



UNIVERSITY OF  
OREGON



Developing Melt Electrowriting on an Open-Source  
Fused Filament Fabrication Platform

a master's thesis authored by

Taavet Kangur

and supervised by

Professor Paul Dalton

Professor Christophe Moser

Professor Jürgen Brugger

7 of February, 2023

# Acknowledgements

I would like to thank Professor Moser and Professor Brugger for their supervision and for providing me this great opportunity. I would like to express my deep and sincere gratitude to Professor Dalton for his continuous support during the tenure of this project. I am very thankful to him for creating a research friendly environment in the Dalton lab which helped me to learn, grow and build friendships. The completion of this project wouldn't have been possible without Simon Luposchinsky whose help was instrumental to my research. I'd like to acknowledge the great help given by Dr. Ievgenii Liashenko for assisting with imaging and the invaluable advice on troubleshooting the problems with MEW. I would like to give my sincere thanks to Dr. Biranche Tandon for the long discussions we had regarding different parameters for MEW.

Finally, I'm extremely grateful to all the people of the Phil and Penny Knight Campus for welcoming me among themselves during this six-month thesis and making my first visit to the United States as memorable as possible through all the time we spent together.

## **Abstract**

Melt electrowriting (MEW) is an additive manufacturing technology where a fiber is deposited in a precise way with the help of an electric field to build micrometric designs for tissue engineering and regenerative medicine (TERM). The commercially available MEW systems currently are expensive and require fine tuning to match the scaffold quality of custom built systems and, as a result, many of the research institutions choose to opt for the latter. This thesis presents a way to reduce the cost of MEW systems by using an open source Fused Deposition Modelling (FDM) printer, converting it into a MEW system to make good quality high-resolution scaffolds. The printer needs additional parts such as a high voltage power supply (HVPS) and different components for high voltage and flow rate control. The resulting scaffold produced by the filament feed MEW system is shown to have good stacking and good dimensional accuracy of the fibers, which was close to the calculated values.

**Keywords:** additive manufacturing, melt electrowriting, open source

# Contents

<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Abbreviations</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context . . . . .	1
1.2 Problem Statement . . . . .	1
1.3 Motivation . . . . .	2
1.4 Objectives . . . . .	2
<b>2 State of the Art</b>	<b>4</b>
2.1 Fused Deposition Modelling . . . . .	4
2.1.1 Cost & availability . . . . .	6
2.1.2 Materials . . . . .	6
2.1.3 Software . . . . .	7
2.2 Melt Electrowriting . . . . .	8
2.2.1 Cost & availability . . . . .	10
2.2.2 Materials . . . . .	11
2.2.3 Software . . . . .	11
2.2.4 Current research & applications . . . . .	12
2.2.5 Challenges . . . . .	12
2.3 FDM and MEW . . . . .	13
2.4 G-code . . . . .	14

2.5	Open Source . . . . .	15
2.5.1	Reprap project . . . . .	15
2.5.2	The Voron Project . . . . .	15
2.5.3	Software . . . . .	18
<b>3</b>	<b>Methodology</b>	<b>20</b>
3.1	Materials . . . . .	20
3.2	Theory . . . . .	21
3.2.1	Melt Electrowriting . . . . .	21
3.2.2	Filament flow equations . . . . .	21
3.2.3	Volume of molten material . . . . .	24
3.3	Hardware . . . . .	25
3.3.1	Available parts . . . . .	25
3.3.2	Extrusion resolution and consistency . . . . .	26
3.3.3	Glass slides . . . . .	26
3.3.4	High voltage . . . . .	26
3.3.5	High voltage control . . . . .	28
3.3.6	Change of nozzle . . . . .	30
3.3.7	Higher gear ratio motor . . . . .	30
3.3.8	Environmental measurements . . . . .	31
3.3.9	FDM conversion to MEW . . . . .	32
3.4	Software . . . . .	34
3.4.1	G-code generation . . . . .	34
3.4.2	Extrusion control . . . . .	35
3.5	Imaging tools and measurements . . . . .	36
3.5.1	Optical microscope and SEM . . . . .	36
3.5.2	Recording of data . . . . .	36
3.5.3	Generation of graphs . . . . .	36
<b>4</b>	<b>Results and discussion</b>	<b>37</b>
4.1	Achieving MEW on the system . . . . .	38
4.1.1	Testing the polymer . . . . .	38

4.1.2	Testing the motion system . . . . .	39
4.1.3	Nozzle design . . . . .	40
4.1.4	Microsteps of the stepper motor . . . . .	41
4.1.5	Skipping of the fiber . . . . .	41
4.2	Resulting scaffolds and comparison . . . . .	42
4.2.1	Fiber diameter linked to extrusion speed . . . . .	43
4.2.2	Fiber spacing . . . . .	44
4.2.3	Comparison with state of the art MEW system . . . . .	45
4.2.4	Voltage control . . . . .	49
4.2.5	Environmental monitoring . . . . .	50
4.2.6	Price of the system . . . . .	51
<b>5</b>	<b>Conclusions</b>	<b>52</b>
<b>6</b>	<b>Future work</b>	<b>53</b>
6.1	Computer vision . . . . .	53
6.2	Tubular collector . . . . .	54
6.3	Reducing the price of the high voltage power supply . . . . .	54
6.4	Glass slide dispensing . . . . .	55
6.5	High Voltage protection . . . . .	55
6.6	Additional polymers . . . . .	55
6.7	FDM and MEW . . . . .	56
	<b>References</b>	<b>57</b>
	<b>Appendices</b>	<b>A1</b>
<b>A</b>	<b>G-code generator parameters</b>	<b>A2</b>
<b>B</b>	<b>G-code generator code in MATLAB</b>	<b>A3</b>
<b>C</b>	<b>Technical Drawings</b>	<b>A14</b>

# List of Figures

2.1	The FDM patent in 1989 [9] . . . . .	4
2.2	Illustration of a hotend [10] . . . . .	5
2.3	Two of the most common FDM printers on the market: The Prusa MK3S+ (left) and the Creality Ender 3 (right) [14] [15] . . . . .	6
2.4	The FDM process chain [20] . . . . .	7
2.5	Schematic of MEW process [23] High voltage is applied to the static print head while the collector is grounded and moves . . . . .	8
2.6	Illustration of jet lag [23] (A) Schematic of jet lag with low and high jet lag. (B left) Programmed G-code path (B right) Result of programmed path with jet lag until stabilization . . . . .	9
2.7	Effect of lowering collector speed below CTS on fiber pattern (A) 100-110% CTS, (B) 75-80% CTS, (C) 30-35% CTS, (D) 10-15% CTS [23] . . . . .	10
2.8	Example of a MEW system: The Queensland University of Technology MEW system . . . . .	11
2.9	The first Voron prototype on the 20th May 2020 [32] . . . . .	15
2.10	The currently developed Voron models, from left to right : V1, V2, V0, Switchwire, Legacy [34] . . . . .	16
2.11	The Voron 0.1 FDM printer [34] . . . . .	17
2.12	Schematic of CoreXY kinematics [35] . . . . .	17
2.13	OctoPrint: View of the temperature and camera feed [36] . . . . .	18
3.1	Effect of pulsing in the jet on deposited fiber The fiber thickness varies during the printing and affects the deposition . . . . .	21

3.2	Most of the components required for a Voron 0.1. Left: Extrusions and large components, Middle: Fasteners, belts and electronics, Right: All required FDM parts . . . . .	25
3.3	Spruce Science’s lab mate [39] . . . . .	27
3.4	Left : TSH82 configuration, right : complete high voltage control board . .	29
3.5	Left : Standard 0.25mm FDM nozzle, right: custom nozzle . . . . .	30
3.6	Tapped hole in the build plate to attach high voltage cable . . . . .	32
3.7	Hole in the back of the Voron for high voltage cable . . . . .	32
3.8	The 3D printed parts required for the conversion Left : high voltage cable holder, right : the motor adaptor . . . . .	33
3.9	Left: original print head, right: custom print head . . . . .	33
3.10	The resulting G-code path with stabilization paths . . . . .	34
3.11	The loops generated by the generator . . . . .	35
4.1	Image of the Voron 0.1 FDM system converted to MEW . . . . .	37
4.2	Process to melt filament to syringe to be used in MEW printer and resulting scaffold. Left to right: Filament pieces in syringe, Syringe with the molten polymer after heating and resulting scaffold . . . . .	38
4.3	5 layer MEW scaffold made with pressure fed system on Voron printer . . .	39
4.4	Same parameters MEW scaffolds with different nozzles Left : made with standard 0.25mm FDM nozzle, right: made with custom nozzle . . . . .	40
4.5	Cold pull of the filament from the nozzle shows the large 120° taper . . . .	41
4.6	10 layer MEW scaffold with 7um fibers and 0.5mm spacing The parameters used are in Table 4.1 . . . . .	42
4.7	Fiber diameter in function of extrusion speed . . . . .	43
4.8	Fiber spacing : Theoretical and experimental . . . . .	44
4.9	MEW scaffolds Left: made on QUT, right: made on Voron . . . . .	45
4.10	SEM pictures of fibers Left on QUT Printer, right on Voron . . . . .	46
4.11	MEW loops Left on QUT Printer, right on Voron . . . . .	47
4.12	Stacking of fibers Left on QUT Printer, right on Voron . . . . .	48
4.13	Distance of 3.2mm and 4.8kV . . . . .	49



4.14	+0.01mm after each layer . . . . .	49
4.15	+10V after each layer . . . . .	49
4.16	+0.01mm and +10V after each . . . . .	49
4.17	Environmental measurements on OctoPrint . . . . .	50
6.1	Attempt to view the jet. From left to right: View of the nozzle from an attached boroscope, raw image and processed image . . . . .	53
6.2	Left: Tubular scaffolds, right: mandrel with motor . . . . .	54
6.3	Peta-pico-Voltron: An open-source high voltage power supply [41] . . . . .	54
6.4	A number of control boards were electrocuted . . . . .	55
C.1	Electric schematic for the high voltage control board . . . . .	A15
C.2	Schematic of the machined nozzle for MEW . . . . .	A16

# List of Tables

2.1	Comparison of FDM and MEW . . . . .	13
2.2	List of commonly used G-codes in FDM . . . . .	14
2.3	Overview of Voron models . . . . .	16
3.1	Characteristics of Poly-Med's Caprorene PCL [38] . . . . .	20
3.2	The Voron 0.1's extrusion system parameters [33] . . . . .	22
3.3	Variation in the flow rate depending on the filament diameter . . . . .	22
3.4	Components of the voltage control board and their function . . . . .	28
3.5	Some of the parameters to define the scaffold design . . . . .	34
4.1	Parameters used to MEW scaffolds on the Voron The resulting scaffold is in Figure 4.6 . . . . .	42
4.2	Fiber diameter inspection results . . . . .	43
4.3	Fiber spacing inspection results . . . . .	44
4.4	List of all the components of the modified Voron . . . . .	51
A.1	List of all the parameters to define a scaffold . . . . .	A2

# List of Abbreviations

**ABS** Acrylonitrile butadiene styrene

**AM** additive manufacturing

**CNC** Computer Numerical Control

**CTS** critical translation speed

**ES** electrospinning

**FDM** Fused Deposition Modelling

**HVPS** high voltage power supply

**MES** melt electrospinning

**MEW** melt electrowriting

**PCL** Poly  $\epsilon$ -caprolactone

**PLA** Polylactic acid

**TERM** tissue engineering and regenerative medicine

# Chapter 1

## Introduction

### 1.1 Context

Melt electrowriting (MEW) is a high-resolution additive manufacturing technique and is commonly used for research in tissue engineering and regenerative medicine (TERM) and soft robotics [1]. MEW can be utilized to make planar or tubular scaffolds with predefined micrometric features designed with the purpose of mimicking the properties of different tissues [2]. The most common use of these scaffolds in TERM is to guide cell growth by tailoring their environment and support the regeneration of damaged tissues [3], [4]. As a temporary structure in the body, its degradation is a key factor and depends on the polymer used [5].

### 1.2 Problem Statement

Research in TERM is expensive partly due to the regulations, the studies and the tools required [6]. The prices of systems for TERM vary depending on their capabilities but start at \$20,000 and can cost more than \$300,000 range [7]. Many research laboratories choose to build their own systems as the cost of different high quality components such as stages, temperature regulation tools and high voltage power supplies is less than the overall cost of a commercial system. In addition, the commercial systems are proprietary and any modification on the device risks voiding their warranty. This slows the development of MEW as research institutes have to collaborate with the system manufacturers.

Committing to a commercial system often limits the capabilities of the research group to that of the commercial system.

### 1.3 Motivation

Research in MEW requires many iterations of a design to fine tune the printing parameters and find the appropriate design for an intended purpose. Designing a MEW system which is less expensive would have the benefit of lowering the budget making it more accessible to the TERM community. It would also increase the throughput by allowing the possibility of having multiple systems working in parallel. Improved accessibility to a MEW system would mean more research groups using the technology, sharing knowledge and making further advances in the TERM field. A community of researchers using the same printer could improve the repeatability of experiments and lead to opportunities for further collaborations between laboratories by simply sharing a scaffold design along with the used parameters.

Using an open-source design to build MEW systems has many advantages such as making the printers easily repairable, customizable and upgradable. Such an approach also ensures that different components are easily available and are cost effective. With a Fused Deposition Modelling (FDM) system capable of MEW, the advances done in the FDM field over the last decade can be leveraged for MEW. Designing a less expensive system would increase the number of users and could possibly make the components for MEW such as high voltage power supply (HVPS) more common and less expensive. Furthermore, such an approach could potentially bring MEW to the same kind of forward momentum as FDM after its patent expiration in 2009.

### 1.4 Objectives

The primary objective of this work is to convert a FDM system into a MEW system capable of producing good quality scaffolds. This entails changing the standard pressure-fed polymer system used in MEW with a direct drive used in FDM to push filament. The parameters utilized for quality assessment of scaffolds are the fiber consistency, fiber placement accuracy and correct stacking of the fibers. The printed design using the mod-

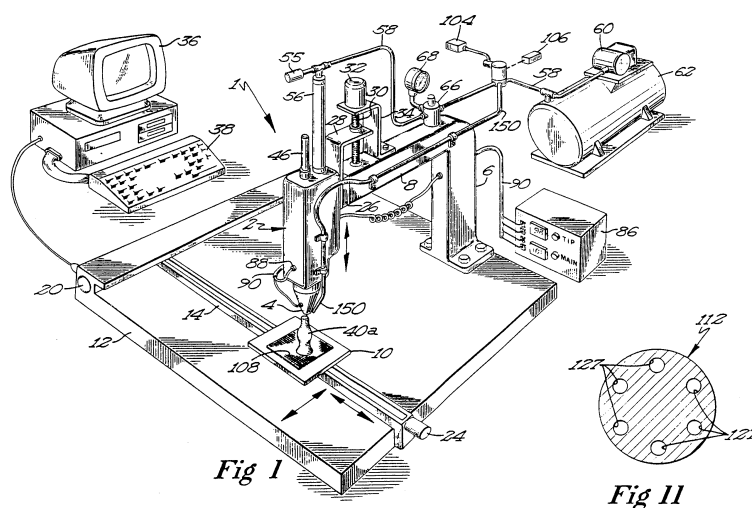
ified FDM printer will be compared to the same scaffold design made on a state of the art MEW system. The additional objectives are to automate voltage control on the newly built system and to keep the cost of the converted system under \$2000.

# Chapter 2

## State of the Art

### 2.1 Fused Deposition Modelling

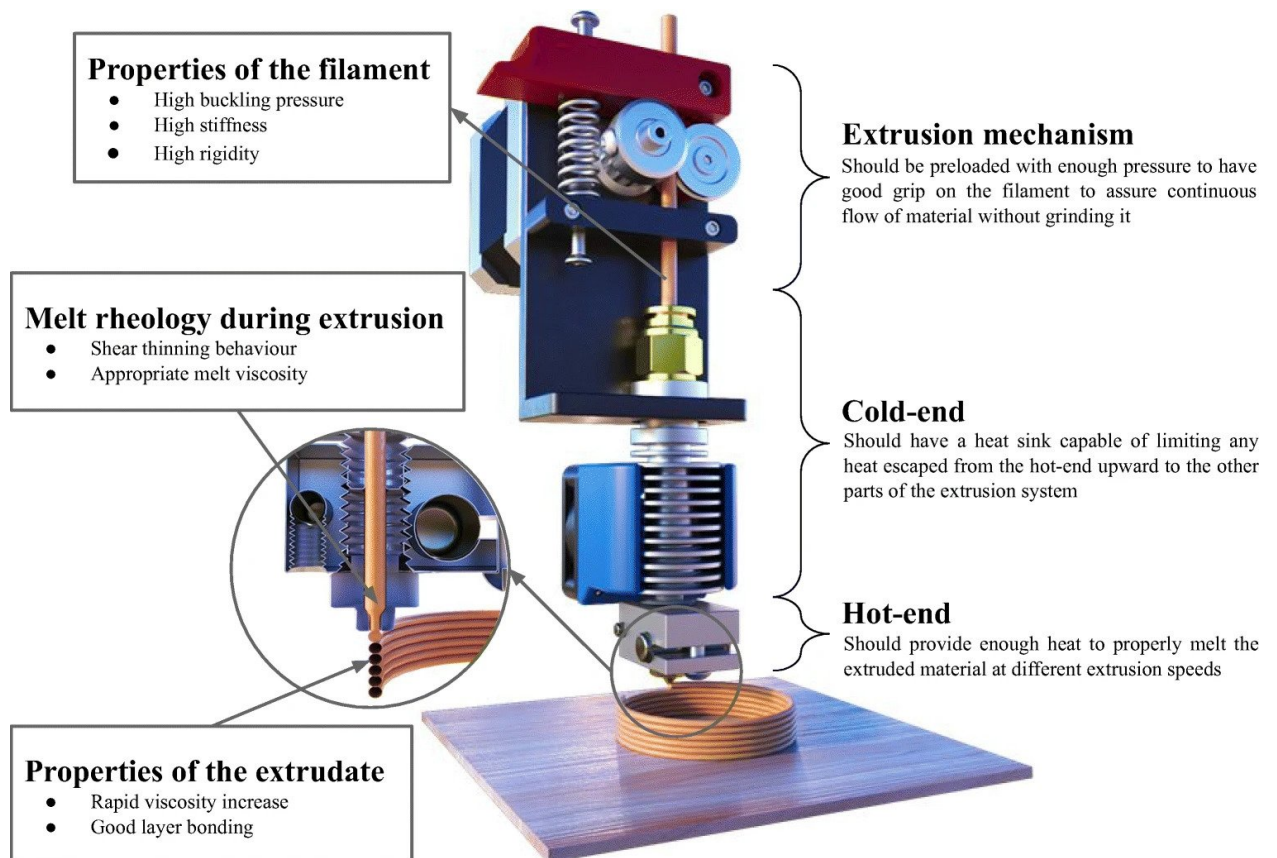
Fused Deposition Modelling is an additive manufacturing technique where a thermoplastic material is extruded through a nozzle onto a build plate. The 3D object is created line by line and stacking layers upon layers of the material [8]. It was developed by Scott Crump in 1988 [9]. The technology, illustrated in Figure 2.1, was applied to be patented in 1989 and Stratasys was founded.



U.S. Patent  
June 9, 1992  
Sheet 1 of 3  
5,121,329

**Figure 2.1**  
*The FDM patent in 1989 [9]*

The term "FDM" was trademarked by the company for their system. Fused filament fabrication is the non trademarked term for FDM but both are interchangeable. The feed system for the extrusion of the material is a motor connected by a gear train to a hobbed gear which pushes a filament of the desired material through a temperature controlled hotend to reach its melting temperature [10] The whole system is illustrated in Figure 2.2. The molten polymer is then extruded through a nozzle [8]. The flow rate of the polymer is controlled by the speed of the motor and is in the range of 1-10  $mm^3/s$  [11].

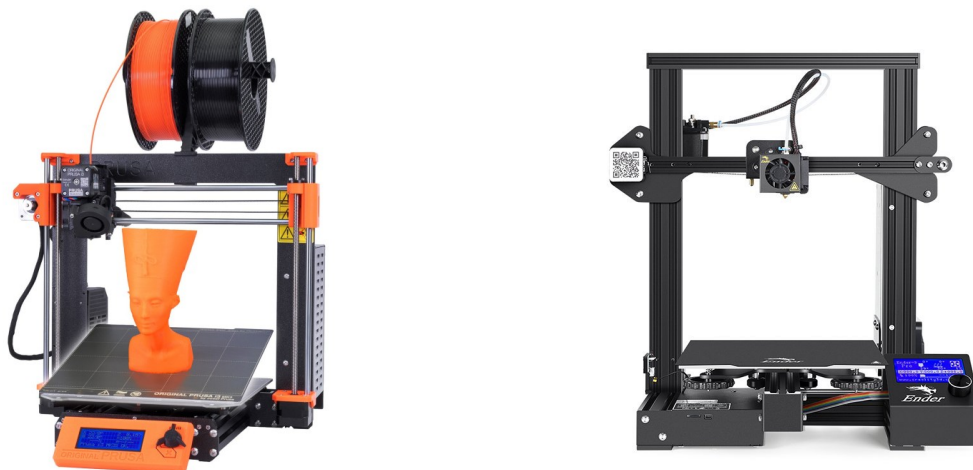


**Figure 2.2**  
*Illustration of a hotend [10]*



### 2.1.1 Cost & availability

After the expiry of the FDM patent in 2009, the technology was replicated many times. In the recent years, the price of FDM printers has lowered drastically. In 2022, a basic FDM printer can cost as low as \$200 with prices going up to \$200,000 for large scale industrial printer with advanced capabilities [12]. FDM, with its lowered price, has become one of the most accessible additive manufacturing techniques and the default one when talking about 3D printing [13]. Two of the most common printers are shown in Figure 2.3. With their price, availability and economy of scale, the individual parts of FDM printers have become less expensive.



**Figure 2.3**

*Two of the most common FDM printers on the market:  
The Prusa MK3S+ (left) and the Creality Ender 3 (right) [14] [15]*

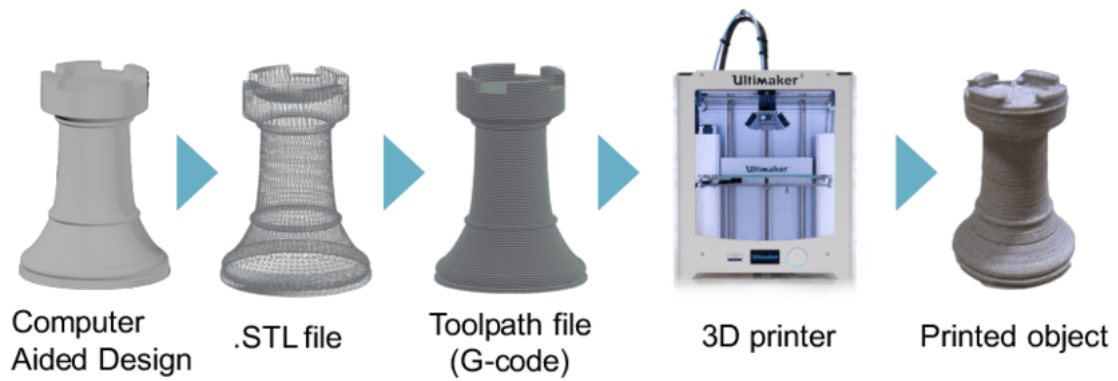
### 2.1.2 Materials

FDM is generally performed with thermoplastics and one of the main requirements of the printer is its suitability for the range of temperatures required for the material. The most commonly used material was Acrylonitrile butadiene styrene (ABS) as it was readily available in form of plastic welding rods. Though ABS has good properties, it warps while printing and releases Styrene when heated, which is toxic. Polylactic acid (PLA) became the standard material as it has similar high printability but releases no toxic fumes. Other materials such as Nylon, PET, PETG, PP, PEEK and composites can be processed in

FDM depending on the system capabilities [16]. The range of temperatures used in FDM to melt the polymer is 150-430°C and the material is usually in a molten state only for seconds before it is pushed on to the build plate and re-solidifies [17], [18].

### 2.1.3 Software

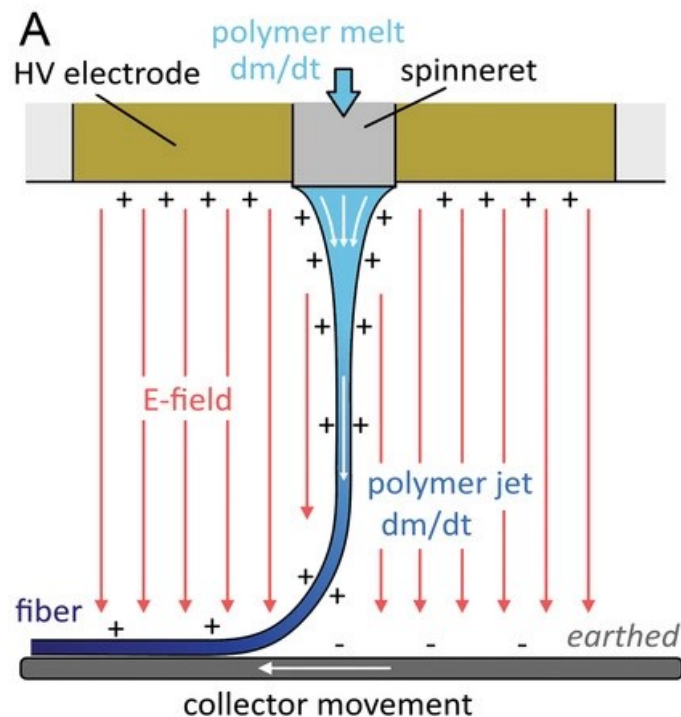
The steps to make a FDM printed object begins with its modelling in a Computer Aided Design (CAD) software or downloading a 3D model from a website such as Thingiverse [19]. Software called a slicer is then used to convert the 3D model into layers and paths written in G-code which the FDM printer interprets (Figure 2.4). The settings in the slicer have to be specific to the FDM system and the material used [20].



**Figure 2.4**  
*The FDM process chain [20]*

## 2.2 Melt Electrowriting

Melt electrowriting is an additive manufacturing technique which uses an electric field to attract and stabilize the melt flow of a polymer to create micrometer size fibers as shown on Figure 2.5. The first paper on MEW was published in 2011 by Brown et al. [21] in an attempt to create defined structure using melt electrospinning (MES). The technology was developed from advances made in electrospinning (ES) which uses a polymer in a solvent to spray sub micrometer fibers in an uncontrolled pattern over an area. MES uses the same concept but without the use of solvents and creates larger fibers in an uncontrolled pattern. By reducing the distance between the collector and spinneret, and lowering the voltage to 3-10 kV, the extruded polymer becomes a steerable jet whose properties can be further optimized by tuning the processes parameters [22].



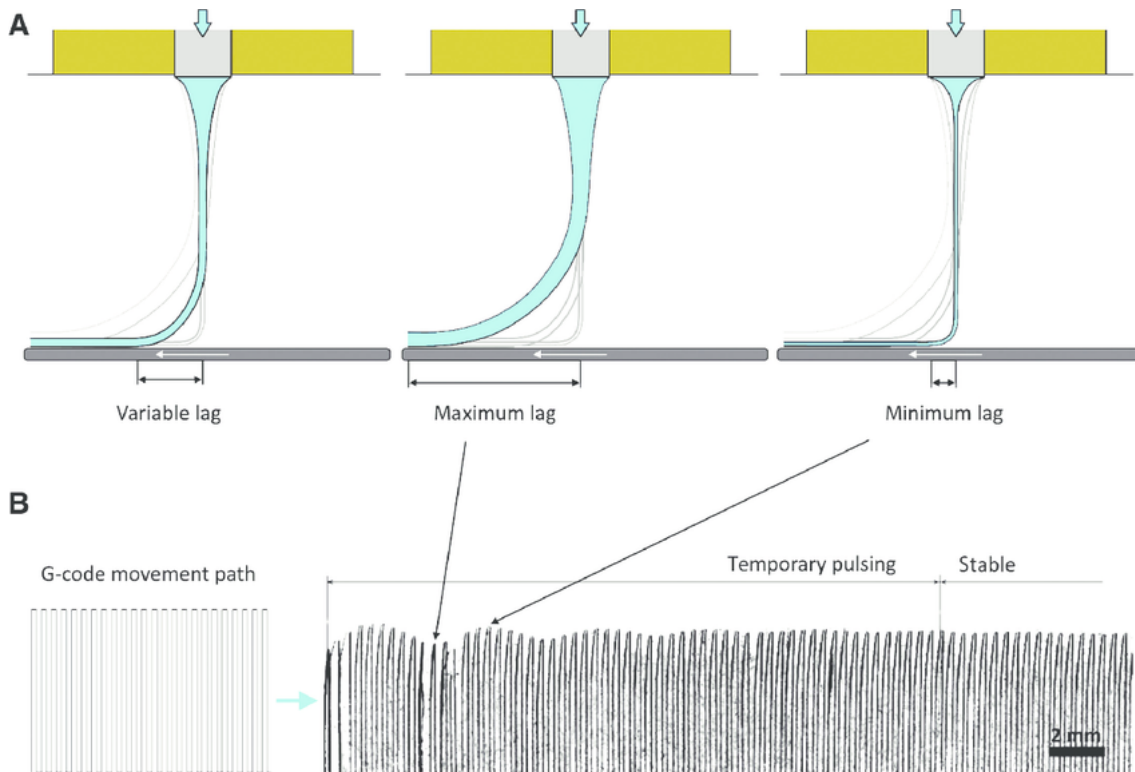
**Figure 2.5**

*Schematic of MEW process [23]*

*High voltage is applied to the static print head while the collector is grounded and moves*

Many parameters affect the jet and need to be tuned to successfully MEW fibers with specific dimensions. Polymer temperature, voltage level, collector distance and collector speed, to name a few, are all interdependent. Adjustments need to be made to reach the

narrow range where the jet becomes and stays stable for the multiple hour print times commonly required by MEW [22].



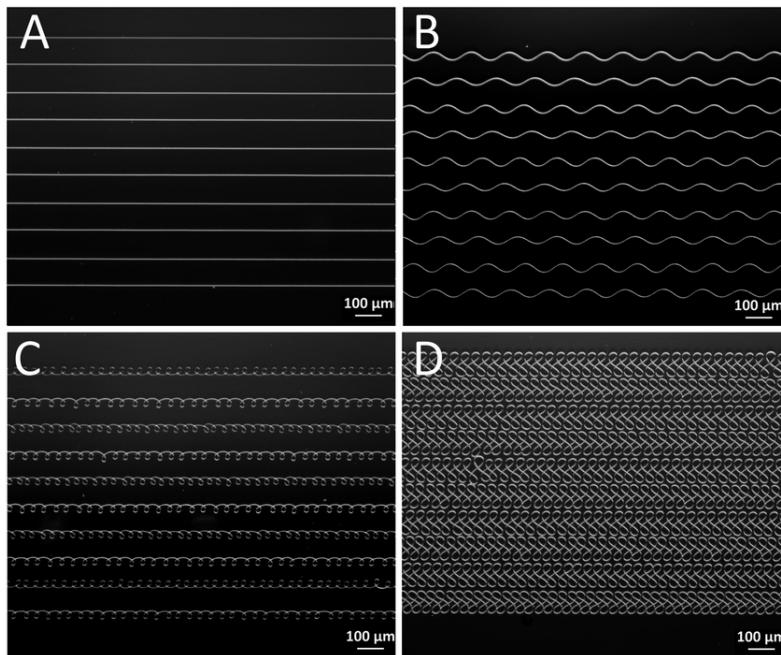
**Figure 2.6**

*Illustration of jet lag [23]*

*(A) Schematic of jet lag with low and high jet lag. (B left) Programmed G-code path  
(B right) Result of programmed path with jet lag until stabilization*

The nozzle is at a distance from the build place and as the print head moves, the jet is trailing behind it. The horizontal distance between the nozzle and the position the jet touches the build place is called jet lag. It is dependent on flow rate, applied voltage, collector speed, distance to collector and viscosity of the polymer. An illustration of the jet lag can be seen in Figure 2.6. The effects of jet lag on the designed pattern are seen as rounding of sharp corners and change in the laydown pattern [24]. When the jet has a minimal jet lag and is following the movement of the collector closely, it has reached its critical translation speed (CTS). If the speed of the collector is lower than the CTS, the jet will start buckling like in Figure 2.7. If the speed is higher, the jet lag will increase and if it is multiple times the CTS, the fibers will be stretched.

The material feeding system in MEW is a heated syringe with a specific nozzle size and filled with a polymer. The syringe is heated until the viscosity of the polymer is low enough



**Figure 2.7**

*Effect of lowering collector speed below CTS on fiber pattern*

*(A) 100-110% CTS, (B) 75-80% CTS, (C) 30-35% CTS, (D) 10-15% CTS [23]*

to be fed through the nozzle by applying air pressure. As the whole reservoir of polymer is heated, the degradation of the polymer needs to be slow otherwise the polymer would degrade too much and the flow rate would be affected resulting in fibers of low quality and varying in size. The flow rates of MEW are in the range 0.5-20  $\mu\text{L}/\text{h}$  depending on the fiber size needed [25].

### 2.2.1 Cost & availability

The price of a MEW system can vary but usually start at \$20,000 and can cost more than \$300,000. As it is used as a research tool for bioengineering, some systems combine other additive manufacturing (AM) technologies such as MES and inkjet to make the system more attractive for research institutes. Most of the commercial MEW systems are not off-the-shelf and require the manufacturing company to be contacted to get a quote on the price of the system which both slows the research and locks it to the capabilities of the bought system. One of these MEW systems is shown in Figure 2.8.



**Figure 2.8**  
*Example of a MEW system:  
The Queensland University of Technology MEW system*

### 2.2.2 Materials

The most utilized polymer for MEW is Poly  $\epsilon$ -caprolactone (PCL). Several other thermoplastics such as PLLA, PVDF, PP and more have been tested [25]. The general requirement for the polymer is that the heat required to reduce the viscosity to MEW is sustainable for long periods of time for the system and that the polymer doesn't degrade too much. The system is the main limiting factor as the components must sustain and contain the amount of heat generated to melt the polymer completely.

### 2.2.3 Software

Software to generate scaffold designs by entering the required parameters exist but the most common way to design them is to write the G-code by hand. This can become a tedious task if a design needs minor changes but the whole file must be written again. The use of programming loops in G-code for scaffolds, if the printing system is capable of reading them, is common as scaffolds have repeating patterns.

#### **2.2.4 Current research & applications**

Research in MEW has increased with the development of commercial systems and the interest in TERM. From potential stroma substitute [26] and customizable porosity heart valve [27] to improvement in skin regeneration [28] and its help in dental medicine [29], MEW has yet to reveal all its potential in the TERM field. The ability to create tailored scaffolds without the need to solvents or additives gives a much better chance of translating a technology to clinical studies.

#### **2.2.5 Challenges**

The price required to buy or build a MEW system currently starts around the \$20,000. Only having one capable machine is a start but making MEW scaffolds is a long process lasting multiple hours if not days and studies in tissue engineering require multiple samples to be tested at a time. The current standard material used for MEW is PCL for its low melting temperature and several studies have used PCL for TERM. The limiting factors in using other polymers is the system's ability to withstand high temperature for multiple hours, as well as the system's ability to create a controlled environment to MEW the polymer.

## 2.3 FDM and MEW

FDM has had a longer development time than MEW and since the expiry of the patent, a large community of developers, engineers and companies have invested time into the technology and made it faster, more intuitive, and less expensive. Table 2.1 summarizes the differences between the two additive manufacturing techniques.

	<b>FDM</b>	<b>MEW</b>
Flow rate [ $\mu\text{L}/\text{h}$ ]	3,600 - 36,000	0.5-20
Temperature ranges [ $^{\circ}\text{C}$ ]	150 - 430	60 - 120
Material feed system	Pushing filament	Pressurized syringe
Movement speed [ $\text{mm}/\text{min}$ ]	1,800 - 4,800	300 - 500
Amount of heated material at one time [ $\mu\text{L}$ ]	30 - 50	500 - 1000
Language for movement	G-code	G-code
Resolution [ $\mu\text{m}$ ]	100	5
Price range [ $\text{\$}$ ]	200 - 200,000+	20,000 - 300,000+

**Table 2.1**  
*Comparison of FDM and MEW*

Some important elements to point out are that the flow rate of MEW is three orders of magnitude lower than the one of FDM and the material feeding system is different. Pushing filament with a motor and pressurizing a syringe, both have their advantages and problems in terms of polymer degradation and flow consistency.



## 2.4 G-code

G-code is a Computer Numerical Control (CNC) programming language created to be used on machine tools such as CNC mills and lathes with specific commands to also change tools and rotations speeds. The first implementation was developed at the MIT servomechanisms Laboratory in the late 1950s [30]. It is a human-readable type of text file with a list of simple commands for the system to execute one line after another. A mistake in the g-code generation can be noticed if familiar with the g-code generator. The main commands which are used in the case of FDM are listed in Table 2.2. As G-code can be read and understood easily, the raw file can be edited to modify movements, speeds or temperatures quickly.

G0 : Linear motion
G1 : Linear motion by convention with extrusion
G2 : Clockwise arc
G3 : Counter - clockwise arc
G4 : Dwell \wait in milliseconds
G28 : Home all axes
G90 : Absolute positioning
G91 : Relative positioning
G104 : Set hotend temperature

**Table 2.2**  
*List of commonly used G-codes in FDM*

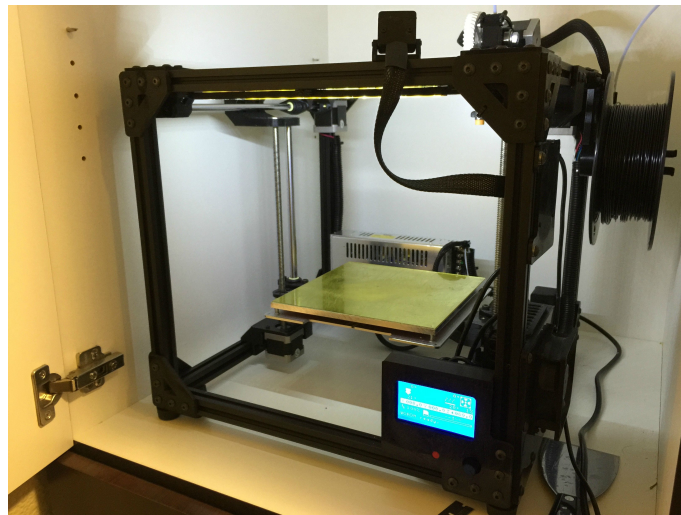
The command G1 is specifically used in Fused Deposition Modelling for movement with extrusion. Having the printer in relative positioning and sending a command such a G1 X10 Y10 Z10 E10 F200 would make the head move +10mm in X, +10mm in Y, +10mm in Z at a speed of 200mm/min and extrude 10mm of filament during the whole movement. This type of extrusion makes the flow dependent on the speed of the movement and thus, makes the pressure applied in the nozzle vary between each movement during a print.

## 2.5 Open Source

### 2.5.1 Reprap project

The Reprap project, **Replicating Rapid-prototyper**, was initiated by Adrian Bowyer at Bath University with the idea appearing online in February 2004. The goal of the project is to make an open source FDM printer with most of its parts off-the-shelf, such as threaded rods and nuts, and some of its parts FDM printed. The printed parts could be made by another Reprap machine and so the notion of machine self-replication came to be. The only obligation is to openly release any improvements of the machine to the community [31].

### 2.5.2 The Voron Project



**Figure 2.9**

*The first Voron prototype on the 20th May 2020 [32]*

The Voron project was started by Maksim Zolin in April 2015 with the idea of creating a no compromise FDM printer. During the project, he built multiple working prototypes, the first one is shown in Figure 2.9, until one finally met the requirement he had set. With the growth of FDM, the race to bring out the best printer at the lowest price and the hassle of the supply chain logistics of making a hardware company while having a day job, he decided to open-source the project and build a community around it.

This led to a large influx of people interested in the new printer and many new models and improvements from the community [33]. The team working on the Voron project grew

and they kept improving the printers, updating the hardware list with quality components they test themselves and managing the community around Voron printers. All the current models of Vorons are shown in Figure 2.10 and their characteristics are listed in Table 2.3.

A large sourcing guide is provided with up-to-date links to components for the different Vorons. All the parts can be bought off the shelf from different suppliers. Some companies have started to make kits of all the parts to make it easier for the user to purchase and assemble them. These kits are sold at a premium but make the building of the printer easier.

Model	V1	V2	V0	Switchwire	Legacy
<b>Version number</b>	9	4	1	0	0
<b>Conception date</b>	2015	2019	April 2020	August 2020	December 2020
<b>Price [\$]</b>	1000-1300	1500-1900	400-600	700-900	600-800
<b>Print volume [mm<sup>3</sup>]</b>	250 <sup>3</sup>	250 <sup>3</sup>	120 <sup>3</sup>	250x210x240	230 <sup>3</sup>
<b>Enclosed</b>	Yes	Yes	Yes	Yes	No
<b>Kinematics</b>	CoreXY	CoreXY	CoreXY	CoreXZ	CoreXY

**Table 2.3**  
*Overview of Voron models*



**Figure 2.10**  
*The currently developed Voron models, from left to right :  
V1, V2, V0, Switchwire, Legacy [34]*

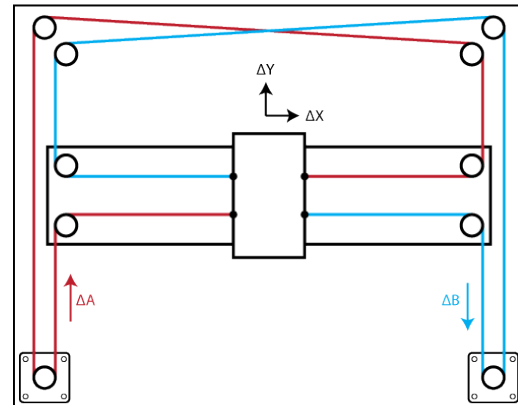
As the printers are made from off-the-shelf and custom 3D printed parts, the Voron community started printing the required parts of the Voron for other people who were also interested in building themselves a Voron. The community calls it Print it Forward.

## The Voron 0.1



**Figure 2.11**

*The Voron 0.1 FDM printer [34]*



Equations of Motion:

$$\Delta X = \frac{1}{2}(\Delta A + \Delta B), \quad \Delta Y = \frac{1}{2}(\Delta A - \Delta B)$$

$$\Delta A = \Delta X + \Delta Y, \quad \Delta B = \Delta X - \Delta Y$$

**Figure 2.12**

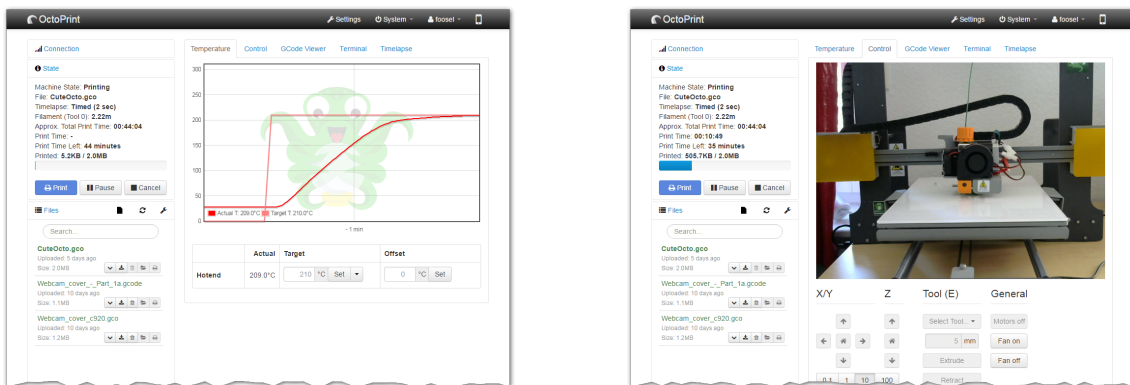
*Schematic of CoreXY kinematics [35]*

The Voron 0.1, shown in Figure 2.11, is designed to have a small footprint (230x230mm), be fully enclosed to print materials, such as ABS, which need to be in a heated atmosphere and be able to FDM with fast speeds and high accelerations. Its CoreXY kinematics and Z lead screw results in the printed part moving minimally while the head moves in the X-Y plane. A schematic of CoreXY kinematics is illustrated in Figure 2.12. For rotation of motor A, the head moves in +X and +Y and for rotation of the motor B, the head moves in +X and -Y.

## 2.5.3 Software

### OctoPrint

The Voron project leverages open source software for FDM to not have to make it themselves. One of them is OctoPrint which features a user friendly way to control the printer through the use of a Raspberry Pi microprocessor. Some tabs of OctoPrint are shown in Figure 2.13.



**Figure 2.13**

*OctoPrint: View of the temperature and camera feed [36]*

OctoPrint comes with a variety of features such as temperature control, automatic homing and many more which make the user experience of using a FDM printer more convenient. Furthermore, as an open source project, updates come often and user created plugins to do a particular task are plentiful.

### Klipper

A second open source software made specifically for FDM printing is Klipper. It is a firmware which is meant to work alongside the microcontroller of the FDM printer on an additional microprocessor. The microprocessor, having more computational power, enables more complex features that the microcontroller cannot do alone due to its lack of computational power. Faster and better quality prints are possible by compensating for the jerk of the printer by having different acceleration pattern and running advanced filament pressure control.

### **OctoPrint - Enclosure plugin**

This Octoprint plugin is used to reduce the development time of MEW on the Voron and allow for more data collection while performing MEW. It was developed by Vitor de Miranda Henrique [37] to automate some tasks while using a FDM printer. It was modified to allow for the control of the high voltage power supply through G-code and for the monitoring of the environmental sensor and the high voltage through OctoPrint.

# Chapter 3

## Methodology

### 3.1 Materials

The polymer used to MEW is Caprorene, a type of PCL in filament form from Poly-Med, Inc. [38]. The properties of the polymer are listed below :

<b>Fiber diameter</b>	1.75 ± 0.05 [mm]
<b>Melting temperature</b>	62.9 [°C]
<b>Inherent Viscosity</b>	1.7 [dL/g]
<b>Residual monomer</b>	0.1%

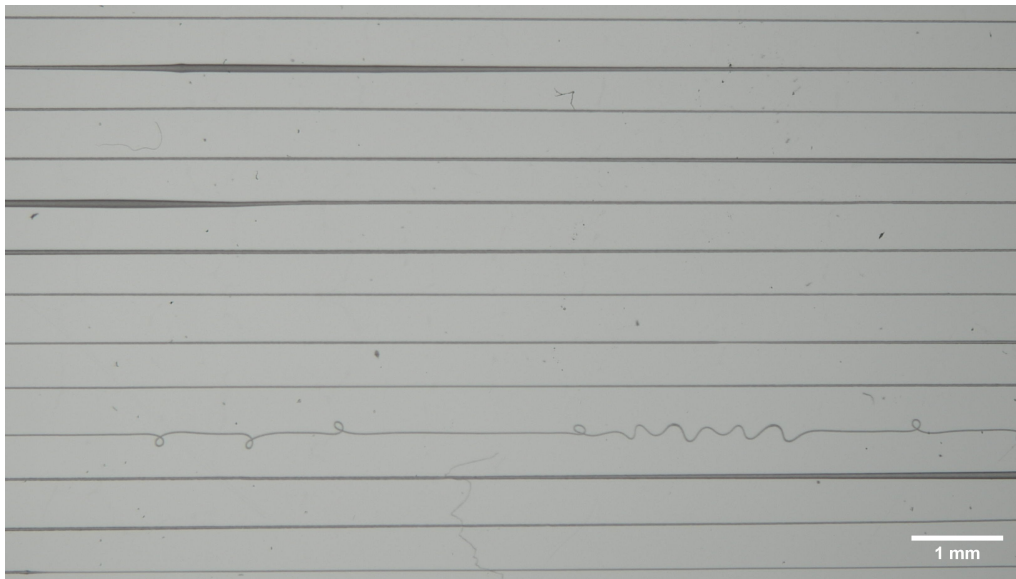
**Table 3.1**  
*Characteristics of Poly-Med's Caprorene PCL [38]*

A standard polymer with known parameters for MEW is Purasorb PCL PC12. The polymer is the principal parameter in the MEW process and dictates the properties and limits in terms of scaffold design. This PCL polymer hasn't been melt electrowritten before and the MEW parameters need to be found.

## 3.2 Theory

### 3.2.1 Melt Electrowriting

In MEW, standard parameters that always work have not been defined. They must be tuned based on the polymer, the ambient temperature, the humidity and the system to reach a range where the jet is formed. Once in that stage, it is not yet stable as the chances of it being pulsing are high. Pulsing is the phenomenon where the thickness of the jet changes depending on CTS, voltage, distance to collector and flow rate. Scaffold size and fiber diameter are considerations to make while attempting to stabilize the jet over the whole print design. Pulsing can be seen in the jet by its jet lag and in the deposited fibers such as shown in Figure 3.1. When doing MEW, a stabilization scaffold is made before the main scaffold to wait for the jet to stabilize.



**Figure 3.1**

*Effect of pulsing in the jet on deposited fiber  
The fiber thickness varies during the printing and affects the deposition*

### 3.2.2 Filament flow equations

As flow rate for MEW is very low, it is important to keep it as constant as possible. The first point is to see the effect of the input fiber diameter on the output fiber. The polymer filament datasheet shows a standard deviation on the thickness of the filament of  $\pm 50\mu m$ . Using the datasheet of the parts of the Voron and some parameters, all listed in



Table 3.2, we can calculate the percentage of difference we would have [33].

Parameter	Symbol	Value
Filament diameter [mm]	$D_{filament}$	1.75
Filament area [mm]	$A_{filament}$	2.405
Hobbed gear radius [mm]	$HB_{radius}$	3.5
Hobbed gear perimeter [mm]	$HB_{perimeter}$	21.991
Steps per rotation	$steps_{rotation}$	200
Number of microsteps	$\#_{\mu steps}$	256
Gear ratio	GR	50:10

**Table 3.2**

*The Voron 0.1's extrusion system parameters [33]*

The volume pushed into the heating element per microstep can be calculated by this formula :

$$V_{\mu step} = \frac{(D_{filament}/2)^2 \cdot \pi \cdot HB_{perimeter}}{GR \cdot steps_{rotation} \cdot \#_{\mu steps}} \quad (3.1)$$

Table 3.3 shows that in the worst case the fiber size could vary of about 10% but only if the filament would change from 1.7 [mm] to 1.8 [mm]. As the filament is continuous and the amount is small, the variation would be small and could be compensated for by looking at the size of the fiber using a microscope.

$$V_{fiber} = \pi \cdot (D_{fiber}/2)^2 \cdot speed \cdot t \quad (3.2)$$

For the theoretical fiber size calculation, we assume that the input material is equal to the output material.

$$V_{fiber} = V_{flow} \cdot t \quad \text{and} \quad V_{flow} = V_{\mu step} \cdot E_{speed} \quad (3.3)$$

Filament diameter [mm]	Volume per microstep [ $\mu L/\mu step$ ]	Percentage
1.70	1.95E-04	0.944
1.725	2.01E-04	0.972
1.75	2.07E-04	1
1.775	2.13E-04	1.029
1.80	2.14E-04	1.035

**Table 3.3**

*Variation in the flow rate depending on the filament diameter*

By combining equation 3.1, equation 3.2 and equation 3.3, we reach a theoretical size of a printed fiber in equation 3.4 :

$$D_{fiber} = 2 \cdot \sqrt{\frac{V_{\mu step} \cdot E_{speed}}{\pi \cdot speed}} \quad (3.4)$$

### 3.2.3 Volume of molten material

Degradation is an important factor in MEW because the polymers used often have to degrade inside the body. If the degradation time is short, the polymer can't be heated for too long otherwise it would result in large flow rate variation and thus, the scaffolds would become inconsistent in terms of fiber quality and spacing.

#### In FDM

The amount of molten polymer depends on the height on the hotend's heat block  $h$ . For high flow hotends, the height can go up to 20mm to allow more time for the heat to transfer to the polymer. The input of the heat block is 2mm to allow for the standard 1.75mm filament to easily go in.

The formula to calculate the volume of molten polymer in a standard FDM heat block is :

$$V = \pi * r^2 * h \quad (3.5)$$

For a 12[mm] height hotend with a 2mm hole, the volume of material would be :

$$V = \pi * (2/2)^2 * 12 = 37.699\mu L \quad (3.6)$$

#### In MEW

MEW uses syringes as the reservoir for the polymer. The whole volume of polymer inside the syringe needs to be heated for the flow to occur. The syringe used is typically 3 cc or 3 mL in volume and half of it is filled with polymer. The volume of the polymer being molten is MEW is in the mL range.

## 3.3 Hardware

### 3.3.1 Available parts

#### Hardware



**Figure 3.2**

*Most of the components required for a Voron 0.1.*

*Left: Extrusions and large components,*

*Middle: Fasteners, belts and electronics,*

*Right: All required FDM parts*

A Voron 0.1 kit comes with all that is needed to do basic FDM except for the FDM printed parts. All the parts are shown in Figure 3.2. The building of the printer takes about one day and a half for someone who hasn't build on yet and has all the right tools to do so. The hotend, comprised of the heat break, heater block and nozzle, is a Phaetus Dragonfly BMO hotend.

#### Electronics

The stepper control board which comes with Voron 0.1 Kit is an SKR mini E3 V2.0 from BigTreeTech. It can drive four steppers with its TMC2209 drivers, heat the hotend to a maximum temperature of 260°C and the print bed to 150°C. The included steppers in the kit are 2x NEMA14 1.8° stepper motors for X and Y displacement, one NEMA17 1.8° with lead screw for the Z axis and one NEMA14 36mm 1.8° pancake stepper for the extrusion.

On the computational side, a Raspberry Pi 3B+ is included to allow for easy connection to the board both wired and wireless and to add computational power to the printer to enable jerk compensation, an understandable user interface through Octoprint and

additional inputs and outputs. It is an important requirement that the on-board computer is as much as possible configurable to add changes to the system both in terms of hardware and software.

### 3.3.2 Extrusion resolution and consistency

As shown in Table 2.1, the flow rate in FDM is orders of magnitude larger than in MEW. To reduce the flow rate, the 50:10 gear ratio gear train of the standard Voron 0.1 is changed to a 26.85:1 stepper motor. The kit contained a 17 teeth motor gear which was modified and put on the higher gear ratio motor output shaft to mesh with the 50 tooth gear in the print head. The final gear ratio can be calculated by multiplying the gear ratio's in the gear train :

$$GearRatio = 26.85 \cdot \frac{50}{17} = 78.97 \approx 80 \quad (3.7)$$

Comparing to the 50:10 gear ratio, the new gear ratio will turn about 16 times slower and the volume resolution will increase by the same amount.

### 3.3.3 Glass slides

To limit the chance of arcing and thus damaging the electronics, MEW on the Voron is done over a glass slide. The usual glass slides used in MEW are microscope slides of 75 x 26 x 1 mm. To allow for more area and further limit the risk of arcing, larger glass slides of 75 x 50 x 1 mm were used.

### 3.3.4 High voltage

MEW requires a high voltage to attract the polymer to the collector. In order to generate the high voltage, high voltage power supplies (HVPS) are used and connected to the print head and the collector is grounded or the high voltage is connected to the collector and the print head is grounded. As the nozzle of the Voron is heated and connected to the control board applying a high voltage to it would break it. Considering this, the high voltage is connected to the build plate and the heating connection and thermistor of the build plate are removed.

The HVPS used is a Spruce Science Lab mate, shown in Figure 3.3. It's a 10kV, 2mA max output power supply with analog control for both the high voltage and its current. The price of this HVPS is around 900\$ [39]. This high voltage power supply was chosen for its lower price than average and for the voltage control options it has.



**Figure 3.3**  
*Spruce Science's lab mate [39]*

### Safety

Safety is essential when using high voltage. This has been taken into consideration throughout the project and specific design choices are made for the user to have the least possibility to enter in contact with high voltage. The choice of HVPS is left to the user but some capabilities need to be specifically chosen to ensure the safety of the user and the automation of the system. Those properties are:

- Be able to limit the current of the HVPS
- Be able to control the voltage of the HVPS through the on-board computer

### 3.3.5 High voltage control

High voltage is dangerous for the system and the user if not managed correctly. Adding a way to control the high voltage both turning it on and off and the amount of voltage is a great addition in terms of functionalities for automation and safety.

Standard HVPS have analog inputs and outputs connections where the voltage and current can be controlled and monitored. Additionally, they have digital input pins to turn on the high voltage and allow for remote control to be active. When remote control is active, the HVPS will measure the input signal voltage and output a high voltage depending on it. The input and output voltages are in the range of 0-10V. To be able to control the HVPS on its full range, a standard microcontroller has to have its 3.3V max voltage amplified to 10V.

As the Raspberry Pi doesn't have analog inputs nor outputs, additional boards have to be added to allow for voltage control. For voltage control, four additional breakout boards are added to the hardware and listed in Table 3.4. As the HVPS require 10V to turn on the high voltage and the remote control, the digital signal from the Raspberry Pi has to be increased. To allow this, a level shifter is used as it is cheaper and it only converts digital signals.

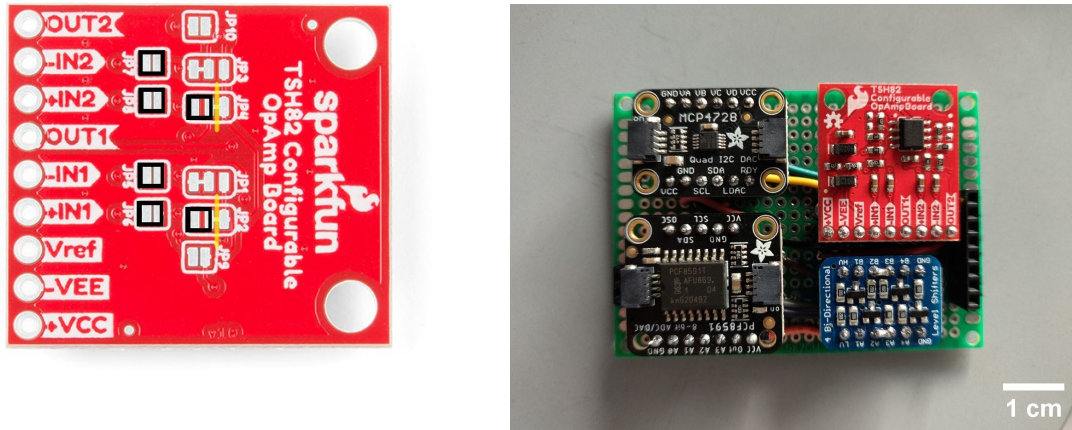
Name	Function
Analog to digital converter - PCF8591	Monitor the high voltage
Digital to analog converter - MCP4728	Allow the raspberry Pi to send analog voltages
Level shifter - BSS138	Increase the digital voltage from 3.3V to 10V
Operation amplifier - TSH82	Increase the analog voltage range from 0-3.3V to 0-10V

**Table 3.4**  
*Components of the voltage control board and their function*

The high voltage and its current are controlled through I2C and the turning on and remote control by two digital pins from the Raspberry Pi. The electrical schematic of the voltage control board is in appendix C.

The amplification board is set by default to multiply the received voltage by 4.7 and if we use it with the maximum 3.3V, it would result in a 15.51V if the amplifier wasn't limited by its 10V maximum input voltage. The amplification board has to be hardwired to be a non-inverting DC-coupled operational-amplifier by soldering the jumper pads on the PCB as shown on Figure 3.4. For the required configuration, two connections must be

severed, by cutting the manufactured copper traces : JP4 and JP2 and connecting JP2, JP4, JP5, JP6, JP7 and JP8.



**Figure 3.4**

*Left : TSH82 configuration, right : complete high voltage control board*

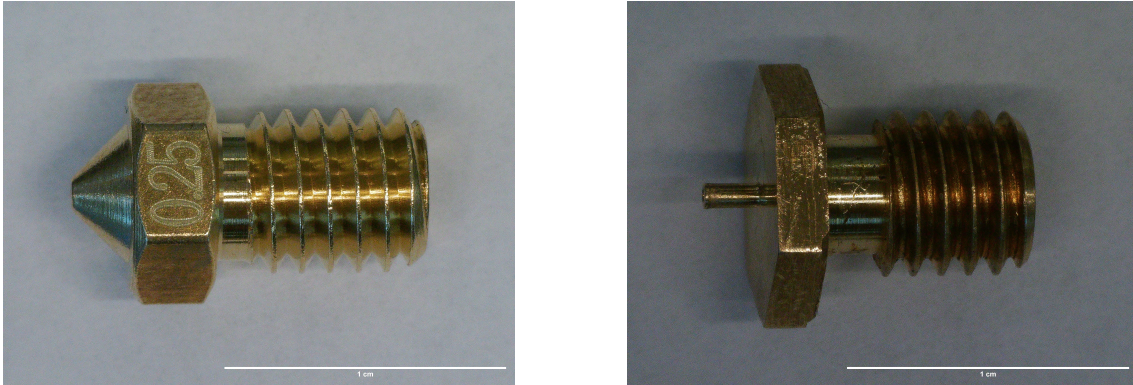
The analog supply to control the voltage and current needs to constantly send out an analog voltage. Therefore the MCP4728 is chosen as it has an electrically erasable programmable read-only memory or EEPROM. With a call of the code, the voltage can be changed and stay active even if the program isn't running anymore.

The full control board with each of its components is shown on Figure 3.4. The components are connected to the prototyping board with female headers to easily remove them if some were to break. Its footprint is small enough to fit inside the back of the Voron 0.1 with the rest of its electronic components.



### 3.3.6 Change of nozzle

Many parameters were tested using a standard 0.25mm FDM nozzle. To make the nozzle for the printer to act more like a standard MEW syringe nozzle, a nozzle was machined out of a M6 brass bolt. The two nozzles are shown in Figure 3.5. The schematic of the custom nozzle is in appendix C.



**Figure 3.5**

*Left : Standard 0.25mm FDM nozzle, right: custom nozzle*

### 3.3.7 Higher gear ratio motor

The standard stepper motor which comes with the Voron 0.1 kit has a gear ratio of 50:10 which limits the resolution of the volume pushed. When speed is slow and the number of microsteps is high, stepper motors have less torque and having a high gear ratio would reduce the required torque. To allow for finer resolution of fiber, the motor is changed to a 27:1 stepper motor with a gear train. To link the shaft of the output of the stepper to the gear linked to extrusion, a 17 teeth motor gear is attached onto it.

From the equation 3.1, we can calculate the flow rate per step.

For a 50:10 gear ratio:

$$V_{\mu step}(GR = 5) = \frac{(1.75/2)^2 \cdot \pi}{5 \cdot 200 \cdot 256 \cdot 21.991} = 206.621nL/step \quad (3.8)$$

For a 80:1 gear ratio:

$$V_{\mu step}(GR = 80) = \frac{(1.75/2)^2 \cdot \pi}{80 \cdot 200 \cdot 256 \cdot 21.991} = 13.08nL/step \quad (3.9)$$

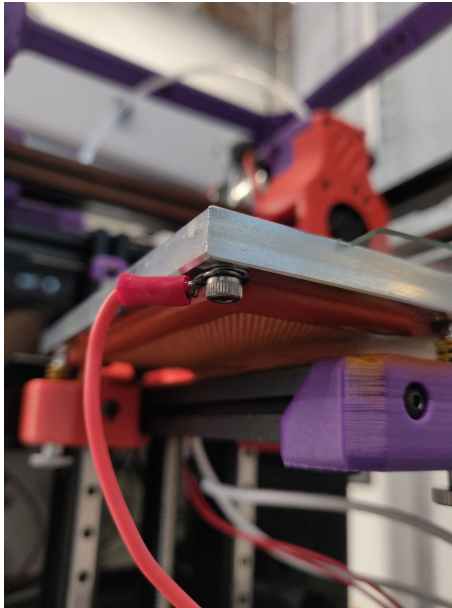
### 3.3.8 Environmental measurements

The quality of the fibers can be affected by the local temperature and humidity. An BME280 environmental sensor [40] is added inside the enclosure to closely monitor the temperature and humidity. The sensors values are shown in the Octoprint user interface and can be logged.

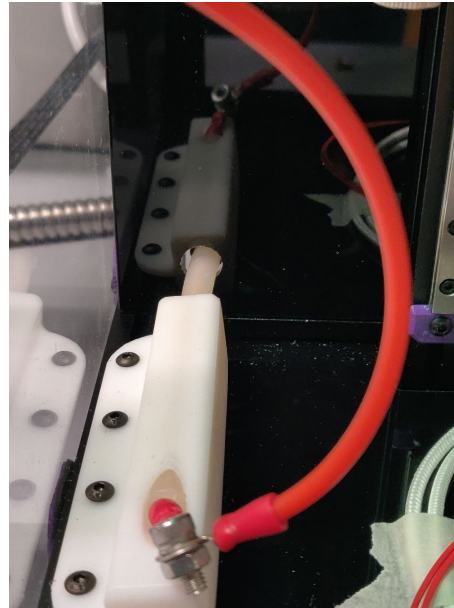
### 3.3.9 FDM conversion to MEW

Most of the standard parts of the Voron 0.1 are kept for the conversion to MEW. Minor modifications are required to the structure to install the high voltage, some 3D printed parts need to be printed and the extrusion motor needs to be changed.

#### Changes to structure



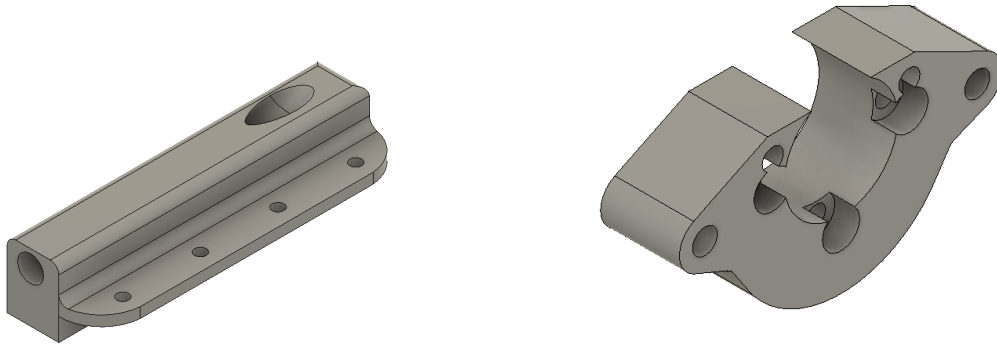
**Figure 3.6**  
*Tapped hole in the build plate to  
attach high voltage cable*



**Figure 3.7**  
*Hole in the back of the Voron for  
high voltage cable*

The high voltage cable is guided into the build chamber by a hole added to the back of the printer shown in Figure 3.7 and a 3D printed part is securing it to the structure of the printer shown in Figure 3.8. A tapped hole is made into the build plate to allow a silicon cable to link to the high voltage cable pictured in Figure 3.6. Both sides are connected using a crimp ring, M4 screws, split washer, and nut to insure a reliable connection.

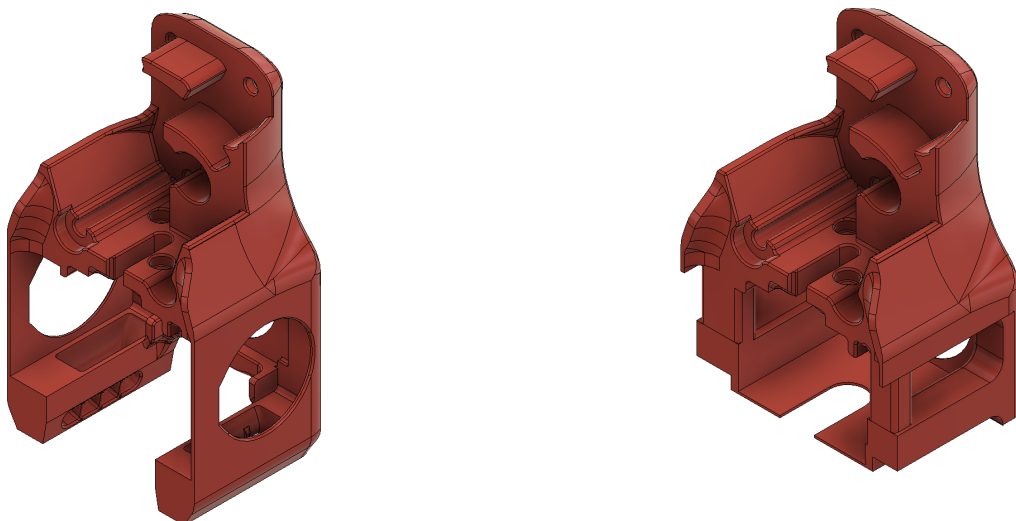
The standard stepper motor was changed to a 27:1 gear ratio stepper motor. A motor adaptor shown in Figure 3.8 allows for a drop-in replacement of the standard motor with the high gear ratio motor.



**Figure 3.8**

*The 3D printed parts required for the conversion  
Left : high voltage cable holder, right : the motor adaptor*

For FDM, the fans on the print head rapidly cool the deposited molten polymer to remove warping. In MEW, the fans aren't needed, and a custom head was designed to reduce the amount of air coming from the extruder cooling fan to the jet. This custom print head is shown in Figure 3.9 with the original print head.



**Figure 3.9**

*Left: original print head, right: custom print head*

## 3.4 Software

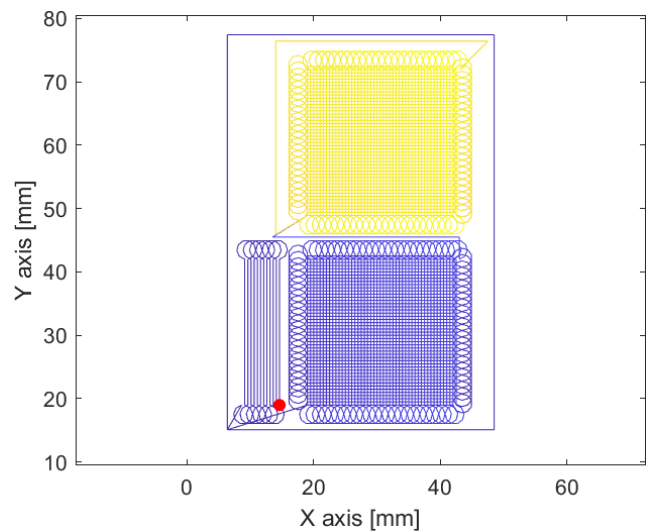
### 3.4.1 G-code generation

The G-code generator allows the user to set several parameters to generate a design to MEW. A short list of some parameters is in Table 3.5 and generated design is shown in Figure 3.10. The generator is improved several times during the project to add functionalities for the scaffolds. The full list of parameters is in appendix A.1. It is based on a code written by Dr. Ievgenii Liashenko. As many parameters define the final scaffolds, it is unlikely to know from the start what will be needed further down the line.

Parameter	Value
Feed rate [mm/min]	300
Length [mm]	23
Width [mm]	23
Spacing 1 [mm]	0.5
Spacing 2 [mm]	0.5
Number of layers	10
Set voltage [kV]	4.8
Loops [ON/OFF]	1
Priming line [ON/OFF]	1

**Table 3.5**

*Some of the parameters to define the scaffold design*



**Figure 3.10**

*The resulting G-code path with stabilization paths*

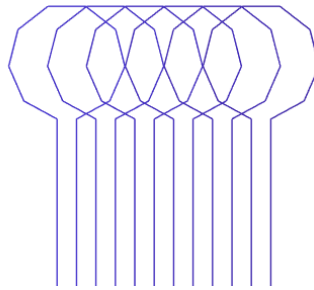
The generator is designed to work mainly in the X and Y direction with a Z adjustment at the start of the print. If set, it can adjust the height at the end of the layers and at the end of the print. The program is written in MATLAB but could easily be implemented in another programming language such as Python as the generator outputs a text file. The full MATLAB code is in appendix B.

The generator makes the task of creating G-codes for MEW as parametric as possible. The values of the scaffold design are defined at the start of the code and a graphical representation of the path to be generated is shown to the user. The generated G-code is only composed of G1 linear movements because G2, G3 arc movements aren't interpreted by all printers.

The G-code generated by the program sets the hotend temperature, moves it into the start position in X, Y, Z and turns the high voltage 'ON'. The movements of the printer are in relative positioning.

### Loops

Loops are the part in a scaffold where the fiber lines have reached its end and need to link with the next one at a defined spacing. The size of the loops should be altered depending on the jet lag. The code is made parametric to always link the end of the lines. A image of generated loops is shown in Figure 3.11.



**Figure 3.11**

*The loops generated by the generator*

### 3.4.2 Extrusion control

The extrusion control is managed on another Raspberry Pi connected to a TMC2209 stepper driver breakout board. This is the same stepper driver which is on the stepper control board of the Voron 0.1 but as it is not soldered on, it allows easy access to electrical pins. A custom library is written to allow the stepper to turn continuously by using an internal step generator after its parameters have been set.

The input parameter  $VACTUAL$  is a value which is multiplied to the internal oscillator frequency of 0.715 [Hz] to obtain the extrusion speed. This defines the  $\mu step$  velocity :

$$E_{speed}[\frac{\mu step}{s}] = VACTUAL \cdot 0.715 \quad (3.10)$$

The extrusion needs to be independent of the movement otherwise the pressure applied to the polymer possibly changes and pulsing is more likely happen in such situations.

## 3.5 Imaging tools and measurements

### 3.5.1 Optical microscope and SEM

Macroscopic images of the scaffold on glass slides were taken using an Andonstar AD409 Digital Microscope with a black glossy acrylic sheet as background. Fiber diameters and spacing between fiber were imaged using a Keyence VHX-7000. Automated stitching of imaged was performed using Keyence's software to keep high resolution while having multiple individual fibers. Quality of fibers and their staking were viewed and recorded using a Thermofisher Apero 2 Scanning electron microscope.

### 3.5.2 Recording of data

Data aggregation was performed using Fiji/IMAGEJ. The scale bar generated by the Keyence's software was adjusted for each image with its overlaid scale bar to calculated the relation of pixel to  $\mu m$ . For the fiber size to extrusion speed measurements, 20 values were taken on three different scaffolds made on different days with the same extrusion speed. It was repeated for three extrusion speeds: 25, 50 and 100. To evaluate the spacing accuracy, 10 vertical and 10 horizontal values are measured on one scaffold per spacing with the spacing being 250, 500, 750  $\mu m$ .

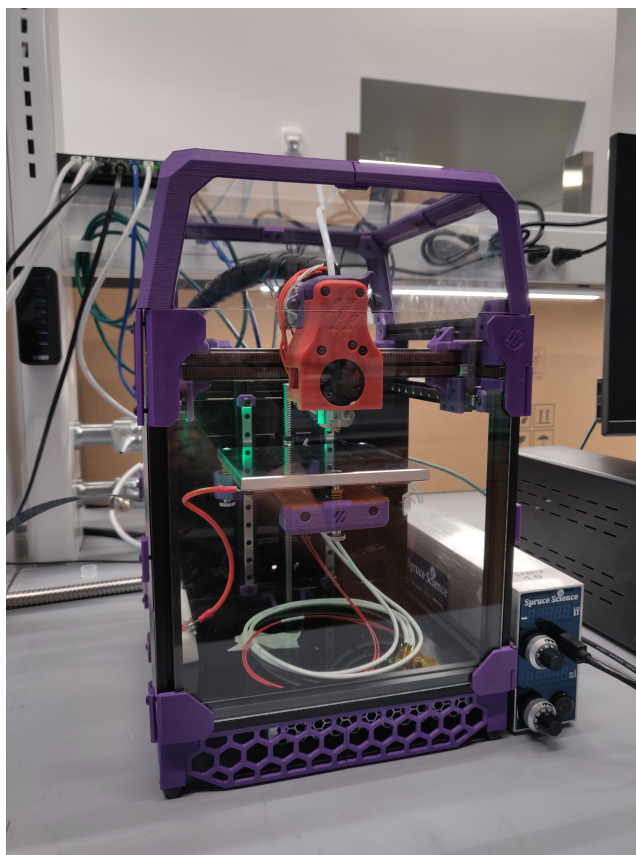
### 3.5.3 Generation of graphs

The graphs presenting the data with mean and standard deviation are generated using Python with pandas and Matplotlib libraries.

## Chapter 4

# Results and discussion

The first part of this chapter summarizes the key findings which allowed the system to perform MEW. The second part shows the resulting scaffolds and compares their quality with the scaffolds made with a state of the art system.



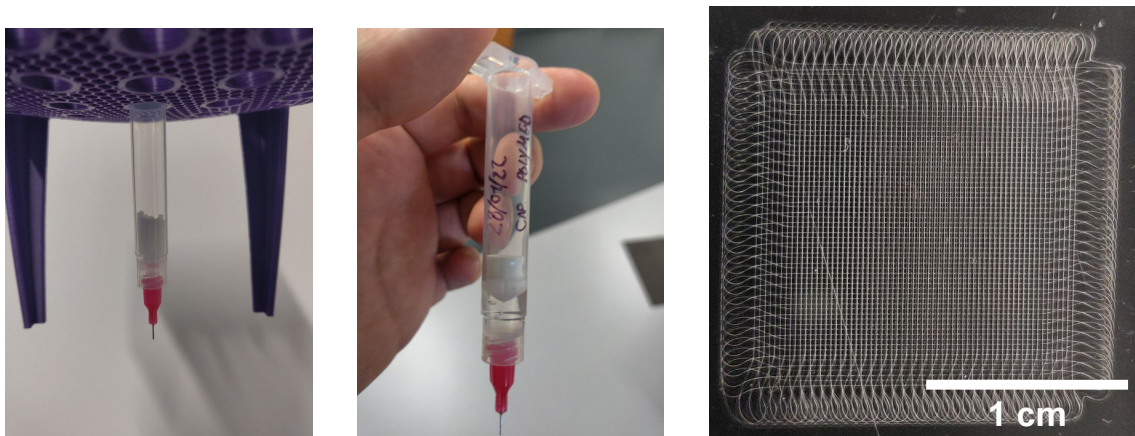
**Figure 4.1**  
*Image of the Voron 0.1 FDM system converted to MEW*



## 4.1 Achieving MEW on the system

### 4.1.1 Testing the polymer

The first step was to characterize the PCL Caproprene polymer from Poly-Med for its ability to be melt electrowritten. The polymer was processed by cutting the filament into 2cm long parts, filled inside a 3ml syringe and left to melt in an oven at 90°C for 3 hours. The processed PCL was utilized to fabricate a scaffold with a QUT MEW system. The resulting scaffold can be seen in Figure 4.2 and shows good results macroscopically, proving that the polymer can be processed using MEW.



**Figure 4.2**

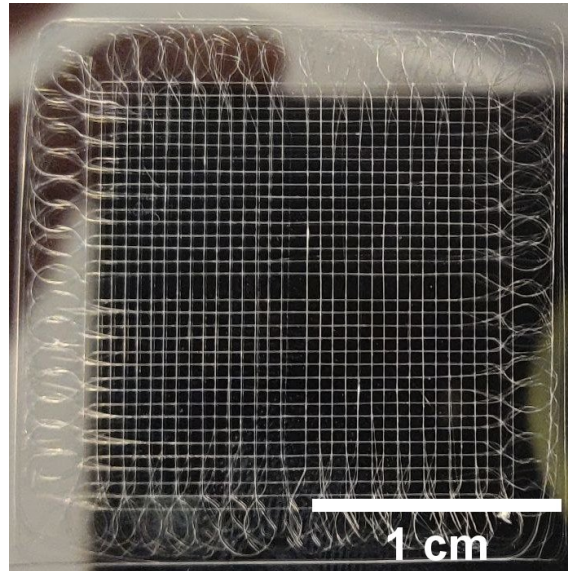
*Process to melt filament to syringe to be used in MEW printer and resulting scaffold.*

*Left to right:*

*Filament pieces in syringe, Syringe with the molten polymer after heating and resulting scaffold*

#### 4.1.2 Testing the motion system

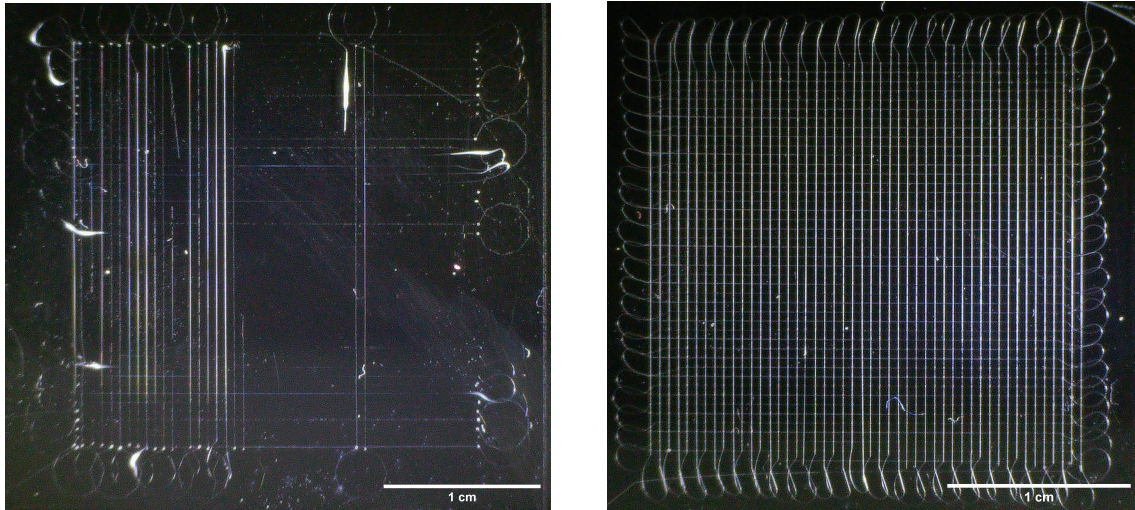
An evaluation of the motion system of Voron is performed by replacing the standard FDM print head with a pressure fed heated syringe system. The scaffold produced with the syringe is shown in Figure 4.3. Fiber stacking is good but not optimal which could be attributed to the motion system.



**Figure 4.3**  
*5 layer MEW scaffold made with pressure fed system on Voron printer*

### 4.1.3 Nozzle design

Redesigning of the standard FDM nozzle proved to be pivotal in successfully performing MEW using Voron. As shown in Figure 4.4, the same printing parameters using two different nozzles, a standard 0.25mm FDM nozzle and the custom designed nozzle, lead to drastically different results.



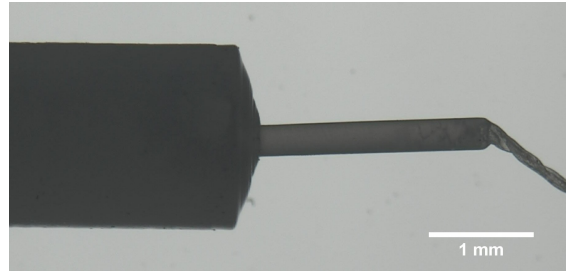
**Figure 4.4**

*Same parameters MEW scaffolds with different nozzles*

*Left : made with standard 0.25mm FDM nozzle, right: made with custom nozzle*

The external geometry of the nozzle has a clear effect on the flow rate and the deposition of the fibers. One hypothesis is that the electric field is more concentrated at the tip of the nozzle due to the needle like shape of the custom designed nozzle. The sharp hexagonal shape around the nozzle which is used to tighten the nozzle to the print head could potentially also affect the electrical field negatively, and it will be interesting to try additional nozzle designs to understand their effect on the jet.

An additional improvement could be to reduce the taper angle inside the custom nozzle. As it was made with a standard 2mm drill for the filament input side, the taper angle of  $120^\circ$  is steep for the flow as shown on Figure 4.5. Machining the nozzle to have a  $90^\circ$  taper would reduce the pressure and possibly make the flow more uniform.



**Figure 4.5**

*Cold pull of the filament from the nozzle shows the large 120° taper*

#### **4.1.4 Microsteps of the stepper motor**

MEW requires a continuous polymer flow at a low flow rate. The principle driving the stepper motor is their precise but discrete steps. It is essential to increase the number of microsteps per step and the speed in order to mimic the pressure continuity of a pressure fed syringe in MEW. A value of 256 microsteps per step was found optimal using the high gear ratio motor to obtain a constant and continuous flow which resembled the flow rate for MEW. The standard extrusion stepper motor of the Voron kit was tested with the same value of microsteps and it was found that torque required by the motor was too high for the rated amperage which resulted in skipping of the motor.

#### **4.1.5 Skipping of the fiber**

During the MEW process, a skipping motion of the jet was observed. This is visible as the fiber would momentarily move in the direction opposite to that of the print head and then continue its movement. This phenomenon is not as obvious with smaller fiber diameters. The general hypothesis is that the fiber gets charged and then discharges quickly, and doing so makes this fast movement.

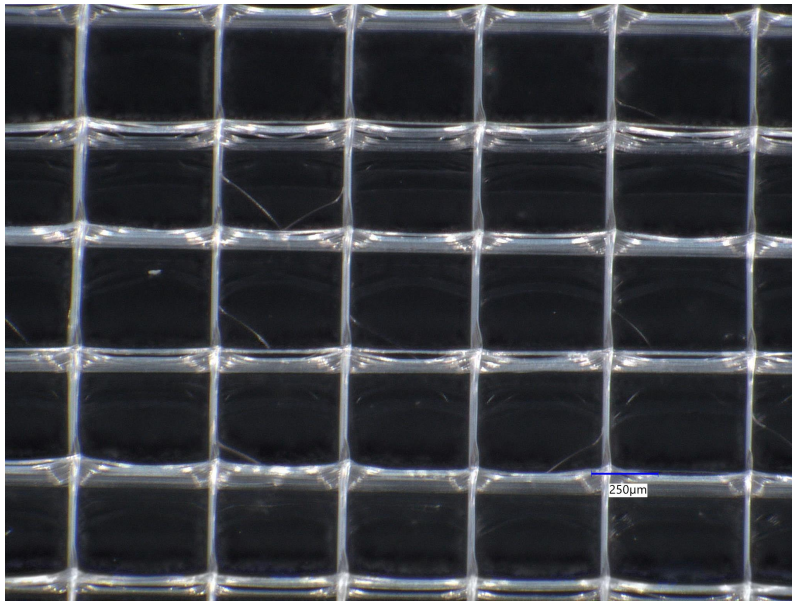
## 4.2 Resulting scaffolds and comparison

The parameters for stability of the jet are found after trial and error, and are listed in Table 4.1. The flow rate is theoretical and calculated base on equation 3.2.

Scaffold size [mm]	23x23
Temperature [°C]	80
Flow rate [ $\mu\text{L}/\text{h}$ ]	1.68
Distance to collector [mm]	3.2
Voltage [kV]	4.8
Feed rate [mm/min]	405
Pause time before loop	600
Pause time after loop	500

**Table 4.1**

*Parameters used to MEW scaffolds on the Voron  
The resulting scaffold is in Figure 4.6*



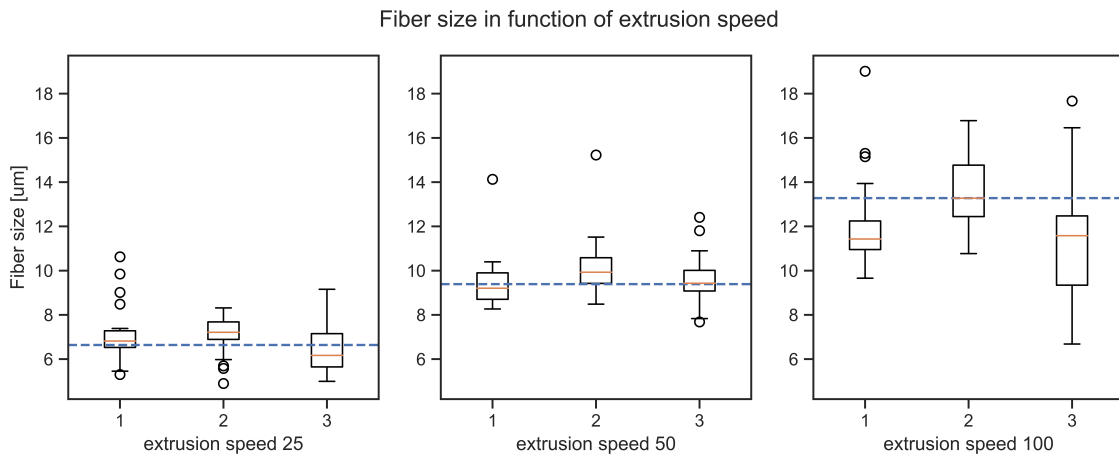
**Figure 4.6**

*10 layer MEW scaffold with 7 $\mu\text{m}$  fibers and 0.5mm spacing  
The parameters used are in Table 4.1*

These parameters helped to achieve a relatively stable jet but some parts of the scaffold still showed fiber pulsing. Further optimization of the parameters is always possible but characterization of the scaffolds was performed as the jet formed was stable.

### 4.2.1 Fiber diameter linked to extrusion speed

The dependence of fiber size on the extrusion speed of the polymer/filament was measured on three scaffolds with 20 measurements per scaffold. The measurements were performed for three extrusion speeds : 25, 50 and 100 which theoretically correspond to 0.84, 1.68 and 3.37  $\mu\text{L}/\text{h}$ . The theoretical fiber sizes for these extrusion speeds are 6.64, 9.39, 13.28  $\mu\text{m}$  and are shown in Figure 4.7 in blue dashed lines.



**Figure 4.7**

*Fiber diameter in function of extrusion speed*

These results were obtained using the parameters which were optimized for an extrusion speed of 50. Larger deviations can be expected for extrusion speeds of 25 and 100 as a lower or higher speed value changes the flow rate of the polymer which can impact other parameters. The fiber sizes depending on the extrusion speed are reported in the Table 4.2. The experimental values correlate well with the theoretical values and a maximum variation of only 7% is observed. The standard deviation is larger for the 100 extrusion speed which is understandable as the additional flow would require increased voltage to have a better stabilization of the jet.

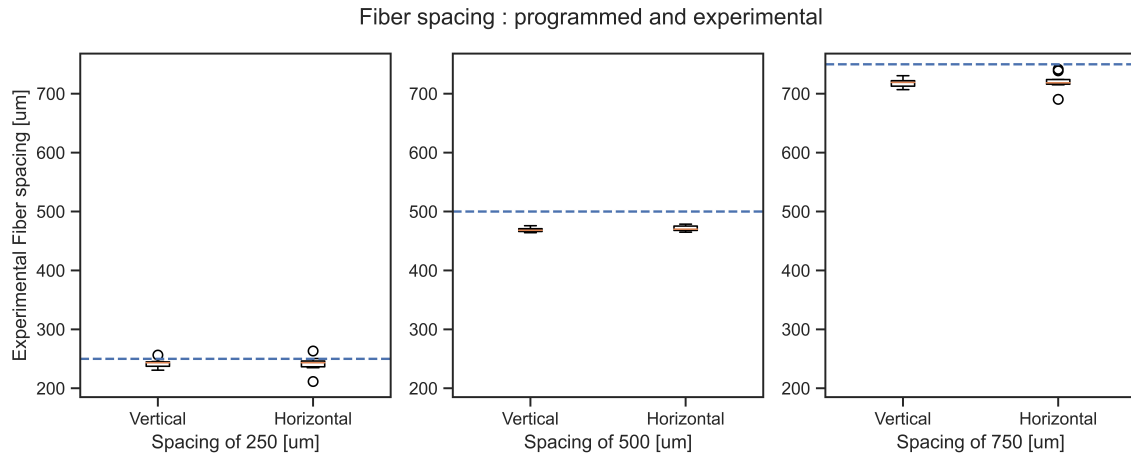
<b>Extrusion speed</b>	25	50	100
<b>Fiber size [μm]</b>	$6.9 \pm 1.2$	$9.8 \pm 1.3$	$12.4 \pm 2.5$
<b>Theoretical fiber size [μm]</b>	6.64	9.39	13.28
<b>Error [%]</b>	3.98	4.33	-6.88

**Table 4.2**

*Fiber diameter inspection results*

### 4.2.2 Fiber spacing

Fiber spacing measurements were performed on scaffolds with 250, 500, and 750  $\mu\text{m}$  spacing. The values are grouped into vertical and horizontal directions. Ten values were measured in both orientations. The mean and standard deviation of the spacing values are reported on Figure 4.8 and the theoretical values are marked as dashed lines.



**Figure 4.8**

*Fiber spacing : Theoretical and experimental*

The first defining element of the spacing is the motion system. As the fiber is attracted by the electrical field on to the previous deposited fiber, the first fiber's position is mostly reliant on the precision of the motion system. The second defining element is the jet lag. If the jet lag is unstable, the fiber will take more time to complete the loop and will be offset until it aligns back with the direction of the print head.

<b>Programmed spacing [um]</b>	250	500	750
<b>Experimental spacing [um]</b>	241.472 ± 10.523	469.912 ± 4.326	719.054 ± 10.992
<b>Error [%]</b>	3.4112	6.018	4.126

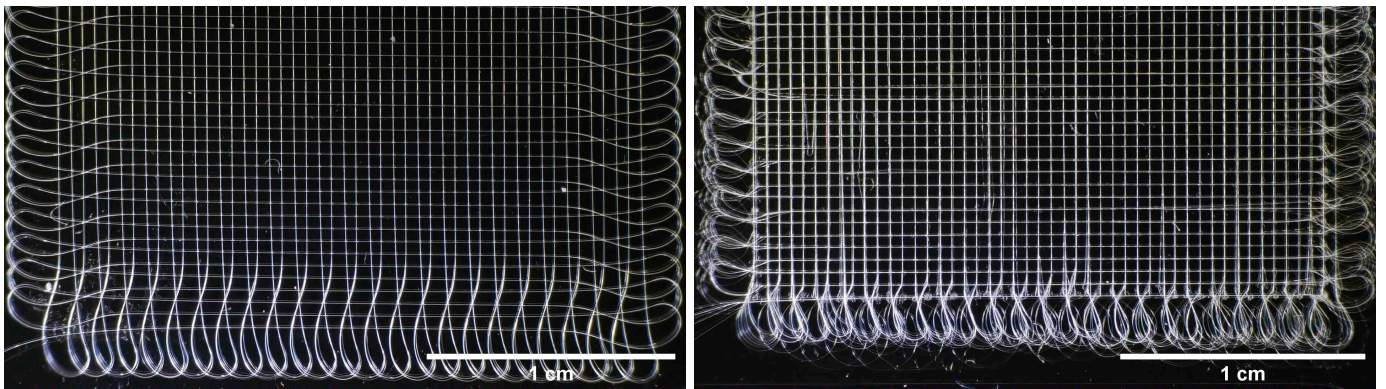
**Table 4.3**

*Fiber spacing inspection results*

Some experimental spacing standard deviation values are  $\pm 10\mu\text{m}$  which could potentially make stacking difficult with small fibers. This deviation could originate from the slack in the belts of the motion system which are tightened by hand. The error regarding the position is in the 5% but is consistent between different spacing values and means it could be compensated by an offset depending on the spacing. This shows good promise on having precise scaffolds with defined structures.

### 4.2.3 Comparison with state of the art MEW system

In this section, two scaffolds made with the same G-code and polymer are examined to observe the difference between the converted MEW system and a state of the art system (QUT). The flow rate on the Voron is set to a theoretical value of  $1.68\mu\text{L}/\text{h}$  and the pressure is set to 1.5 bar on the QUT printer. The speed on the Voron is set to  $400\text{ mm}/\text{min}$  and to  $500\text{ mm}/\text{min}$  as the jet is below CTS at  $400\text{ mm}/\text{min}$  on the QUT. Scaffolds made on both systems are shown in Figure 4.9.



**Figure 4.9**

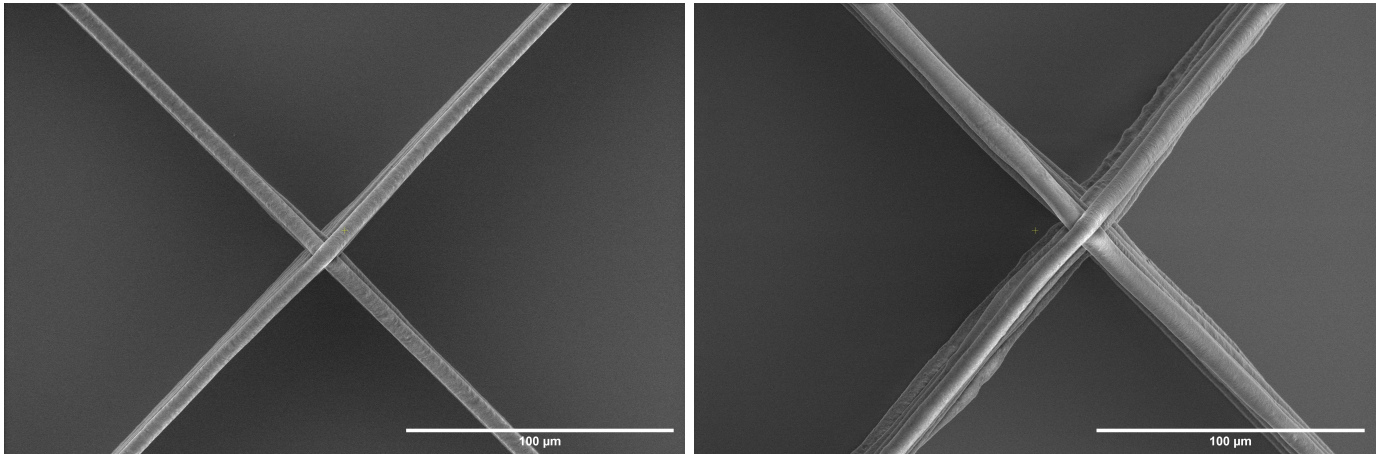
*MEW scaffolds*

*Left: made on QUT, right: made on Voron*



### Fiber surface and thickness

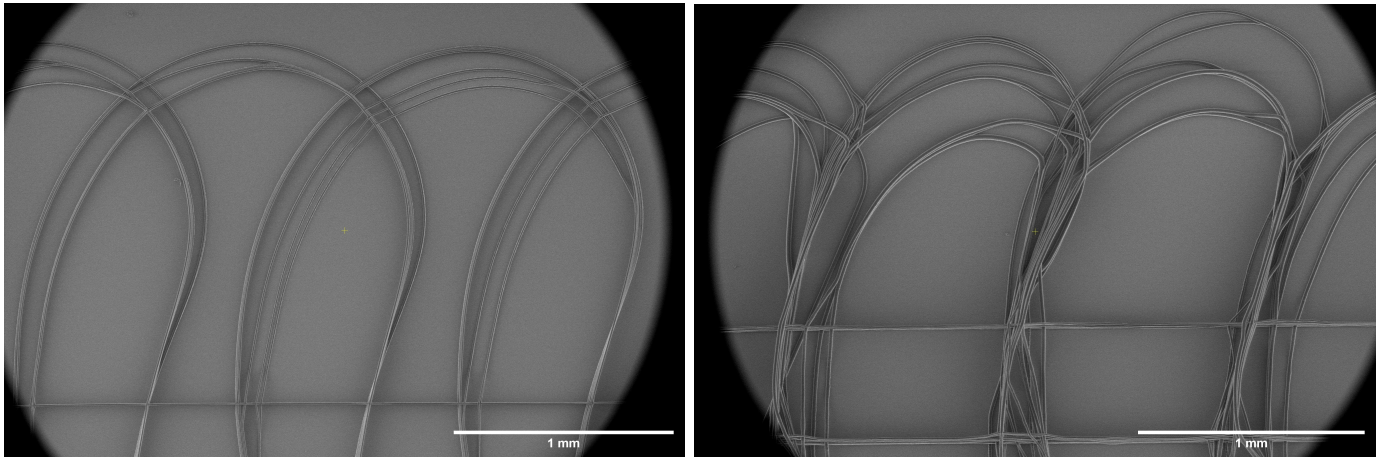
Fibers made with the QUT MEW system have a better fiber diameter homogeneity compared to the ones made on the Voron. The ones made on the Voron have visible variations in fiber diameter from bottom to top layer of the scaffold and it is also noticeable on some regions of individual fibers. The results on the Voron could be caused by a lower temperature at the exit point of the nozzle and a non consistent flow rate.



**Figure 4.10**  
*SEM pictures of fibers*  
*Left on QUT Printer, right on Voron*

## Loops

The SEM images of the scaffolds were compared, and it was noted that the loops on the QUT scaffolds do not all stack but the shape of the loops, in terms of length and width are the same. This can be attributed to little difference in the flow rate during printing. On the Voron, the length of the loops varies more than the width which is most likely due to the variation in the flow rate which probably isn't as constant than the QUT.



**Figure 4.11**

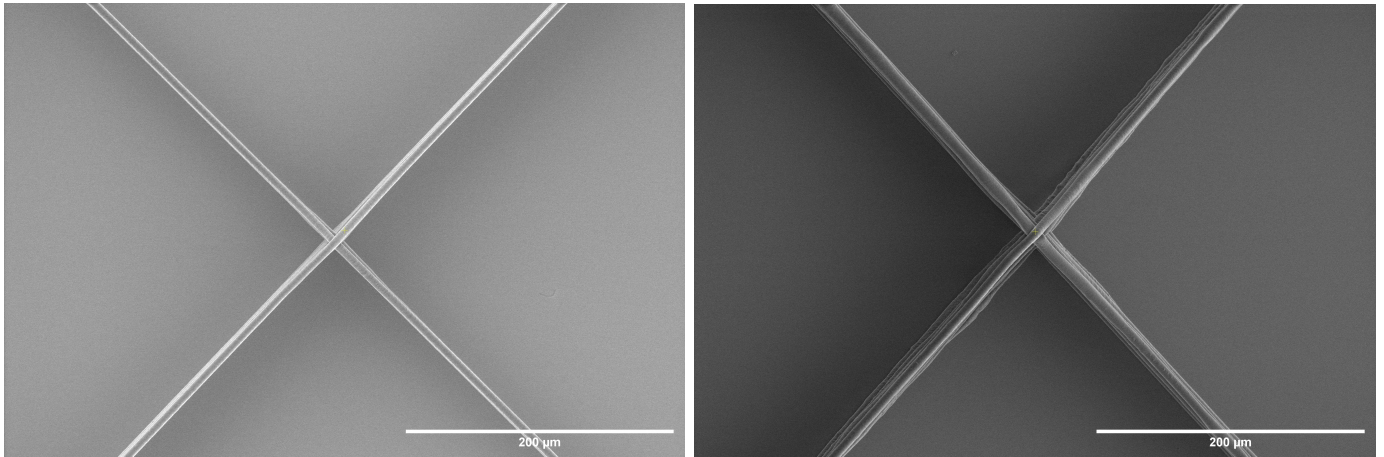
*MEW loops*

*Left on QUT Printer, right on Voron*

Loops in MEW depend on many parameters and are a good macroscopic evaluation feature to analyze the state of the jet. Reaching perfect stacking of the loops on the Voron would require more time in fine tuning the parameters and maybe even the hardware but the resulting fibers do share similar thicknesses and the stacking of the loops is acceptable for their purpose.

## Stacking

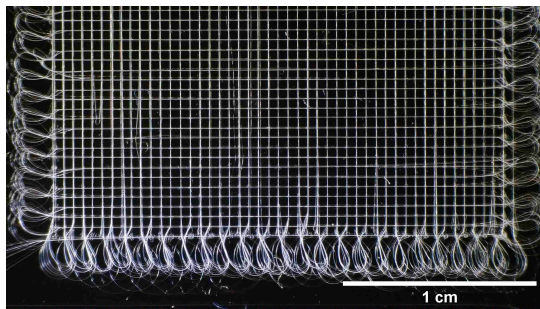
The stacking for both the QUT and Voron scaffolds was successful as all the fibers from bottom to top stack on the same location. On both systems, the fibers didn't fuse into a single mass but stayed separate. The small deviation of the position of the head is compensated by the focusing of the fiber on the previously deposited fiber with the electrical field.



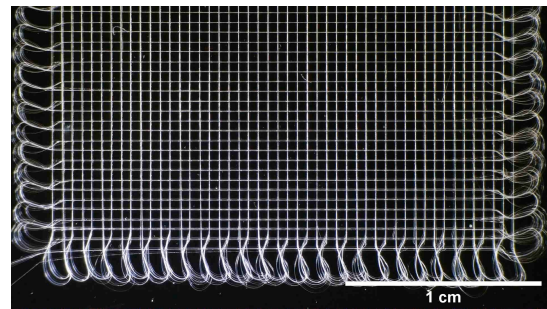
**Figure 4.12**  
*Stacking of fibers*  
*Left on QUT Printer, right on Voron*

#### 4.2.4 Voltage control

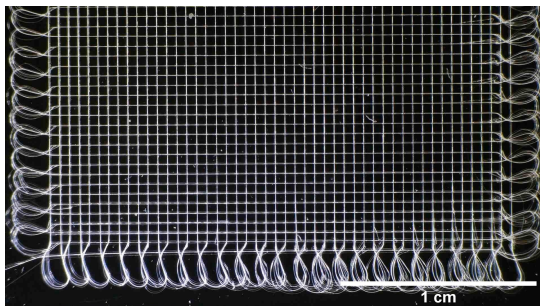
The automation of voltage control allows alteration of the high voltage in increments of 10V. This was tested by increasing the collector distance and incrementing the voltage after each layer to maintain the ratio of voltage to collector distance. In Figures 4.13, 4.14, 4.15 and 4.16, four scaffold variations obtained by altering the nozzle-collector distance and high voltage are shown (Layers 10). One can notice that the loops in the bottom right side of all scaffolds have more stacking issues. As all the scaffolds are made in the same exact coordinates on the Voron, it could mean this specific location is problematic on the system. Another hypothesis is that the nozzle to collector distance between the left and right side of the glass slide is slightly different and could arise from a small tilt of the build plate.



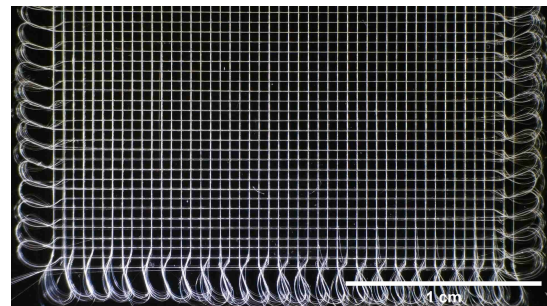
**Figure 4.13**  
*Distance of 3.2mm and 4.8kV*



**Figure 4.14**  
*+0.01mm after each layer*



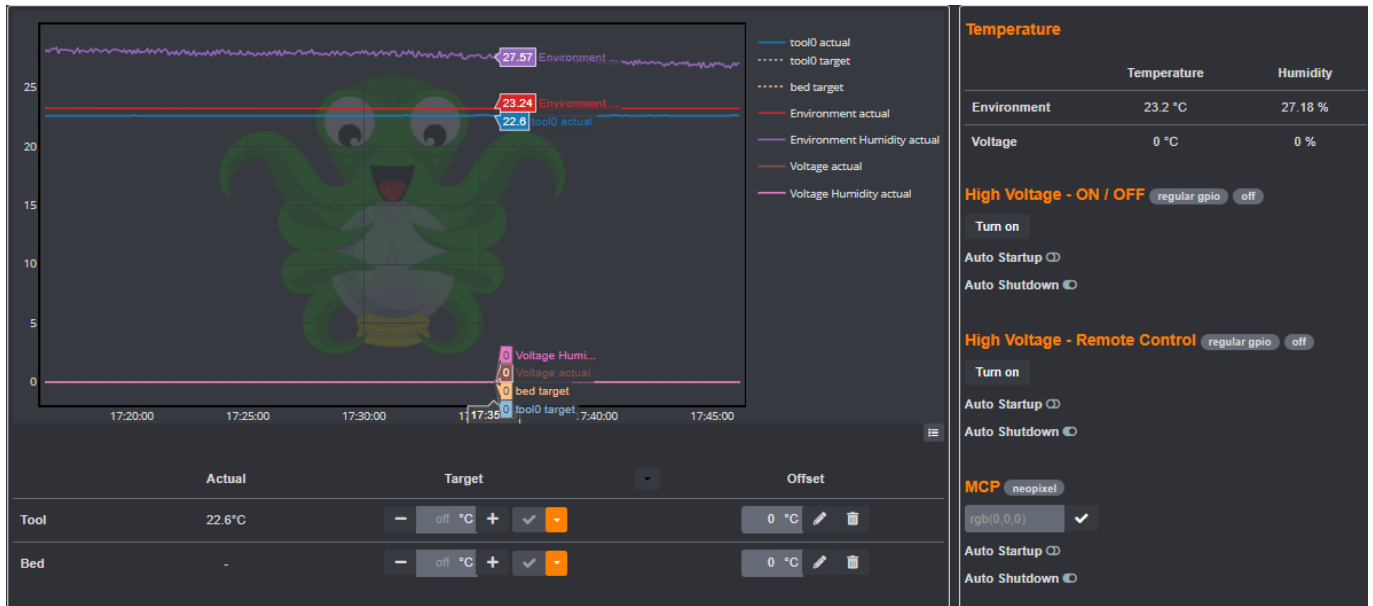
**Figure 4.15**  
*+10V after each layer*



**Figure 4.16**  
*+0.01mm and +10V after each*

### 4.2.5 Environmental monitoring

The environmental sensor reliably measures the temperature and humidity. The measurements are shown on the OctoPrint user interface graph with the hotend temperature and voltage level. Logging of these values can help understand large variations in fiber sizes over a number of scaffolds. A graph with the environmental data is shown in Figure 4.17.



**Figure 4.17**  
*Environmental measurements on OctoPrint*

#### 4.2.6 Price of the system

The cost of the system with its components is summarized in Table 4.4. The price of the 3D printed parts, if bought, can vary between \$50 and \$150 depending on the suppliers. The price was fixed to the average price of 1kg of ABS filament as it was assumed that the user would have access to a FDM printer to make all the required parts.

Component	Price [\$]
Voron 0.1 Kit	600
Voron 0.1 3D printed parts - 1 kg of ABS filament	20
High voltage power supply	900
ADC - MCP4728	7.5
DAC - PCF8591	6.5
Op-amp - TSH82	6
Level shifter - BSS138	4
Environmental sensor - BME280	15
27:1 Gear ratio stepper	35
Custom brass nozzle	50
Miscellaneous components (cables, headers, etc...)	10
<b>Total</b>	<b>1,654</b>

**Table 4.4**

*List of all the components of the modified Voron*

The high voltage power supply is the main contributor to the price of the system followed by the Voron 0.1 kit. If every component of the kit were to be sourced separately, it's price could be lowered to \$400. The price of the custom nozzle was deduced from the time it would take for a machinist to make it on a lathe and the price of a standard M6 brass bolt. The voltage control elements, while not essential to the function of the system, do add an additional layer of safety by allowing the system to turn off the high voltage after finishing a print.

## Chapter 5

# Conclusions

A commercially available FDM system (Voron) was successfully converted to a MEW system. The printed scaffolds were found to have accurate and homogeneous fiber diameters. Stacking accuracy and the inter fiber spacing was also found to be consistent and within 5% error. A small amount of fiber pulsing was still observed. The changes adopted in the Voron 0.1 printer can be implemented quickly to achieve a MEW system with a smaller footprint and build area. The overall cost of the system, including voltage automation and environmental chamber monitoring was \$1,654 which was well below the \$2,000 limit set at the beginning. The G-code generator was modified to make the iterations on the scaffold designs easier and faster. It allowed printing of scaffolds with up to 25 layers, ranging from 5 to 25mm in size and having inter fiber spacing of 250, 500, 750 $\mu$ m.

## Chapter 6

### Future work

As the platform is capable of MEW, many improvements to the system are possible to allow for better user experience. Automating scaffold printing and increasing the amount of data collection about the fibers and printing conditions will be very useful.

#### 6.1 Computer vision

Image analysis through computer vision would bring an addition level of automation to the printing process. It will also allow continuous monitoring of the jet throughout the whole printing process and possibly tune the parameters while printing. The CoreXY kinematics of the Voron forces the camera to be mounted directly to the print head to have a close view of the jet. During the project attempts, shown in Figure 6.1, were made to capture the jet but focus was shifted on to attaining a stable jet.



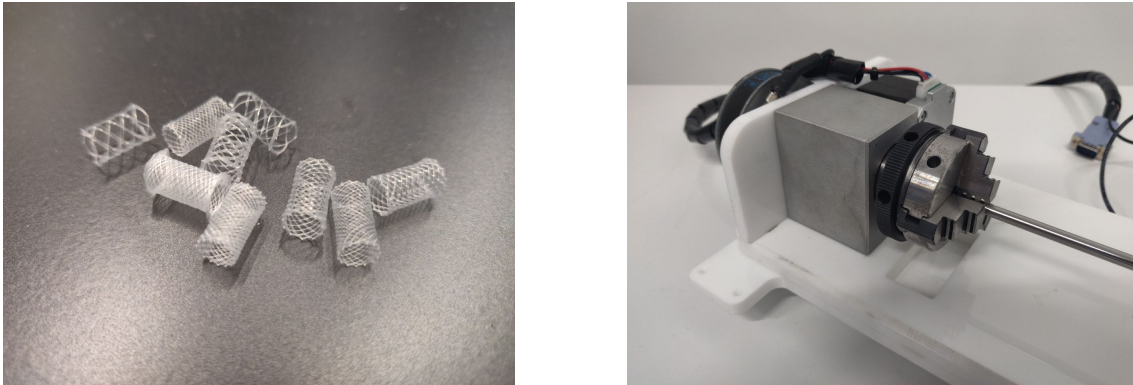
**Figure 6.1**

*Attempt to view the jet. From left to right:  
View of the nozzle from an attached boroscope, raw image and processed image*



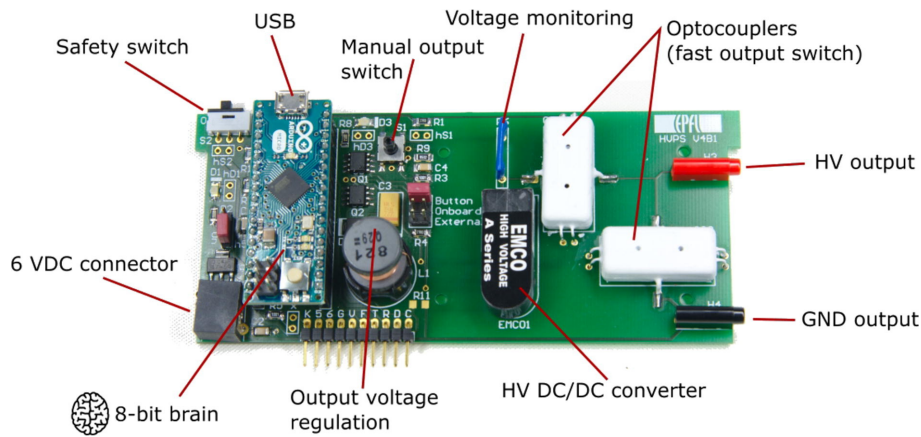
## 6.2 Tubular collector

On the hardware side, the ability of using MEW for printing tubular scaffolds would be a great addition to the system. This is achievable by adding a motor with a mandrel like shown in Figure 6.2 on the Z platform. It is relatively simple, but the control board would have to be changed to allow for the control of more motors.



**Figure 6.2**  
*Left: Tubular scaffolds, right: mandrel with motor*

## 6.3 Reducing the price of the high voltage power supply



**Figure 6.3**  
*Peta-pico-Voltron: An open-source high voltage power supply [41]*

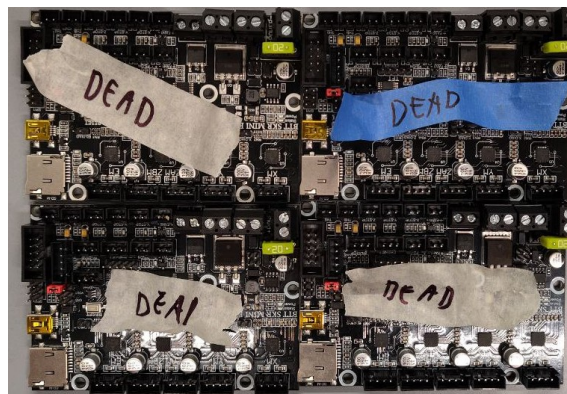
The price of the system could be further reduced by integrating the high voltage power supply in the system using an open-source design such as the one in Figure 6.3.

## 6.4 Glass slide dispensing

As the area used to MEW is relatively small, an automated system could be designed to fit inside the Voron to dispense glass slides and remove them after they've been finished. This would increase throughput of the system and remove the need of a user being present to put a new glass slide in.

## 6.5 High Voltage protection

As the head is grounded and the heating elements are inserted into it. If an arc occurs, there is a chance for the control board to be damaged or destroyed like the ones on Figure 6.4. Isolating the heating elements from the head would protect the control board from getting damaged.



**Figure 6.4**

*A number of control boards were electrocuted*

## 6.6 Additional polymers

With the ability to reach higher temperatures, the polymers used in FDM could be tested for their ability to be melt electrowritten. Additionally, fast degrading polymers, interesting for TERM, could be tested on the system. With a custom heater block, a shorter heating length could be fabricated to reduce the volume of polymer heated. It would allow polymers which can't be used in the standard MEW syringes to be melt electrowritten.

## 6.7 FDM and MEW

With the combination of macrostructure in FDM and microstructure in MEW, new types of designs are possible and could be made in the same system without the need of additional processes.

# References

- [1] Navid T. Saidy et al. “Melt Electrowriting of Complex 3D Anatomically Relevant Scaffolds”. In: *Frontiers in Bioengineering and Biotechnology* 8 (2020). ISSN: 2296-4185. URL: <https://www.frontiersin.org/article/10.3389/fbioe.2020.00793> (visited on 03/29/2022).
- [2] Toby D. Brown et al. “Design and Fabrication of Tubular Scaffolds via Direct Writing in a Melt Electrospinning Mode”. en. In: *Biointerphases* 7.1 (Dec. 2012), pp. 1–16. ISSN: 1559-4106. DOI: 10.1007/s13758-011-0013-7. URL: <https://biointerphases.springeropen.com/articles/10.1007/s13758-011-0013-7> (visited on 03/29/2022).
- [3] Anne Metje van Genderen et al. “Topographic Guidance in Melt-Electrowritten Tubular Scaffolds Enhances Engineered Kidney Tubule Performance”. In: *Frontiers in Bioengineering and Biotechnology* 8 (2021). ISSN: 2296-4185. URL: <https://www.frontiersin.org/article/10.3389/fbioe.2020.617364> (visited on 03/31/2022).
- [4] Eve Hewitt et al. “Melt-electrowriting with novel milk protein/PCL biomaterials for skin regeneration”. eng. In: *Biomedical Materials (Bristol, England)* 14.5 (Aug. 2019), p. 055013. ISSN: 1748-605X. DOI: 10.1088/1748-605X/ab3344.
- [5] Gernot Hochleitner et al. “Melt Electrowriting of Thermoplastic Elastomers”. en. In: *Macromolecular Rapid Communications* 39.10 (2018), p. 1800055. ISSN: 1521-3927. DOI: 10.1002/marc.201800055. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/marc.201800055> (visited on 03/30/2022).
- [6] Benjamin T. O’Donnell et al. “Beyond the Present Constraints That Prevent a Wide Spread of Tissue Engineering and Regenerative Medicine Approaches”. In: *Frontiers*

- 
- in Bioengineering and Biotechnology* 7 (2019). ISSN: 2296-4185. URL: <https://www.frontiersin.org/article/10.3389/fbioe.2019.00095> (visited on 03/31/2022).
- [7] *3D bioprinting 2022: Comprehensive guide and bioprinter selection*. en-US. URL: <https://www.aniwaa.com/buyers-guide/3d-printers/3d-bioprinting-3d-bioprinters/> (visited on 03/03/2022).
- [8] Fredrick Madaraka Mwema and Esther Titilayo Akinlabi. “Basics of Fused Deposition Modelling (FDM)”. en. In: *Fused Deposition Modeling: Strategies for Quality Enhancement*. Ed. by Fredrick Madaraka Mwema and Esther Titilayo Akinlabi. Cham: Springer International Publishing, 2020, pp. 1–15. ISBN: 978-3-030-48259-6. DOI: 10.1007/978-3-030-48259-6\_1. URL: [https://doi.org/10.1007/978-3-030-48259-6\\_1](https://doi.org/10.1007/978-3-030-48259-6_1) (visited on 03/30/2022).
- [9] S. Scott Crump. “Apparatus and method for creating three-dimensional objects”. US5121329A. June 1992. URL: <https://patents.google.com/patent/US5121329A/en> (visited on 03/11/2022).
- [10] Bahaa Shaqour et al. “Gaining a better understanding of the extrusion process in fused filament fabrication 3D printing: a review”. en. In: *The International Journal of Advanced Manufacturing Technology* 114.5 (May 2021), pp. 1279–1291. ISSN: 1433-3015. DOI: 10.1007/s00170-021-06918-6. URL: <https://doi.org/10.1007/s00170-021-06918-6> (visited on 03/31/2022).
- [11] Jean-François Agassant et al. “Flow analysis of the polymer spreading during extrusion additive manufacturing”. en. In: *Additive Manufacturing* 29 (Oct. 2019), p. 100794. ISSN: 2214-8604. DOI: 10.1016/j.addma.2019.100794. URL: <https://www.sciencedirect.com/science/article/pii/S2214860419304932> (visited on 03/30/2022).
- [12] *Best Industrial FDM 3D Printers 2021*. en. May 2021. URL: <https://all3dp.com/1/best-industrial-fdm-3d-printers-2021/> (visited on 03/31/2022).
- [13] Tuan D. Ngo et al. “Additive manufacturing (3D printing): A review of materials, methods, applications and challenges”. en. In: *Composites Part B: Engineering* 143 (June 2018), pp. 172–196. ISSN: 1359-8368. DOI: 10.1016/j.compositesb.2018.02
-

- .012. URL: <https://www.sciencedirect.com/science/article/pii/S1359836817342944> (visited on 03/31/2022).
- [14] *Original Prusa i3 MK3S+ — Original Prusa 3D printers directly from Josef Prusa.* en. URL: [https://www.prusa3d.com/category/original-prusa-i3-mk3s/?gclid=Cj0KCQjw\\_4-SBhCgARIsAAIegrVYl67ucELm6IK4hOTgx713T76EWRKczDNCHsivBjVbfaKZQP8nJTgaAlvZEALw\\_wcB](https://www.prusa3d.com/category/original-prusa-i3-mk3s/?gclid=Cj0KCQjw_4-SBhCgARIsAAIegrVYl67ucELm6IK4hOTgx713T76EWRKczDNCHsivBjVbfaKZQP8nJTgaAlvZEALw_wcB) (visited on 03/30/2022).
- [15] *Official Creality3D® Store—Best DIY 3D Printer & Accessories, Filament.* zxx. URL: <https://www.creality3dofficial.com> (visited on 03/30/2022).
- [16] Pavan Kumar Penumakala, Jose Santo, and Alen Thomas. “A critical review on the fused deposition modeling of thermoplastic polymer composites”. en. In: *Composites Part B: Engineering* 201 (Nov. 2020), p. 108336. ISSN: 1359-8368. DOI: 10.1016/j.compositesb.2020.108336. URL: <https://www.sciencedirect.com/science/article/pii/S1359836820333850> (visited on 03/24/2022).
- [17] Gayathri Kollamaram et al. “Low temperature fused deposition modeling (FDM) 3D printing of thermolabile drugs”. eng. In: *International Journal of Pharmaceutics* 545.1-2 (July 2018), pp. 144–152. ISSN: 1873-3476. DOI: 10.1016/j.ijpharm.2018.04.055.
- [18] Seong Je Park et al. “High-temperature 3D printing of polyetheretherketone products: Perspective on industrial manufacturing applications of super engineering plastics”. en. In: *Materials & Design* 211 (Dec. 2021), p. 110163. ISSN: 0264-1275. DOI: 10.1016/j.matdes.2021.110163. URL: <https://www.sciencedirect.com/science/article/pii/S0264127521007188> (visited on 03/30/2022).
- [19] Thingiverse.com. *Thingiverse - Digital Designs for Physical Objects.* en. URL: <https://www.thingiverse.com/> (visited on 03/29/2022).
- [20] Yang Gao et al. “Watching and Safeguarding Your 3D Printer: Online Process Monitoring Against Cyber-Physical Attacks”. en. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.3 (Sept. 2018), pp. 1–27. ISSN: 2474-9567. DOI: 10.1145/3264918. URL: <https://dl.acm.org/doi/10.1145/3264918> (visited on 03/23/2022).

- 
- [21] Toby D. Brown, Paul D. Dalton, and Dietmar W. Hutmacher. “Direct Writing By Way of Melt Electrospinning”. In: *Advanced Materials* 23.47 (2011), pp. 5651–5657. ISSN: 1521-4095. DOI: 10.1002/adma.201103482. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201103482> (visited on 03/17/2022).
- [22] *The Next Frontier in Melt Electrospinning: Taming the Jet - Robinson - 2019 - Advanced Functional Materials - Wiley Online Library*. URL: <https://onlinelibrary.wiley.com/doi/full/10.1002/adfm.201904664> (visited on 03/30/2022).
- [23] Gernot Hochleitner et al. “Fibre pulsing during melt electrospinning writing”. In: *BioNanoMaterials* 17 (Jan. 2016). DOI: 10.1515/bnm-2015-0022.
- [24] Andrei Hrynevich, Ievgenii Liashenko, and Paul D. Dalton. “Accurate Prediction of Melt Electrowritten Laydown Patterns from Simple Geometrical Considerations”. en. In: *Advanced Materials Technologies* 5.12 (2020), p. 2000772. ISSN: 2365-709X. DOI: 10.1002/admt.202000772. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/admt.202000772> (visited on 03/29/2022).
- [25] Juliane C. Kade and Paul D. Dalton. “Polymers for Melt Electrowriting”. en. In: *Advanced Healthcare Materials* 10.1 (2021), p. 2001232. ISSN: 2192-2659. DOI: 10.1002/adhm.202001232. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adhm.202001232> (visited on 03/31/2022).
- [26] Qi Gao et al. “Tissue engineering of corneal stroma via melt electrowriting”. en. In: *Journal of Tissue Engineering and Regenerative Medicine* 15.10 (2021), pp. 841–851. ISSN: 1932-7005. DOI: 10.1002/term.3235. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/term.3235> (visited on 03/16/2022).
- [27] Navid Toosi Saidy et al. “Spatially Heterogeneous Tubular Scaffolds for In Situ Heart Valve Tissue Engineering Using Melt Electrowriting”. en. In: *Advanced Functional Materials* n/a.n/a (), p. 2110716. ISSN: 1616-3028. DOI: 10.1002/adfm.202110716. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.202110716> (visited on 03/16/2022).
- [28] Ferdows Afghah et al. “3D Fiber Reinforced Hydrogel Scaffolds by Melt Electrowriting and Gel Casting as a Hybrid Design for Wound Healing”. en. In: *Advanced Healthcare Materials* n/a.n/a (), p. 2102068. ISSN: 2192-2659. DOI: 10.1002/adhm.2021020
-

68. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adhm.202102068> (visited on 03/16/2022).
- [29] Arwa Daghrery et al. “Unveiling the potential of melt electrowriting in regenerative dental medicine”. en. In: *Acta Biomaterialia* (Jan. 2022). ISSN: 1742-7061. DOI: 10.1016/j.actbio.2022.01.010. URL: <https://www.sciencedirect.com/science/article/pii/S1742706122000101> (visited on 03/16/2022).
- [30] *G-code*. en. Page Version ID: 1079602597. Mar. 2022. URL: <https://en.wikipedia.org/w/index.php?title=G-code&oldid=1079602597> (visited on 03/29/2022).
- [31] *About - RepRap*. URL: <https://reprap.org/wiki/About> (visited on 03/09/2022).
- [32] Imgur. *Voron Prototype*. en. URL: <https://imgur.com/a/bVNhx> (visited on 03/03/2022).
- [33] *VORON 0.1*. URL: <https://vorondesign.com/voron0.1> (visited on 03/30/2022).
- [34] *Which Voron Printer To Build — 3D Lab Tech*. en-US. May 2021. URL: <https://www.3dlabtech.ca/which-voron-printer-to-build/> (visited on 03/03/2022).
- [35] *CoreXY — Cartesian Motion Platform*. URL: <https://corexy.com/theory.html> (visited on 03/10/2022).
- [36] Gina Häußge. *OctoPrint.org*. en. URL: <https://octoprint.org/> (visited on 03/03/2022).
- [37] *vitormhenrique - Overview*. en. URL: <https://github.com/vitormhenrique> (visited on 03/14/2022).
- [38] *Caproprene® 100M 3D Printing Filament*. en-US. URL: <https://www.poly-med.com/additive-manufacturing-technologies/caproprene-100m-3d-printing-filament/> (visited on 03/09/2022).
- [39] *Lab Mate High Voltage Power Supply — Superior 30 kV*. en-US. URL: <https://www.sprucescience.com/products/lab-mate-high-voltage-power-supply-10kv-2ma/> (visited on 03/11/2022).
- [40] *BME280*. en. URL: <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/> (visited on 03/09/2022).



- [41] Samuel Schlatter, Patrin Illenberger, and Samuel Rosset. “Peta-pico-Voltron: An open-source high voltage power supply”. en. In: *HardwareX* 4 (Oct. 2018), e00039. ISSN: 2468-0672. DOI: 10.1016/j.ohx.2018.e00039. URL: <https://www.sciencedirect.com/science/article/pii/S2468067218300531> (visited on 03/03/2022).

# Appendices

# Appendix A

## G-code generator parameters

Parameter
Start position X, Y, Z
Hotend temperature [C]
Feed rate [mm/min]
Length [mm]
Width [mm]
Spacing 1 [mm]
Spacing 2 [mm]
Number of layers
Set voltage [kV]
Loops [ON/OFF]
Loop sampling
Loop overlay [mm]
Pause before loop [ms]
Pause after loop [ms]
Feed rate in loop [mm/min]
Priming line [ON/OFF]
Priming space [mm]
Number of priming lines
Height added after layer [mm]

**Table A.1**

*List of all the parameters to define a scaffold*

# Appendix B

## G-code generator code in MATLAB

```
1  clc
2  clear
3
4  % don't forget to disable synchronus scrolling on the output window
5  % slide size : 75 x 51 [mm]
6
7  tic
8
9  time_date = datestr(now, 'dd.mm.yy-HHMMSS');
10 time_date_file = datestr(now, 'dd/mm/yy-HH:MM:SS');
11 author = 'TKangur';
12
13
14 % parameters
15 error_check = 1;
16 draw_buildplate_outline = 0;
17 touch = 0;
18 rotation = 0;
19 buildplate_tolerance = 5; % [mm] % tolerance on build plate size
20
21
22 % printer build size parameters
23 X_build = 120; % maximum X value of printer [mm]
24 Y_build = 120; % maximum Y value of printer [mm]
25 Z_build = 120; % maximum Z value of printer [mm]
26
27 % G-code start position
28 X_start = 19; % [mm]
29 Y_start = 19; % [mm]
30 Z_start = 3.2; % [mm]
31
32 % scaffold parameters
33 extruder_temp = 80; % [ C ] set the temperature of the nozzle while extruding
34 feed_rate = 300; % [mm/min]
35 length = 23; % length = Overal length in X [mm]
36 width = 23; % width = Overal width in Y [mm]
```

```

37 spacing_1 = 0.5;           % spacing between lines 1      [mm]
38 spacing_2 = 0.5;           % spacing between lines 2      [mm]
39 num_layers = 1;            % number of layers in the main scaffold
40 set_voltage = 4.8;         % set voltage of the high voltage power supply
41
42 % loops settings
43 loops = 1;                 % boolean to activate loops
44 loop_sampling = 10;        % how many points on the loops
45 overlay = 1;               % how much loops overlay on the next one
46 pause_before_loop = 600;   % pause time before the loop in microseconds
47 pause_after_loop = 500;    % pause time after the loop in microseconds
48 loop_speed = 300;          % speed at which the loop is done
49 Z_added = 0.00;           % increase in height after a layer is finished
50
51 % priming line settings
52 prime_line = 1;            % boolean to activate priming line
53 prime_space = 1;           % gap between priming line and the design [mm]
54 num_primeline = 2;         % number of prime line around the design
55
56 % output file name
57 output = ['scaf' ...
58     num2str(length, '%0.1f') , 'x', num2str(width, '%0.1f'), '—' ...
59     num2str(spacing_1, '%0.2f'), '-', num2str(spacing_2, '%0.2f') ...
60     '-layers_', num2str(num_layers, '%d'), '—', time_date, '.gcode'];
61
62 fileID = fopen(output, 'w');
63
64
65
66 % add scaffold parameter and creator to the start of the gcode
67 add_line(fileID, ['; author : ', author ]);
68 add_line(fileID, ['; date and time : ', time_date_file ]);
69 add_line(fileID, ['']);
70 add_line(fileID, ['; extruder temperature : ', num2str(extruder_temp,2) , ' [ C ]',]);
71 add_line(fileID, ['; feed rate : ', num2str(feed_rate) , ' [mm/min]');
72 add_line(fileID, ['; print volume X,Y,Z : ', num2str(X_build),',', ', num2str(Y_build),',', ',
73     num2str(Z_build) , ' [mm] ',]);
74 add_line(fileID, ['; scaffold size in X , Y : ', num2str(length,2),',', ', num2str(width,2) , ' [mm]
75     ']);
76 add_line(fileID, ['; start position X, Y, Z : ', num2str(X_start),',', ', num2str(Y_start),',', ',
77     num2str(Z_start), ' [mm] ']);
78 add_line(fileID, ['; spacings : ', num2str(spacing_1,3) , ', ', ', num2str(spacing_2,3) , ' [mm]']);
79 add_line(fileID, ['; number of layers : ', num2str(num_layers) ]);
80 add_line(fileID, ['; added Z height after layer : ', num2str(Z_added), ' [mm]']);
81 add_line(fileID, ['; start voltage : ', num2str(set_voltage), ' [kV]']);
82 add_line(fileID, ['']);
83
84 if (loops)
85     add_line(fileID, ['; loops : ', num2str(loops) ]);
86     if (loops)
87         add_line(fileID, ['; loop sampling : ', num2str(loop_sampling)]);
88         add_line(fileID, ['; loop speed : ', num2str(loop_speed), ' [mm/min] ']);
89         add_line(fileID, ['; loop overlay : ', num2str(overlay) , ' [mm]']);
90         add_line(fileID, ['; pause before and after loop : ', num2str(pause_before_loop),',', ',
91             num2str(pause_after_loop) , ' [um] ']);
92         add_line(fileID, ['']);
93     end
94
95 add_line(fileID, ['; primeline : ', num2str(prime_line) ]);
96 if prime_line

```

```

92     add_line(fileID, ['; primeline space : ', num2str(prime_space), ' [mm]' ]);
93     add_line(fileID, ['; number of primelines : ', num2str(num_primeline) ]);
94 end
95 add_line(fileID, ['']);
96 add_line(fileID, ['; Start of the G-code _____']);
97 add_line(fileID, ['']);
98
99
100 Scaffold Designs
101 PrimeS = scaffold(length, 5, spacing_1, spacing_2, 2, feed_rate, loops,
    loop_sampling, overlay, X_start - 0.8*width, Y_start, Z_start ,2);
102 F = scaffold(length, width, spacing_1, spacing_2, 1, feed_rate, loops,
    loop_sampling, overlay, X_start, Y_start, Z_start, 1);
103 K = scaffold(length, width, spacing_1, spacing_2, num_layers, feed_rate, loops,
    loop_sampling, overlay, X_start, Y_start + 1.3*width, Z_start, 1);
104
105 start_coor = [X_start, Y_start];
106
107 % Put the design into P
108 P = [start_coor; PrimeS; F; K];
109
110
111
112
113 add_line(fileID, ['M109 S' num2str(extruder_temp, '%d') ' ; set temperature of nozzle and wait
    until reached']);
114
115
116 add_line(fileID, 'M220 S100 ; set feedrate to 100%');
117
118 add_line(fileID, 'G90 ; absolute positioning');
119 add_line(fileID, 'G1 F3000 ; speed to get into postion');
120 add_line(fileID, ['G1 ' ...
121     'X' num2str(X_start, '%0.3f') ...
122     ' Y' num2str(Y_start, '%0.3f') ...
123     ' Z' num2str(Z_start, '%0.3f') ...
124     '; set start position']);
125 add_line(fileID, ['ENC O5 ' ...
126     'R' num2str(set_voltage*100, '%d') ...
127     ' G1 ' ...
128     'B0 ; set voltage to set amount ']);
129
130
131 add_line(fileID, 'SET_LED LED=MY_NEOPIXEL RED=0.2');
132 add_line(fileID, 'ENC O3 S1 ; Turn voltage on');
133 add_line(fileID, 'ENC O4 S1 ; Turn remote control on');
134
135 if prime_line
136     % locate the corners of the scaffolds
137     add_line(fileID, '; priming line');
138     X_min = min(P(:,1));
139     Y_min = min(P(:,2));
140     X_max = max(P(:,1));
141     Y_max = max(P(:,2));
142     prime_X_bot = X_min - prime_space;
143     prime_X_top = X_max + prime_space;
144     prime_Y_bot = Y_min - prime_space;
145     prime_Y_top = Y_max + prime_space;
146

```

```

147     prime_top_right = [prime_X_top, prime_Y_top];
148     prime_top_left  = [prime_X_bot,  prime_Y_top];
149     prime_bot_right = [prime_X_top,  prime_Y_bot];
150     prime_bot_left  = [prime_X_bot,  prime_Y_bot];
151
152     Prime = [    prime_bot_left ,
153              prime_bot_right ,
154              prime_top_right ,
155              prime_top_left ,
156              prime_bot_left ];
157
158     gcode_0 = ['G1 X', num2str(prime_X_bot), ' Y', num2str(prime_Y_bot), ' F3000'];
159     gcode_1 = ['G1 X', num2str(prime_X_bot), ' Y', num2str(prime_Y_top), ' F3000'];
160     gcode_2 = ['G1 X', num2str(prime_X_top), ' Y', num2str(prime_Y_top), ' F3000'];
161     gcode_3 = ['G1 X', num2str(prime_X_top), ' Y', num2str(prime_Y_bot), ' F3000'];
162
163     disp(gcode_0)
164     disp(gcode_1)
165     disp(gcode_2)
166     disp(gcode_3)
167
168     for i=1:num_primeline
169         F = [Prime; F];
170     end
171 end
172
173 P = [PrimeS; F; K];
174
175
176 add_gcode_path(P,    feed_rate, fileID);
177
178 add_line(fileID, 'G1 Z20 ; print finished , increase height by 20 mm to be far from collector')
179
180
181 add_line(fileID, 'M104 S0 ; turn off extruder temperature')
182 add_line(fileID, 'ENC O5 R0 G0 B0 ; set voltage to 0kV')
183 add_line(fileID, 'ENC O3 S0 ; Turn voltage off')
184 add_line(fileID, 'ENC O4 S0 ; Turn remote control off')
185 add_line(fileID, 'SET_LED LED=MY_NEOPIXEL GREEN=0.2')
186
187
188 % closes the file so it can be modified by other programs while matlab is
189 % running
190 fclose(fileID);
191
192 % check if printing outside of build plate
193 if error_check
194     if (max(P(:,1))+ X_start > X_build) || (max(P(:,2)) + Y_start > Y_build )
195         disp('Error : will be printed outside printer build plate')
196         error()
197     elseif ((abs(X_build - (max(P(:,1)) + X_start)) < buildplate_tolerance) ...
198            || (abs(Y_build - (max(P(:,2)) + Y_start)) < buildplate_tolerance))
199         warning(['Be careful with placement: close to buildplate max size' ...
200                ' Max X : ' num2str(max(P(:,1)),'%d') ...
201                ', Max Y : ' num2str(max(P(:,2)),'%d')])
202     end
203 end
204
205 % change the color of the plot

```

---

```

206
207 %color = jet(size(P,1));
208 color = parula(size(P,1));
209
210 % searches the output file and changes the lines to other lines
211
212 % add different pauses and loop speed by changing the output file
213 fileID = [output];
214 OutputFile = [output];
215
216 spacing = [spacing_1, spacing_2];
217 voltage_increase_path = (width + 7*overlay + max(spacing));
218
219 SearchString = [{'G1 X'          num2str(length, '%d') '.00000 Y0.00000'},
220                ['G1 X-'        num2str(length, '%d') '.00000 Y0.00000'],
221                ['G1 X0.00000 Y' num2str(width, '%d') '.00000'],
222                ['G1 X0.00000 Y-' num2str(width, '%d') '.00000'],
223                ['G1 X0.00000 Y-' num2str(voltage_increase_path, '%0.5f')]];
224
225
226 ReplaceString = [{'G4 P' num2str(pause_after_loop, '%d') ' ; pause before line \r\n'...
227                  'G1 F' num2str(feed_rate, '%d') ' ; MEWing speed \r\n'...
228                  SearchString {1}...
229                  '\r\n'...
230                  'G4 P' num2str(pause_before_loop, '%d') ' ; pause after line \r\n'...
231                  'G1 F' num2str(loop_speed, '%d') ' ; loop speed'},
232
233                  ['G4 P' num2str(pause_after_loop, '%d') ' ; pause before line \r\n'...
234                  'G1 F' num2str(feed_rate, '%d') ' ; MEWing speed \r\n'...
235                  SearchString {2}...
236                  '\r\n'...
237                  'G4 P' num2str(pause_before_loop, '%d') ' ; pause after line \r\n'...
238                  'G1 F' num2str(loop_speed, '%d') ' ; loop speed'},
239
240                  ['G4 P' num2str(pause_after_loop, '%d') ' ; pause before line \r\n'...
241                  'G1 F' num2str(feed_rate, '%d') ' ; MEWing speed \r\n'...
242                  SearchString {3}...
243                  '\r\n'...
244                  'G4 P' num2str(pause_before_loop, '%d') ' ; pause after line \r\n'...
245                  'G1 F' num2str(loop_speed, '%d') ' ; loop speed'},
246
247                  ['G4 P' num2str(pause_after_loop, '%d') ' ; pause before line \r\n'...
248                  'G1 F' num2str(feed_rate, '%d') ' ; MEWing speed \r\n'...
249                  SearchString {4}...
250                  '\r\n'...
251                  'G4 P' num2str(pause_before_loop, '%d') ' ; pause after line \r\n'...
252                  'G1 F' num2str(loop_speed, '%d') ' ; loop speed'},
253
254                  ['G1 Z' num2str(Z_added, '%0.2f') ' ; added Z height after layer \r\n'...
255                  'ENC O5 R480 G1 B0 ; increase in voltage \r\n'...
256                  SearchString {5}
257                  ]};
258
259 ReplaceString = compose(ReplaceString);
260
261 for i=1:size(SearchString,1)
262     func_replace_string(fileID, OutputFile, SearchString(i), ReplaceString(i));
263 end
264

```



```

265
266 % plotting on graph
267
268 for i=1:size(P,1)-1
269     plot(P(i:i+1,1),P(i:i+1,2), 'Color',color(i,:));
270     hold on
271 end
272
273 if draw_buildplate_outline
274     O(1,:) = [0,0];
275     O(2,:) = [X_build,0];
276     O(3,:) = [X_build,Y_build];
277     O(4,:) = [0,Y_build];
278     O(5,:) = [0,0];
279     for i=1:size(O,1)-1
280         plot(O(i:i+1,1),O(i:i+1,2), 'black');
281         hold on
282     end
283 end
284
285 % show the starting point as a red filled circle
286 plot(P(1,1),P(1,2), '.r', 'MarkerSize',20);
287 hold on
288
289 xlabel('X axis [mm]')
290 ylabel('Y axis [mm]')
291 axis equal
292 axis padded
293 hold off
294 toc
295
296
297
298 function add_line(file_path,string)
299     % add line with newline at the end to outputfile
300     fprintf(file_path, '%s' , string);
301     fprintf(file_path, '\r\n');
302 end
303
304 function add_gcode_path(absolute_path, feedrate, file_name)
305     add_line(file_name, '; new path')
306     relative_path(1,:) = [0,0];
307
308     % skip the first value as it is the start coordinate in absolute values
309     % change the absolute path to a relative path by subtracting the
310     % current position by the one before
311     for i=2:size(absolute_path,1)
312         relative_path(end+1,:)= [absolute_path(i,1)-absolute_path(i-1,1),
313                                 absolute_path(i,2)-absolute_path(i-1,2)];
314         i=i+1;
315     end
316
317
318     % remove move positions like G1 X0 Y0 to limit filing the command buffer
319     relative_path_clean = [];
320     for i=1:size(relative_path,1) - 1
321         if (relative_path(i,1) + relative_path(i,2) ~= 0)
322             relative_path(i,:);
323             relative_path_clean(end+1, :) = relative_path(i, :);

```

```

324     end
325 end
326 relative_path = relative_path_clean;
327
328 % set the speed while mewing
329 add_line(file_name, ['G1 F' num2str(feedrate, '%d') ' ; set feedrate'])
330
331 % writes the Gcode commands into the .gcode file
332 for i=1:1:size(relative_path,1)
333
334     gt=['G1 X' num2str(relative_path(i,1), '%.5f') ' Y' num2str(relative_path(i,2), '%.5f')];
335
336     if i == 1
337         add_line(file_name, 'G91 ; relative positioning')
338     end
339
340     fprintf(file_name, '%s' ,gt);    fprintf(file_name, '\r\n');
341
342 end
343
344 end
345
346 function absolute_gcode_path(absolute_path, feedrate, file_name)
347     add_line(file_name, ' ; position with absolute values')
348     add_line(file_name, ['G1 F' num2str(feedrate, '%d') ' ; set feedrate'])
349     add_line(file_name, 'G90 ; absolute positioning')
350     for i=1:1:size(relative_path,1)
351
352         gt=['G1 X' num2str(relative_path(i,1), '%.5f') ' Y' num2str(relative_path(i,2), '%.5f')];
353
354         fprintf(file_name, '%s' ,gt);    fprintf(file_name, '\r\n');
355     end
356
357 end
358
359 function [X, Y] = loop_coordinates(radius, sampling, overlay, reverse)
360
361     q = asin((radius/2)/(radius+overlay));
362
363     if reverse
364         theta = linspace(pi-q, -(pi-q), sampling);
365     else
366         theta = linspace(-(pi-q), pi-q, sampling);
367
368     end
369
370     A = (radius+overlay) * cos(theta);
371     B = (radius+overlay) * sin(theta);
372
373     for i=1:1:sampling
374         if i < sampling
375             X(:,i) = A(i+1)-A(i);
376             Y(:,i) = B(i+1)-B(i);
377         else
378             X(:,i) = 0;
379             Y(:,i) = 0;
380         end
381     end
382 end

```

```

383
384
385
386
387 function path = scaffold(size_x, size_y, spacing_1, spacing_2, layers, feed, loops, loop_samp,
    overlay, start_X, start_Y, start_Z, rotation)
388
389     t=0; %line number from bottom to top in Y
390
391     X0 = - size_x / 2;
392     Y0 = - size_y / 2;
393
394     path(1,:)=[X0,Y0];
395
396
397     while t < abs((size_y/spacing_1)/2) % total number of lines along Y axis
398
399         path(end+1,:)=[X0,Y0+spacing_1*(t)]; %left to right lines
400         path(end+1,:)=[X0+size_x ,Y0+spacing_1*(t)];
401
402         if loops
403             [X_C, Y_C] = loop_coordinates(spacing_1, loop_samp, overlay, 0);
404
405             for i=1:loop_samp
406                 path(end+1,:)=[path(end,1)+X_C(1,i),
407                             path(end,2)+Y_C(1,i)];
408             end
409         end
410
411         %rectangular area : right to left lines
412         for i=0:size_x:size_x
413             x=size_x-i;
414             path(end+1,:)=[X0+x,Y0+spacing_1*(t+1)];
415             %path(end+1,:)=[X0,Y0+spacing_1*(t+1)];
416         end
417
418
419
420         if loops
421             [X_C, Y_C] = loop_coordinates(spacing_1, loop_samp, overlay, 0);
422             for i=1:loop_samp
423                 path(end+1,:)=[path(end,1)-X_C(1,i),
424                             path(end,2)+Y_C(1,i)];
425             end
426         end
427
428         t=t+2;
429     end
430
431     %-----
432
433
434
435     t=0; %line number from bottom to top in Y
436     while t<abs((-spacing_1+size_y/spacing_2)/2) % total number of lines along Y axis
437
438         for i=0:size_x:size_x           %rectangular area
439             path(end+1,:)=[X0+i, spacing_1+spacing_2*(t)]; %left to right lines
440         end

```

```

441
442     if loops
443         [X_C, Y_C] = loop_coordinates(spacing_2, loop_samp, overlay, 0);
444         for i=1:loop_samp
445             path(end+1,:)=[path(end,1)+X_C(1,i),
446                           path(end,2)+Y_C(1,i)];
447         end
448     end
449
450     for i=0:size_x:size_x           %rectangular area
451         x=size_x-i;
452         path(end+1,:)=[X0+x,spacing_1+spacing_2*(t+1)]; %right to left lines
453     end
454
455     if loops
456         [X_C, Y_C] = loop_coordinates(spacing_2, loop_samp, overlay, 0);
457         for i=1:loop_samp
458             path(end+1,:)=[path(end,1)-X_C(1,i),
459                           path(end,2)+Y_C(1,i)];
460         end
461     end
462
463     t=t+2;
464 end
465
466 %%%% ROTATING COORDINATES %%%%
467 if rotation == 1
468
469     thetal=-90;                               %TO ROTATE CLOCKWISE BY X
470     DEGREES
471
472     R=[cosd(thetal) -sind(thetal);
473       sind(thetal)  cosd(thetal)];
474
475     P1=path*R';                               %MULTIPLY VECTORS BY THE
476     ROT MATRIX
477
478     path=[path;P1];                           % appending 2 layers with
479     different angles
480
481 end
482
483 if rotation == 2
484
485     thetal=90;                                %TO ROTATE CLOCKWISE BY X
486     DEGREES
487
488     R=[cosd(thetal) -sind(thetal);
489       sind(thetal)  cosd(thetal)];
490
491     P1=path*R';                               %MULTIPLY VECTORS BY THE
492     ROT MATRIX
493
494     path=[P1];                               % appending 2 layers with different
495     angles
496
497 end

```

```

494
495
496     spacing = [spacing_1, spacing_2];
497
498     if layers == 1
499         % linking to next layer without going through the scaffold
500         path(end+1,:)=[path(end,1), path(end,2) + (3*overlay + max(spacing))];
501         path(end+1,:)=[path(end,1) - (size_x + 6*overlay + max(spacing)) , path(end,2)];
502         %path(end+1,:)=[path(end,1), path(end,2) - (size_y + 7*overlay + max(spacing))];
503     end
504
505
506 % number of layers
507 if layers > 1 & rotation == 1
508     new_layer = path;
509     for i=1:layers-1
510
511         % linking to next layer without going through the scaffold
512         path(end+1,:)=[path(end,1) + (4*overlay + max(spacing)), path(end,2) + (4*overlay +
                    max(spacing))];
513         path(end+1,:)=[path(end,1) - (size_x + 10*overlay + max(spacing)) , path(end,2)];
514         path(end+1,:)=[path(end,1), path(end,2) - (size_y + 7*overlay + max(spacing))];
515
516         % adding new layer
517         path = [path; new_layer];
518     end
519 end
520
521 if layers > 1 & rotation == 0
522     new_layer = path;
523     for i=1:layers-1
524
525         % linking to next layer without going through the scaffold
526         %path(end+1,:)=[path(end,1) - size_y, path(end,2)];
527         %path(end+1,:)=[path(end,1), path(end,2) - size_x];
528
529         % adding new layer
530         path = [path; new_layer];
531     end
532 end
533
534 % shifting whole scaffold away from zero coordinate
535 path=[path+size_x/2, path+size_y/2];
536
537 % shifting whole scaffold away to the enter start coordinate
538 path=[path(:,1)+start_X, path(:,2)+start_Y];
539
540 end
541
542
543 function func_replace_string(InputFile, OutputFile, SearchString, ReplaceString)
544     %%change data [e.g. initial conditions] in model file
545     % InputFile - string
546     % OutputFile - string
547     % SearchString - string
548     % ReplaceString - string
549     % read whole model file data into cell array
550     fid = fopen(InputFile);
551     data = textscan(fid, '%s', 'Delimiter', '\n', 'CollectOutput', true);

```

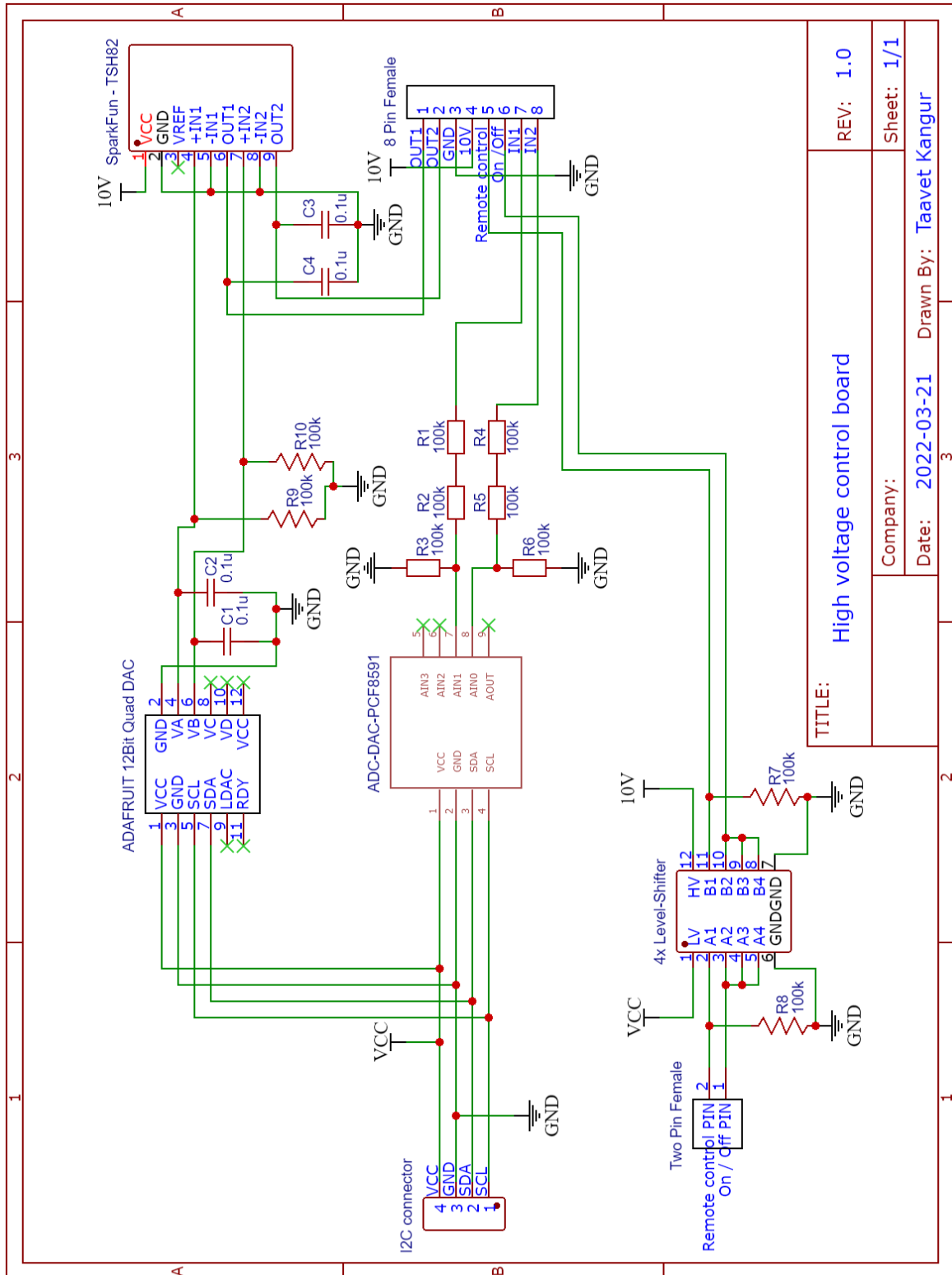
```

552     fclose(fid);
553     % modify the cell array
554     % find the position where changes need to be applied and insert new data
555     for I = 1:length(data{1})
556         tf = strcmp(data{1}{I}, SearchString); % search for this string in the array
557         if tf == 1
558             data{1}{I} = ReplaceString; % replace with this string
559         end
560     end
561     % write the modified cell array into the text file
562     fid = fopen(OutputFile, 'w');
563     for I = 1:length(data{1})
564         fprintf(fid, '%s\n', char(data{1}{I}));
565     end
566     fclose(fid);
567 end

```

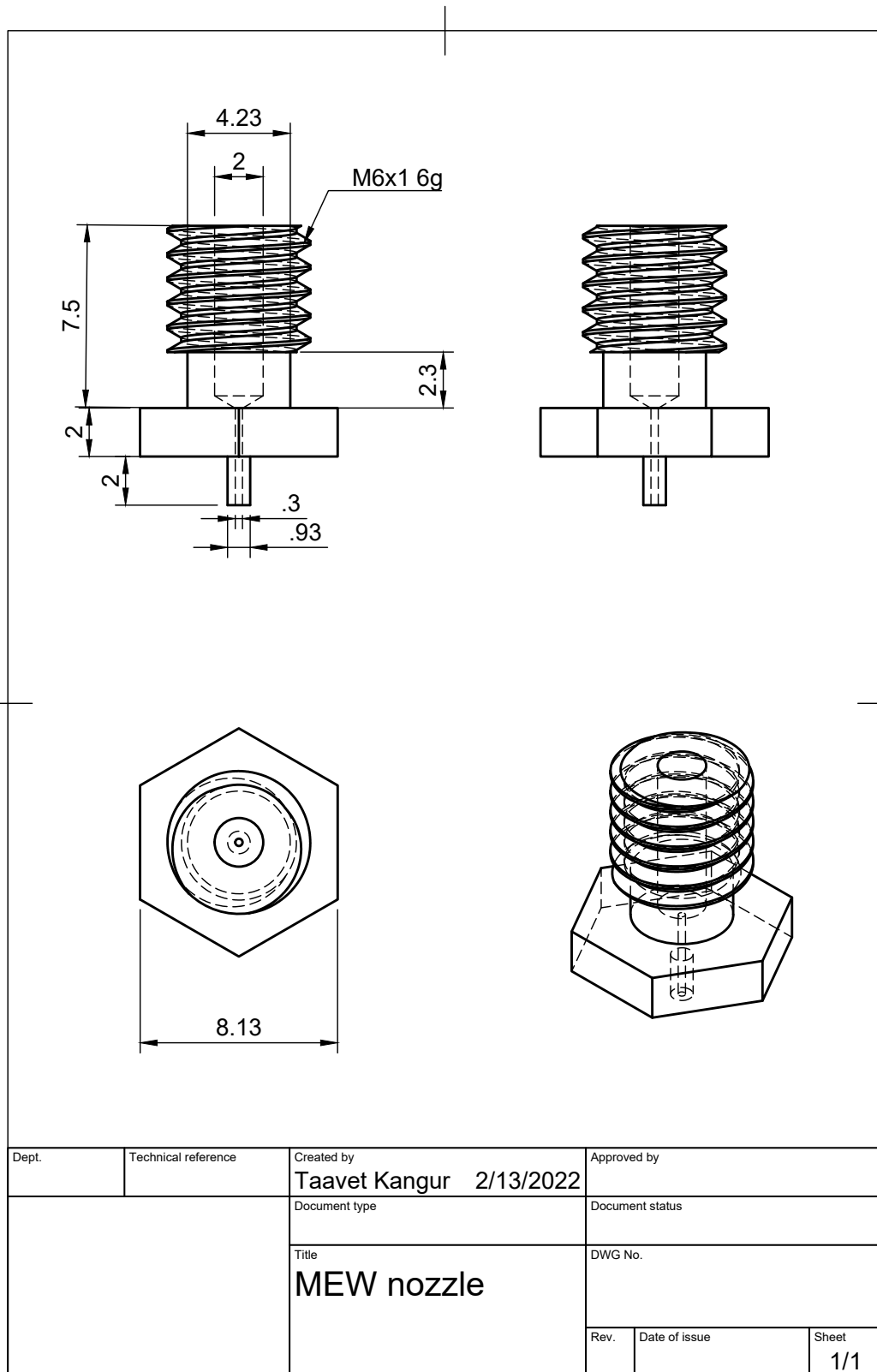
## Appendix C

# Technical Drawings



**Figure C.1**  
*Electric schematic for the high voltage control board*





**Figure C.2**  
*Schematic of the machined nozzle for MEW*