylabel("Integers")
    plt.legend()
plt.show()
```

Figure 25: Code reads data from an excel file "Tabulated.xlsx' and plots 12 columns of the data in a 3x4 grid on a graph to analyse the columns with distinct patterns for potential analysis

5.5.7 Cross Checking with Existing InfluxDB Database

A way to verify the data given from the string of data from the packets is to cross-check with the existing data collection with InfluxDB. I picked out key parameters that are coherent to the analysis, such as the recoater speed & position, recirculating filter system, dispenser position, building platform position & temperature and collector platform position.

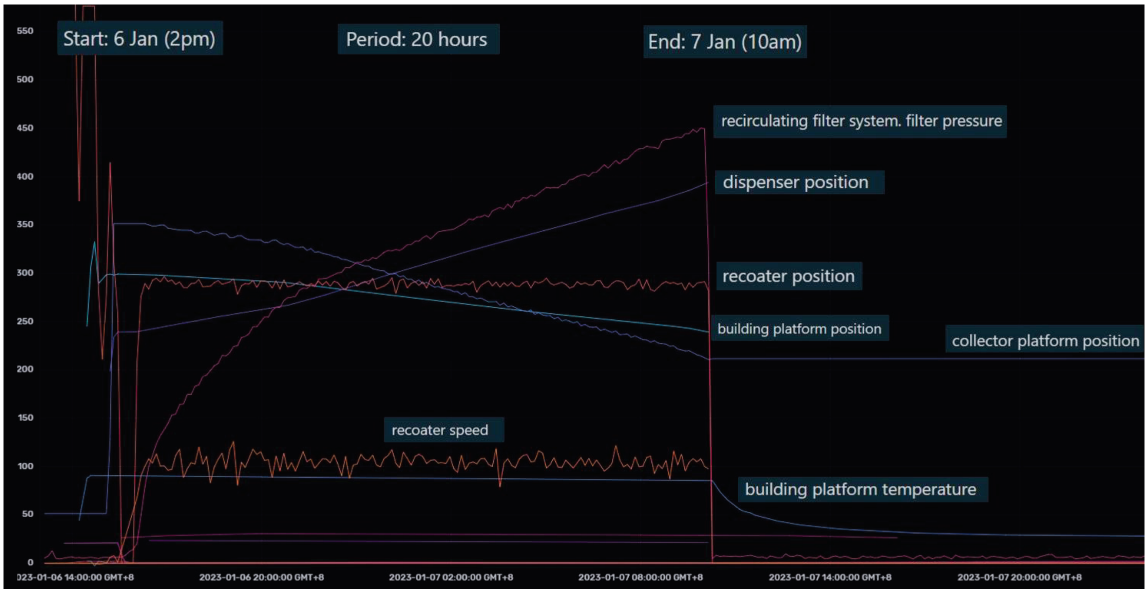


Figure 26: Graph of InfluxDB data with key parameters of the EOS M 290 statuses; the outline of each status is taken to match any similarities of the packets data's column of integers

5.5.8 Analysis Outcome (Node ID 2 Servomotor Corelating to the Recoater)

From the electrical cabinet of where the servomotor resides, the node indication of the servomotor is "2". The excel sheet of Node ID 2's columns of integer data were analysed with InfluxDB data of the recoater blade position and speed shown in the figure below.

From the 5 graphs plotted in Matplotlib, the columns b17, b18, b23, b24, b39 shows similar shape, where there is valley shaped dipped near the start of the graphs. the other columns were taken out for comparison due to 0 values throughout the print period. Similarly, all the columns of the other node IDs (1, 40 and 41) were match against, but none had the same findings as node 2.

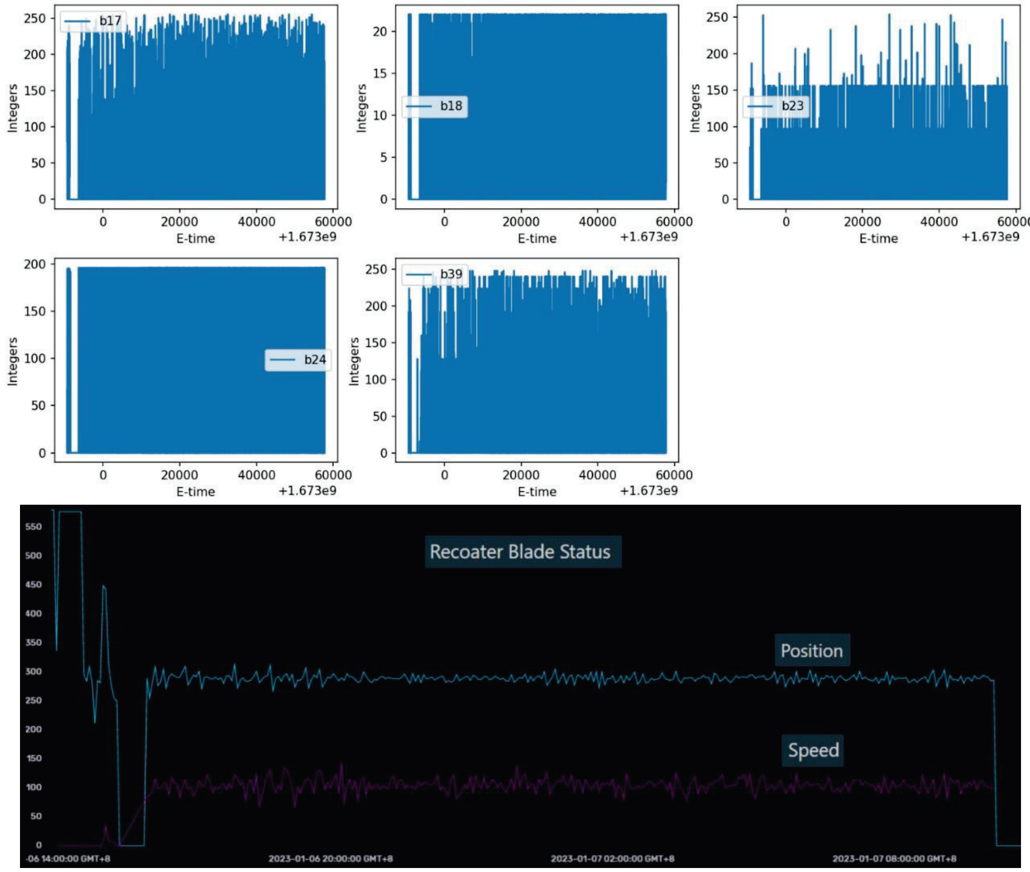


Figure 27: Recoater blade's position and speed during the print job

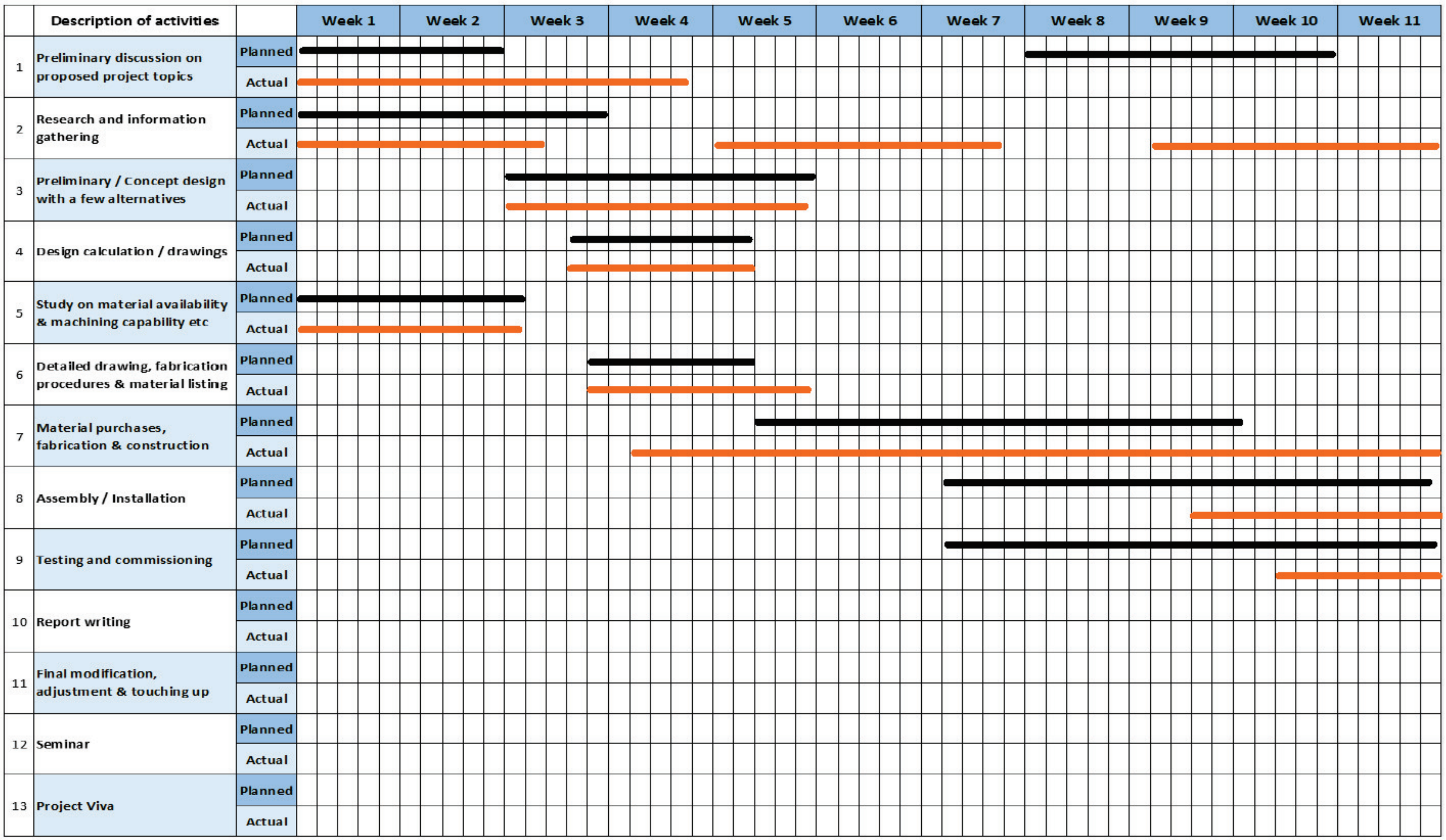
5.5.9 Next Steps for the Analysis

As the task was brought up by Dr. Tan on 10th January 2023, the progress of the analysis is up to the stage where I am matching the existing known node to the correct components used in the printer. This is the simple baseline process of subsequent findings as it is crucial to have a fair analysis of other nodes and various hidden components that may not be immediately noticeable. From the previous findings of node 2, it is made possible due to the fact that the servomotor is indicated as "2". However, subsequent components crucial such as the height of the building platform, the position of the dispenser are indicated as 0.

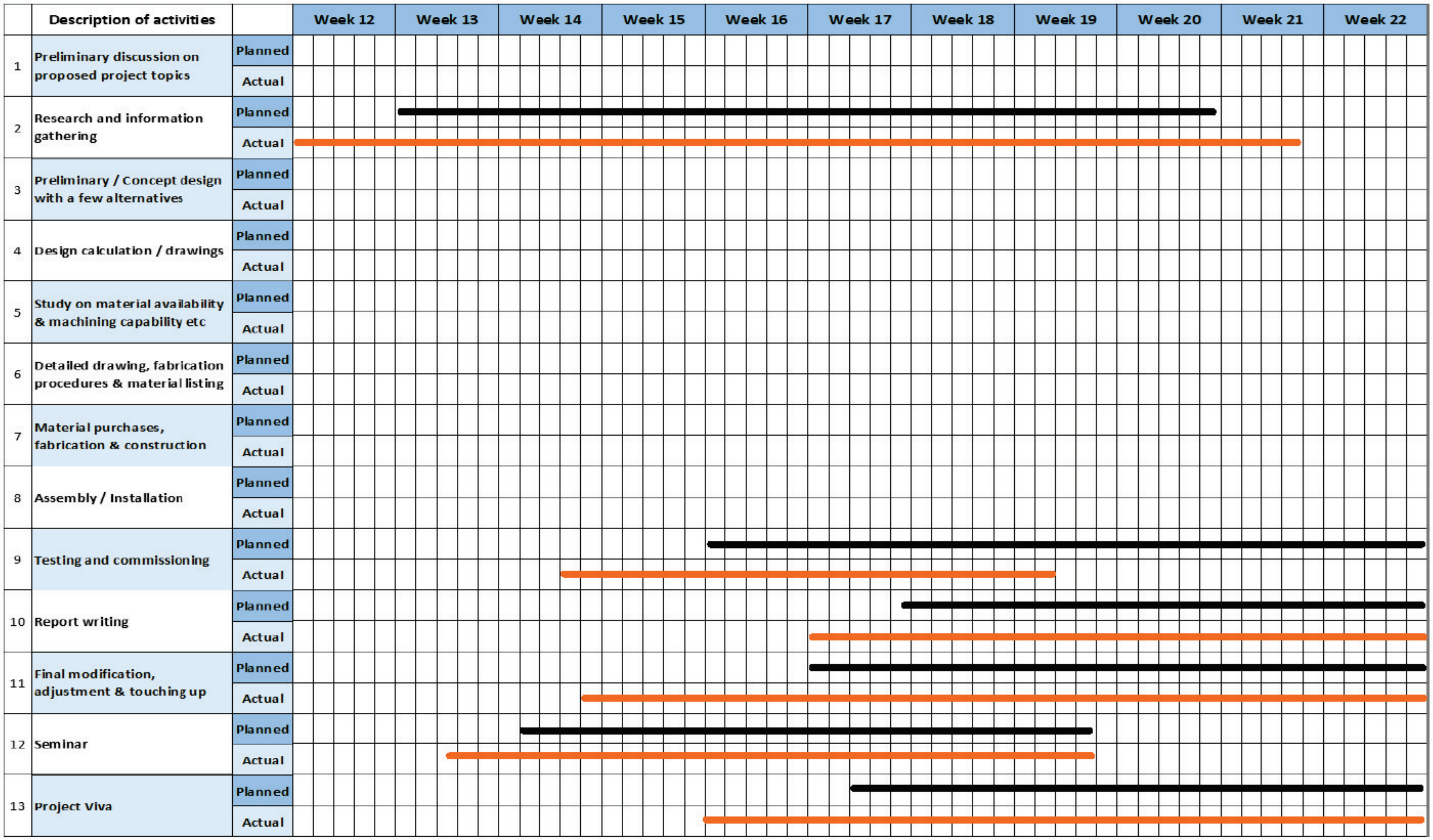
The subsequent plan are to run a mechanical trial test where the IPC retrieves the packet data from the network traffic while the individual move the recoater blade, dispenser platform, building platform. A secondary set of data is useful to cross check the first print to verify the analysis. Ultimately, the identified response data from the components will be known to the specific components used in the printers at a particular time. Additionally, my hardware connection of the spectrometer will aid the in providing quality results to determine if spectroscopy-based in-situ monitoring would be useful for quality assurance in metal AM processes [8]. Therefore, evaluate how the elemental composition of the powder can be determined by the optical spectrum from its plasma emission when the powder is excited by a laser.

6. Gantt Chart

Project Planning Chart



Project Planning Chart



Due to the nature of the multiple projects taken over the spread of 22 weeks, the activities presented in the project planning chart will be overlapped. The activities are noted with descriptions. The 4 key projects are the Feasibility Study of Integrating Sensors to 3D Printer for Conditional Monitoring, M5Stack FIRE IoT Development, Webcam Streaming in EOS M 290 Chamber, and Spectrometer Integration and Development.

Preliminary discussions on proposed projects – the projects are discussed to be in line with the scope of the internship equivalent -FYP project title. The scope, purpose and constraints are listed prior so as to mapped out the required timeline for each projects as multiple projects are running concurrently.

Research and information gathering – the 4 projects are within the same scope but encompasses different skillsets from 3D modelling, writing scripts with Python and C++. Hence, to build up my knowledge, I constantly research to help aid in applying the required skills to contribute to the projects done.

Study on material availability & machining capability etc – 3D printing with PLA is chosen for fast prototyping of useful parts. Little machining capability needed.

Detailed drawings, fabrication procedures & material listing – detailed drawing produced is for the secure holder for PCB with Separator board. The components used are not made in-house hence limited fabrication procedures.

Material purchases, fabrication, and construction – the purchases consists of various modified PCBs for the EOS M 290 machine, and optic fiber patchcord, collimator lens. These were bought from AliExpress which have a shipping time of a month

Assembly / installation & Testing and commissioning – the physical assembly and testing activities are related to the feasibility study and the webcam streaming

Final modification adjustment & touching up – the feasibility study and webcam streaming projects require no further adjustments. The M5 stack IIoT device was revisited once more to refurbished the code in GitLab.

Report writing, Seminar & Project viva – both are planned earlier as to consolidate the projects done such as the feasibility study of the integration of sensors to the 3D printer for conditional monitoring, M5 stack IIoT device, webcam streaming and development of the spectrometer.

Conclusions

In the past 22 weeks, I have successfully conceived, designed, implement, and operate the sensor integration system for metal 3D printing and powder spreading equipment. These efforts have the potential to greatly enhance the efficiency and accuracy of these processes. Through the integration of sensors, real-time monitoring of critical parameters can be performed. This ability to monitor in real-time allows for immediate adjustments to be made, resulting in improved precision and reduced waste. Furthermore, the use of sensors provides valuable data that can be used for process optimization and quality control purposes. The wide flexibility of the using open-source products and software in the integration is an added advantage to furthering the cutting edge of metal additive manufacturing industry.

The industry-focused and research exposure have taught me multiple skillsets from fields of mechanical, materials, software, and system engineering. The hard skills learnt and applied are computer programming languages (Python and C++), electrical and mechanical systems, understanding of project management methodologies and the ability to use data analysis tools and techniques. Working together with other interns and colleagues from other disciplines build up my soft skills with effective communication, collaboration and critical thinking. Additionally, supplementary activities such as material characterization and operating the metal 3D printers enabled me to understand the projects' environments and requirements better. These experiences will provide me a competitive edge to work effectively in a multi-disciplinary team and to keep up with the industrial challenges of today and tomorrow.

References

[1] Yaojie Wen , Baicheng Zhang , Shuying Liu , Wei Cai , Pei Wang , Coryl Jing Jun Lee , Junda Ma & Xuanhui Qu (2020) A novel experimental method for insitu strain measurement during selective laser melting, Virtual and Physical Prototyping, 15:sup1, 583-595, DOI: 10.1080/17452759.2020.1842137

[2] Xie, D., Lv, F., Yang, Y., Shen, L., Tian, Z., Shuai, C., Chen, B., & Zhao, J. (2022). A Review on Distortion and Residual Stress in Additive Manufacturing. In Chinese Journal of Mechanical Engineering: Additive Manufacturing Frontiers (Vol. 1, Issue 3, p. 100039). Elsevier BV. https://doi.org/10.1016/j.cjmeam.2022.100039

[3] Hehr, A., Norfolk, M., Kominsky, D., Boulanger, A., Davis, M., & Boulware, P. (2020). Smart Build-Plate for Metal Additive Manufacturing Processes. In Sensors (Vol. 20, Issue 2, p. 360). MDPI AG. https://doi.org/10.3390/s20020360

[4] R., Spurgeon, T. J., & Reutzel, E. W. (2014). Sensing Defects during Directed-Energy Additive Manufacturing of Metal Parts using Optical Emissions Spectroscopy. University of Texas at Austin. https://doi.org/10.26153/TSW/15684

[5] L. Song and J. Mazumder, "Real Time Cr Measurement Using Optical Emission Spectroscopy During Direct Metal Deposition Process," in IEEE Sensors Journal, vol. 12, no. 5, pp. 958-964, May 2012, doi: 10.1109/JSEN.2011.2162316.

[6] Brandau, B. (2022). Spectral analysis in laser powder bed fusion (Licentiate dissertation, Luleå tekniska universitet). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-90525

[7] Kouprianoff D, Yadroitsava I, du Plessis A, Luwes N and Yadroitsev I (2021) Monitoring of Laser Powder Bed Fusion by Acoustic Emission: Investigation of Single Tracks and Layers. Front. Mech. Eng 7:678076. doi: 10.3389/fmech.2021.678076

[8] Lijun Song, Cunshan Wang, Jyoti Mazumder, "Identification of phase transformation using optical emission spectroscopy for direct metal deposition process," Proc. SPIE 8239, High Power Laser Materials Processing: Lasers, Beam Delivery, Diagnostics, and Applications, 82390G (6 February 2012); doi: 10.1117/12.908264

Appendix 1:

Plagiarism Declaration Form

(This form is to be signed by the student, scanned, and inserted back into the report in pdf format. The original hard copy form will be attached in the hard copy version of the report)

I am aware of the Singapore Polytechnic penalty for plagiarism, which is that if I am guilty of plagiarism in one module I will fail **ALL** my modules in the semester, or even be liable for expulsion. I understand that plagiarism means passing off another person's work as my own, without crediting the person.

I declare that the work I am submitting is my own. I have acknowledged all sources of information referenced.

Module Code: IE5003

Submission Date: 2ND FEBRUARY2023

Clement Toh

Signature

Student(s) Name: TOH EN RUI CLEMENT

Student number: 2034135

Appendix 2:

Confidentiality Declaration Form

(This form is to be signed by the student and company staff, scanned, and inserted back into the report in pdf format. The original hard copy form will be attached in the hard copy version of the report)

I declare that my report has been vetted by the company, and the contents found in this report has been cleared by the company.

Module Code: IE5003

Submission Date: 2ND FEBRUARY2023

Clement Toh

Student Signature

Student(s) Name: TOH EN RUI CLEMENT

Student number: 2034135

I acknowledge that the contents of the report have been cleared by the company. The student is allowed to share his report in the SP database.

Company Stamp/Signature

Company Supervisor Name: Tan Zheng Jie

Designation: Development Scientist