



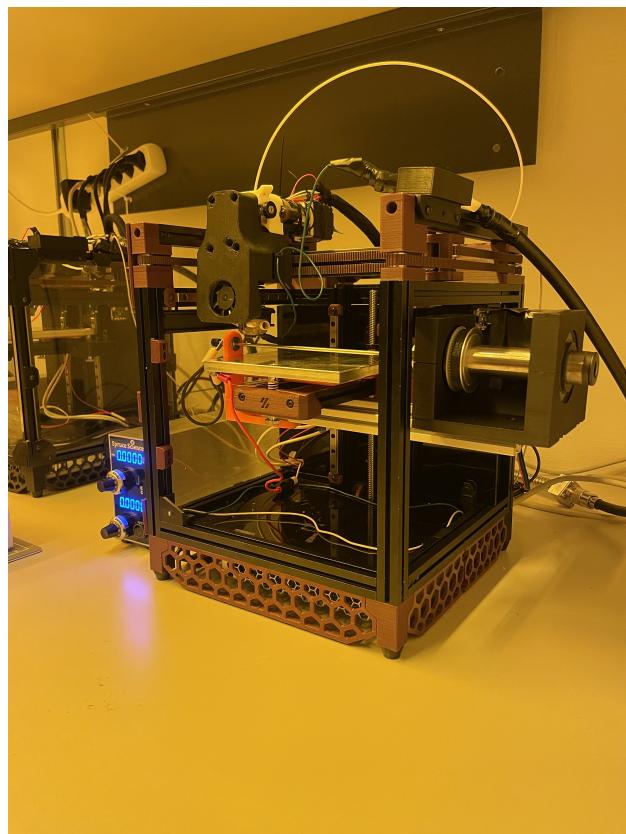
---

# Melt Electrowriting Report

---

SUMMER INTERNSHIP

SUMMER 2023



Alessandro Zedgitt

August 31, 2023

## 1 Introduction

Additive manufacturing (AM) comprises of rapidly evolving technologies which are easy to use and provide highly tailorable options to manufacture custom and complex design parts using polymers, ceramics, metals, etc.. Fused deposition modelling (FDM) is an example of AM technique used to make 3D objects using extrusion of fibres processed from polymer melts. To make thinner fibres than FDM, one can use electrospinning. This method relies on applying a high voltage to a polymer melt or solution in order to manufacture sub-micron fibres. The fibres produced are deposited on either a flat collector or a tubular rotating collector and the main disadvantage of this technique is the lack of control on the deposition of the fibres. By combining the principles of FDM and electrospinning, melt electrowriting (MEW) was created. This process works exactly like a regular FDM printer, with the addition of a high voltage applied between the nozzle and the collector. A significant difference between the two technologies is that in MEW, unlike FDM, introduces a gap (up to 10 millimetres) between the nozzle and the collector. The commercially available machines are expensive and the user has limited access to the tool for upgrades, so some researchers decided to use an already existing FDM open source system and convert it to a MEW capable tool [1].

The first iteration was designed upon a Voron 0.1 platform [2]. Upon the design two heads were developed: one with a syringe to dispense a liquid polymer melt and one that could process solid polymer filaments. The Microsystems Laboratory (LMIS1) at EPFL is equipped with four similar tools. The first three are equipped with the filament head, the fourth is equipped with the syringe extrusion head. The last machine has undergone an upgrade, conducted this summer, that added an extra axis to the tool. This upgrade is known as the tubular collector. This was done to print tubular scaffolds, to expand the catalogue of possible patterns. At the time of writing the tubular collector is operational, the next steps would be to push the limits of the technology in the tool, such as the minimum possible feature that can be processed, the fibre diameter or improve the complexity of the scaffold.

## 2 Hardware

The tubular collector is an add-on to a regular Voron 0.1 which is bolted into position underneath the print bed. This add-on assembly is designed to be removable, however in practice the upgrade is permanent. The chuck coupled to the motor through a belt drive is very bulky and places the Voron print bed under unnecessary stress. For the duration of the internship, no failures due to the tubular collector were detected.

The assembly takes advantage of the bearings to facilitate the rotation of the chuck as well as the shaft attached to it. The shaft is maintained in place by a collet inserted

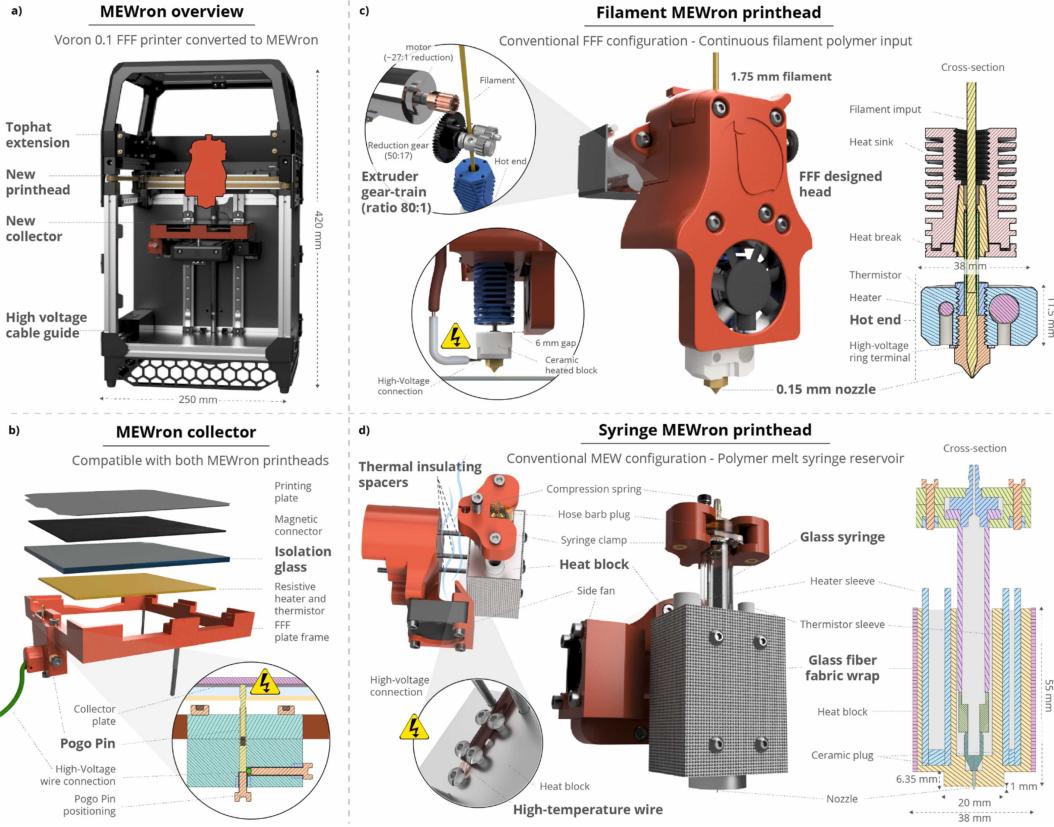


Figure 1: a) The Voron 0.1 system is shown with a protective enclosure. b) This figure shows the designed flat collector, it varies from the base model because high voltage is applied to it. Thus, electrical insulation is required. c) The filament capable print head is displayed. Some features were adapted whilst testing the design. d) This is the designed syringe. This figure was taken from MEWron: An open-source melt electrowriting platform [1]

inside the chuck. Custom ABS parts were made to ensure parallelism of the collector, they were made in-house using a FDM printer. The designs vary from the CAD model to fit the physical printer. Those changes mainly revolved around adjusting the diameters and distances between the parts. The extrusions were added for mechanical rigidity and serve as an anchor point for the printed parts. The extrusions are used to mount the collector on the printer by using screws. Multiple improvements can be made to make the tool more compact and reduced the stress applied on the print bed, such as placing the motor and the chuck under the bed, reducing the size of the chuck.

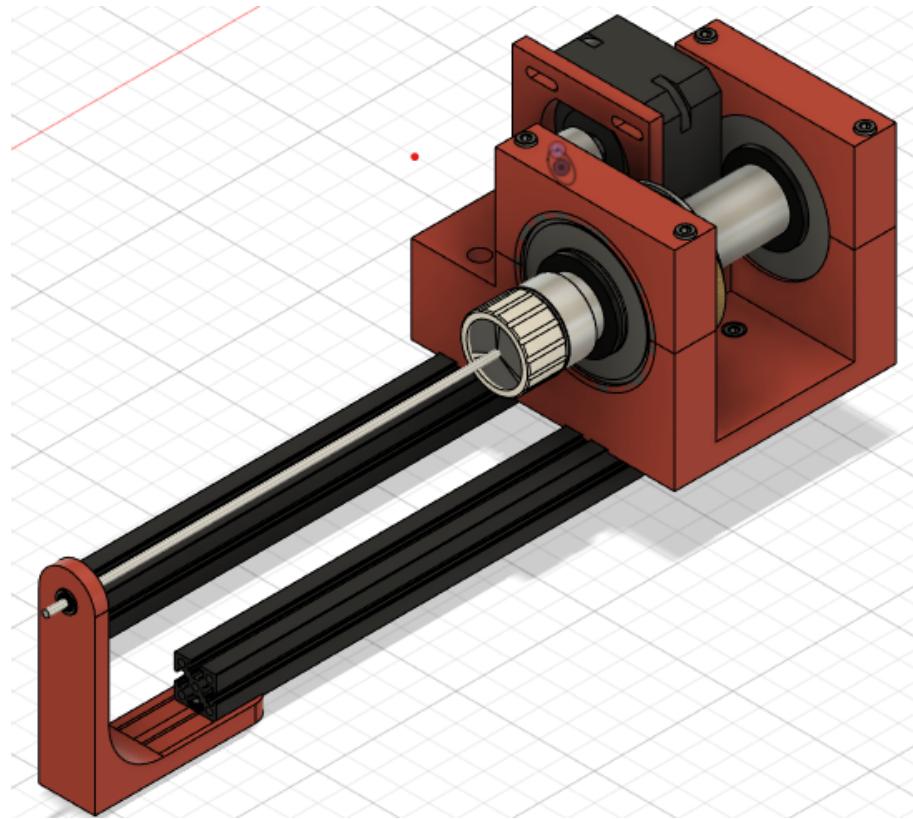


Figure 2: The tubular collector for the MEWron. The picture was made by screenshotsing the CAD model.

### 3 Software

The tool is controlled by multiple parts of software, each of which provided control over different parts of the machine. Their use is discussed in this section. The documentation is readily accessible on the internet and the forums contain useful solutions. If a problem should occur, it is recommended to consult the documentation as well as the forums. The installation requires the terminal so limited knowledge in using it is useful.

#### 3.1 Klipper

Once the hardware was assembled, the electronics were wired. The basic printer peripherals are all wired to the controller board (Fysetc spider). This controller board is configured by the klipper firmware. Despite the spider board being specifically designed for the Voron printers, the process wasn't explicitly described for the version used (2.3). When an issue occurred, the documentation generally provided the solution. The installation instructions provided by Voron [3] for the software installation were followed. As the task was to upgrade an existing system, the installation of Octoprint was skipped.

The board was also upgraded to the Fysetc spider V2.3, this process is described also on the Voron website.

In order to flash the board, we need to make the firmware straight from the raspberry pi. The process described by the Voron website is reliable and does not need any addition unless you have problems. In the event of a problem, please consult the documentation or the forums for the solution, generally issues aren't unique and someone may have posted the solution.

Once the firmware is acquired, it needs to be flashed onto the spider board. This is achieved by placing the board in DFU (device firmware upgrade) mode. The following pins need to be shorted : the first pair BT0/3.3 V and the second 5VDC/5V. Then the controller board should be connected to the raspberry pi by usb. Once the board is powered on, the board is in DFU mode. To connect through ssh to the raspberry pi, the following command should be executed in the terminal: `ssh user@hostname.local` then enter the password when prompted. Once completed, the command `cd ~ klipper` is executed, this set the terminal to work in the klipper folder. The `lsusb` displays a list of all the detected usb peripherals attached to the raspberry pi. Each device is assigned a unique identifier of the form 1234:5678, once that number is found, the flash process can begin. The command used is: `make flash FLASH_DEVICE=1234:5678`. Once the task is finished, the board needs to be powered off and the jumper cables used for shorting the pins can be removed. There is a simple verification that can be done to see whether the process succeeded, `ls /dev/serial/by-id` lists the flashed devices with a Klipper firmware. If the process failed, the bootloader needs reloading. To complete this task, the STM32CubeProgrammer [4] software is required. The board in DFU mode is connected to the work computer, select the board from the recognised devices. The bootloader needs to be flashed with the bootloader file found on the official GitHub page [5]. The instructions provided worked without any changes for this version of the board.

Once the board is flashed, the motors, heaters, end stops, etc. needs to be configured. In order to do so, the recommended way is to copy the `printer.cfg` file out of the raspberry pi using WinSCP [6], after editing the file can be recopied onto the pi using the same software. If the file needs to be lightly edited, then the preferred method is logging into the printer by ssh and then issuing a `nano printer.cfg` command. This opens the file in edit mode, then after saving the work (`ctrl + S`), the file can be closed by pressing `ctrl+X`.

## 3.2 Octoprint

Octoprint is the printer user interface, that is released under the Affero General Public License (AGPL). This means that the software is free to distribute and modify. However, once the new software is released to the public, the source code of the modified version

needs to be released to the public [7]. In our system, we use the regular version and we control the extruder via the enclosure plugin.

The enclosure plugin [8] was modified in collaboration with the developer, in order to run custom g-code straight from the Raspberry pi GPIO (general purpose inputs and outputs). The command has this syntax, for example ENC 03 R0 G0 B0, turns the extruder off. The speed rate can be controlled through the value given in the flow speed section. To increase the speed, we can go into the plugin settings and change the value of LEDs assigned to the extruder output. The maths behind the extrusion rate is explained in the supplementary material of the mewron article in the section S2.3.2 [1]. The link between the extrusion rate and the values entered in octoprint is simple: Extrusion rate = LED  $\times$  steps.

The other plugin that was installed is the Simple Emergency Stop [9]. This enables a software emergency stop that interrupts the motor motion when pressed. The button issues the M112 g-code to the printer. This command shuts down the motors and heaters, to resume operation, the user must reset the micro controller [10]. As emergency stops are regulated devices, this can't be considered as such. However, we can refer to this as a full stop button with no redundancy built-in.

### 3.3 G-code generator

To produce a reliable and repeatable pattern, the g-code instructions should be generated by software as described below. It was developed on python version 3.10. It uses some standard libraries such as *datetime* to get the current time, *math* for the definition of  $\pi$  and *sys* to control the execution in case of an error. Thus there is no need to install external packages using pip. The program is constituted by three different files: main.py, g\_code\_pipe.py and constants.py. They each have their use and together, they take the system and material parameters as well as the pattern settings to produce the required g-code.

#### 3.3.1 Patterns

The patterns that are currently coded are the tube, the pipe and the alignment channel. The custom pattern is a fully controllable pattern that may or may not be constructed using the previous patterns. The tube pattern arranges the fibres in a spiral around the rod, as shown on figure 4. The tube pattern, when printed with multiple layers, will stack the fibres on top of each other.

The pipe pattern is constructed by two alternating layers: the straight channels and the tube pattern. The straight channels are printed first for a single layer, then they are wrapped by the tube pattern as defined above. For multiple layers, the pair will be stacked one above the other until the correct number of layers is reached.

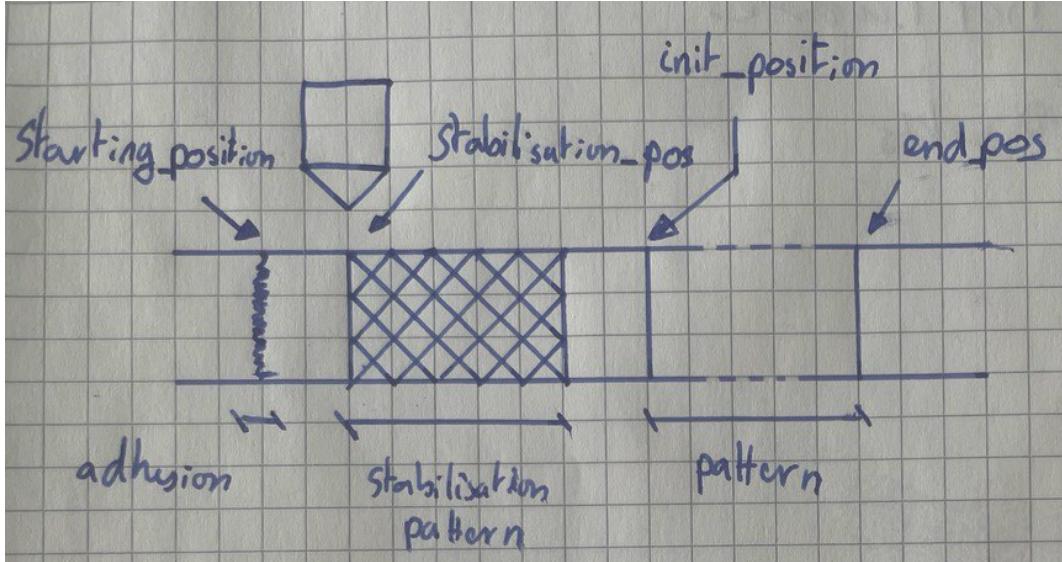


Figure 3: This is the generic print template. The print starts with the adhesion of the fibre on the shaft, then the fibre is stabilised by printing a 5 mm long tube pattern. The pattern selected is then printed.

The last pattern is the alignment channels. The layers define the number of straight fibres that are piled up. The heap of fibres is then wrapped around by a tube to ensure structure stability.

The code is available on GitHub as a repository and contains further instructions on how to operate the code [11]. This software was tested on an already established material as we are testing the software's capabilities before using it on a new polymer as described in the materials section.

### 3.3.2 Relation between the speeds

The main challenge of the code is to synchronise the rotation of the mandrel and the movement of the tool-head. The printer firmware klipper is equipped with such a function [12]. For our application, the shaft motor was defined as a manual stepper that can be controlled independently from the rest of the printer. The main idea is to complete a rotation and move the nozzle on the length of a single spiral ( $x_{pitch}$ ). With this principle, we can then define the following system of equations:

$$\begin{cases} v_x \times t_{onerevolution} = x_{pitch} \\ v_\theta \times t_{onerevolution} = 2 \times \pi \times R_{mandrel} \end{cases} . \quad (1)$$

From equation 1, a relation between the rotation speed  $v_\theta$  and the linear speed  $v_x$  is given:  $v_\theta = \frac{v_x \times 2 \times \pi \times R_{mandrel}}{x_{pitch}}$ . There exists a maximum speed for the shaft, such that the fibre remains in place. In our experiments, the maximum speed is of about 0.25 mm/s until the

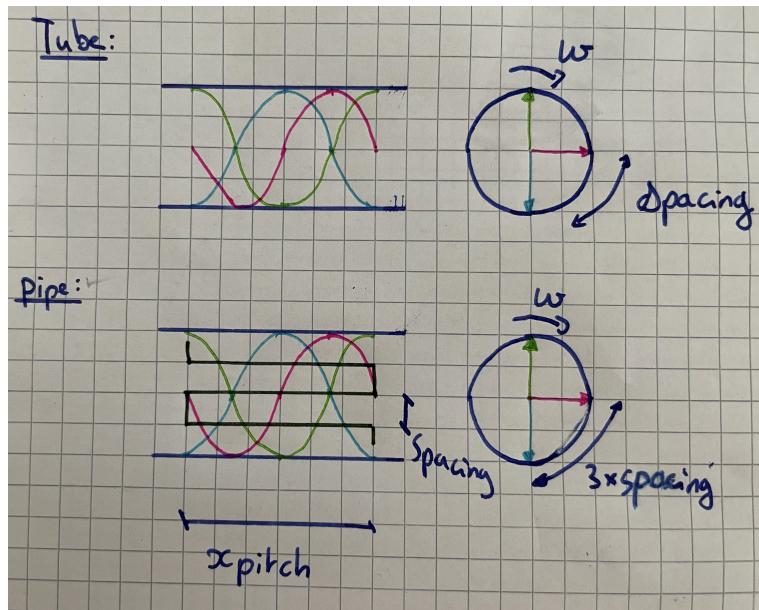


Figure 4: Caption

adhesion of the fibre is complete then one can increase to higher speeds such as 900 mm/s. The upper limit depends greatly on the polymer and the extrusion rate used at the time of printing. The spacing between the fibres is achieved by minimising the jet-lag with the

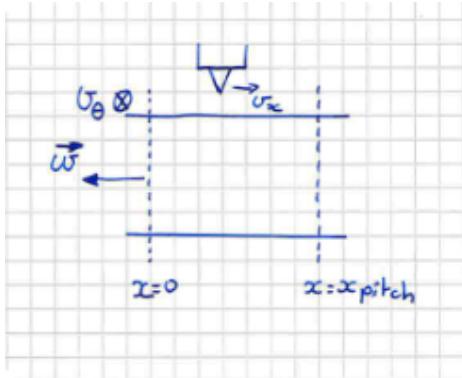


Figure 5: Diagram of the tool-head and rotational speeds

introduction of pauses (using the g-code G4 P500, the tool waits for a period of 500 [ms]). The spacing is based on the value inputted by the user. However due to geometrical incompatibilities, the spacing is corrected such that a layer finishes where it started. The new spacing is computed using the following expression: number of fibres =  $\lfloor \frac{2 \times \pi \times R_{mandrel}}{\text{spacing}} \rfloor$  and then the new spacing is computed with: spacing =  $\frac{2 \times \pi \times R_{mandrel}}{\text{number of fibres}}$ .

### 3.4 G-code

A FDM printer is a tool that computes the moves required with a kinematics formula to convert the coordinates of the workspace to motor positions. Generally the g-code is generated by a slicer that takes a CAD model and converts it to positions and a tool path that join those positions. G-code describes how the tool moves and controls the peripherals for turning a heater on or off to detecting that the tool head is out of bounds, etc.

The tubular collector doesn't correspond to any predefined kinematics (in klipper), so in our application the extra motor is defined as a manual stepper in klipper [12]. The command to move the motor is the following : MANUAL\_STEPPER=stepper\_name MOVE=distance SPEED=speed ACCEL=acceleration SYNC=0. There is a dependency of these values for the stepper motor in the config file.

For each stepper motor, klipper requires the rotation distance. This value is a linearised distance that the motor completes in one rotation. For our system, this value is equal to 9. Through testing and changing the config file, higher values for the rotation distance reduces the rotation speed and lower values increases the speed without changing the klipper command. It seems that the software tries to maintain the tangential speed constant. By changing the rotation distance, we are telling the board that the distance is covered in a rotation is larger than what it is. The speed is governed by the maximum speed allowed in the config file. This is the only dependence between the two that was noticed whilst testing. The acceleration value tells the motors how much acceleration it must give to the shaft to acquire the given speed. The most useful feature of this command is the SYNC option. When SYNC equals to 0, the printer will execute the next g-code command in the sequence. This feature is particularly useful for the tube pattern, it lightens the g-code file because the path can be defined in term of revolutions than in term of sequences of steps.

```
; sample g-code for tube

G1 X10 y40 Z2
MANUAL_STEPPER STEPPER=stepper_1 MOVE=6.75 SPEED=0.5 ACCEL=100 SYNC=0
; turns the shaft for 1 revolution
G1 X15 Y40 Z2
; makes the print head move by 5 mm toward x positive
```

Figure 6: This is a g-code example of the SYNC = 0. This will produce a spiral of length 5. If SYNC=1 then the pattern produced would be different. It would be a loop followed with a 5 mm long fibre.

## 4 Materials

The first material used on the printer was polycaprolactone (PCL), that is already established in the MEW field. It is biodegradable which makes it suitable for biomedical uses. The G-code generating software was tested on PCL. Once the patterns were up to standards, the software was validated for this particular pattern.

One of the objectives of the internship was to use a new polymer. Polyhydroxyalkanoates (PHA) was chosen because of it's fully biodegradable and bio derived properties as they are very interesting for tissue engineering, one of the main area of application of MEW. As

Parameter name	Value for PHA	Value for PCL
Voltage applied [kV]	2.68	1.8
Z height [mm]	2.5	2
$v_x$ speed [mm/s]	800-900	15
extrusion rate [ $\frac{\mu\text{m}}{\text{min}}$ ]	5000	1000
melting temperature [°C]	185	85
fibre diameter [ $\mu\text{m}$ ]	30 - 40	5 - 15[13]

Table 1: MEW parameters and their values for PCL and PHA.

we can see on the 1 table, the processing conditions are similar between the two polymers. The values presented enabled us to produce the samples shown in the Results section. Due to the high process temperature for PHA, it is normal that the speeds are increased to deposit the material in a similar fashion. During testing, when the material was not extruded fast enough it would solidify before touching the shaft and therefore not adhere to it.

## 5 Results

This section will discuss the PHA samples as the PCL samples would be similar in structure. The samples were produced using g-codes produced by the developed software. Once generated they were uploaded to the printer and printed multiple times to present repeatability within the print session. The tested g-codes were never deleted and were tested across multiple print sessions to show repeatability across multiple sessions. A session is defined as the time that the printer was powered on.

The samples shown in figure 7, were produced with the mew parameters described in 1. As we can see the stacking behaviour is good. As we print on the centre of the shaft, the fibre will always feel a vertical electrical field locally. Thus, using the same tool path will result in the same deposition with limited variations from the electrical field.

As shown on figure 8, the fibres stack reasonably well, however there are some variations that can be seen on the sub-figure 8c. These images were used to measure the

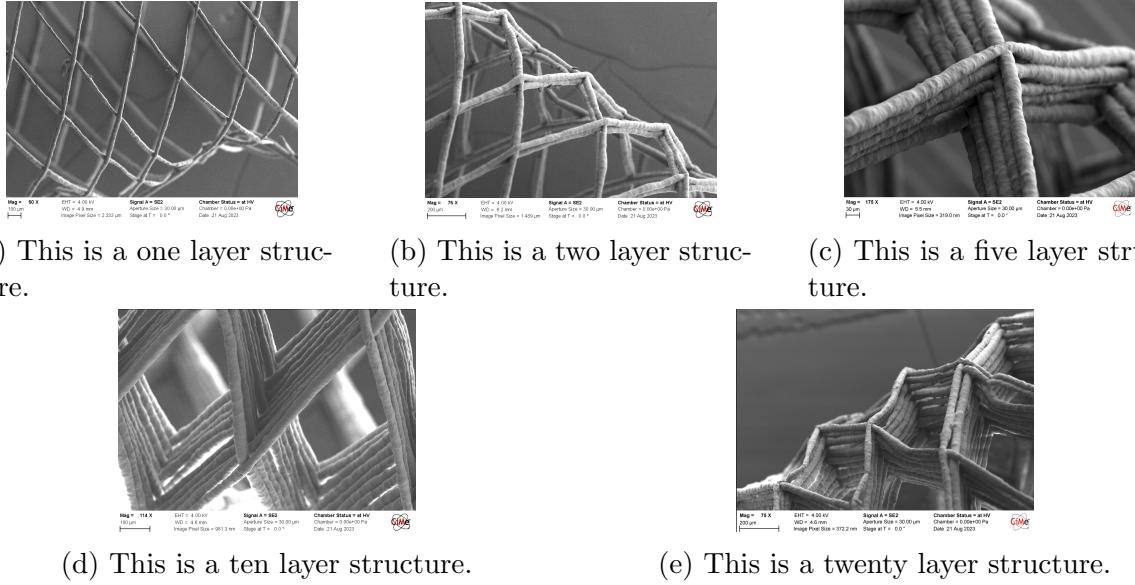


Figure 7: These figures show the tube structures imaged using the Gemini SEM.

fibre diameter using the GIFT method [14] [15]. The measurements are summarised in the table 2 The GIFT method is an automatic method that can process large batches of

Number of layers	Fibre diameter [ $\mu\text{m}$ ]	standard deviation
1	34.2	5.4
2	36.6	4.3
5	26.3	10.4
10	27.1	6.1
20	39.4	7.5

Table 2: This table summarises the fibre diameters that were measured using the GIFT method. These results are just an indication for these samples, they may not be representative of the polymer.

images to produce fibre measurements. It detects the edges of the fibres, then computes the frequency of the distances. Once the frequency histogram is generated, the software will apply a Gaussian fit to the histogram and will return the average value as well as the standard deviation.

## 6 Hardware limit

The pattern size is limited by the shaft's diameter. The main issue in reducing the shaft diameter is the bending of the shaft. Should the shaft bend, the pattern will take

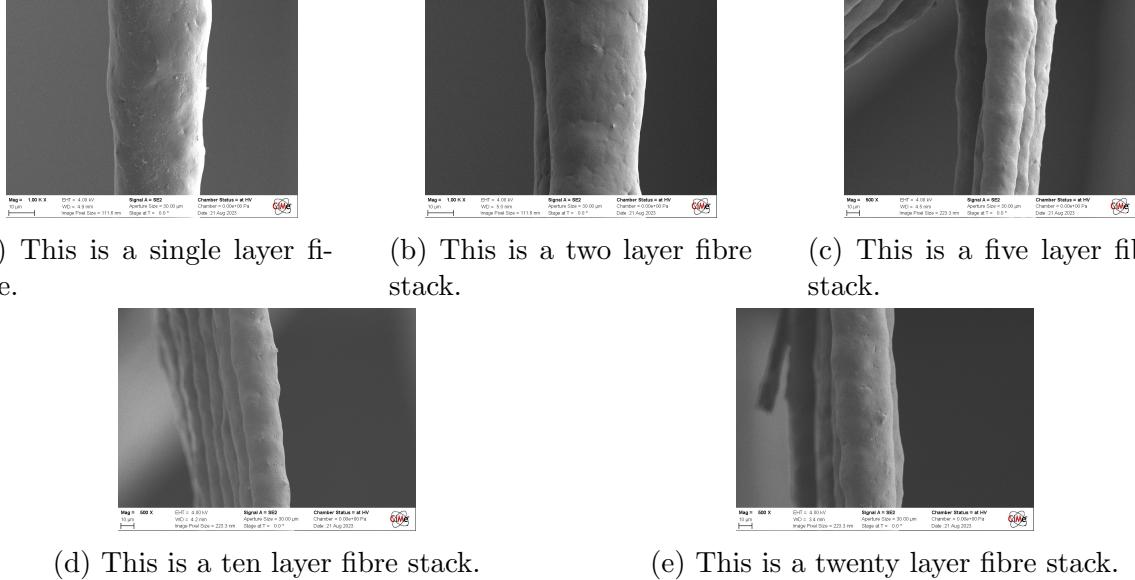


Figure 8: These figures show the tube structures imaged using the Gemini SEM.

the curvature of the shaft, thus altering the spacing and the fibre properties within the structure. To compute the smallest radius that can be applied to the shaft, the principles of beam theory.

The deflection of a beam is linked to the forces applied to it by differential relations. Our beam is subjected to gravity and the electrical force applied by the high voltage. To model the electrical force, we approximate the system to a parallel plate capacitor with a surface equal to the nozzle. By applying the Gauss theorem to one capacitor plate, we obtain the charge on the plate:

$$\oint_S E(s) ds = \frac{Q}{\epsilon_0} \quad (2)$$

For a capacitor the relation between the capacitance, the charge and the voltage is given by:

$$Q = C \times V \quad (3)$$

By inserting 3 in 2, we can obtain a relation between the surface of the plate, the capacitance and the voltage. The capacitance of a plate capacitor is well known and is equal to plate surface  $S$  divided by the distance between the plates:  $C = \frac{\epsilon_0 S}{d}$ .

$$\oint_S E(s) ds = \frac{Q}{\epsilon_0} = \frac{CV}{\epsilon_0} = \frac{SV}{d} \quad (4)$$

Under the hypothesis of a plate capacitor, the edge effects are ignored and thus the electrical field is uniform between the plates. The integral thus becomes:

$$\oint_S E(s) ds = E \times S \quad (5)$$

This results in a simple formula to compute the electrical field:

$$E = \frac{V}{d} \quad (6)$$

The force require the charge on the other plate and the electrical field applied between the plates. As the capacitor has a neutral charge, the charge found on the first plate is equal in absolute value to the charge on the first plate.

$$F = q \times E = \frac{\epsilon_0 S V}{d} \times \frac{V}{d} = \epsilon_0 S \left(\frac{V}{d}\right)^2 \quad (7)$$

By using the correct values for our system:  $V = 3$  [kV],  $d = 2$  [mm],  $S = \pi R^2$  and  $R = 0.5$  [mm]. We obtain a force of about a magnitude of  $10^{-5}$ . When compared to gravity (about  $10^{-1}$ ), the electrical force only has a limited effect and can be ignored.

Now that the forces have been identified, we can start computing the deflection. We use the force diagram 9 to determine the boundary conditions when solving the differential equations. The reaction forces are still unknown to us. To compute , we apply the 2<sup>nd</sup>

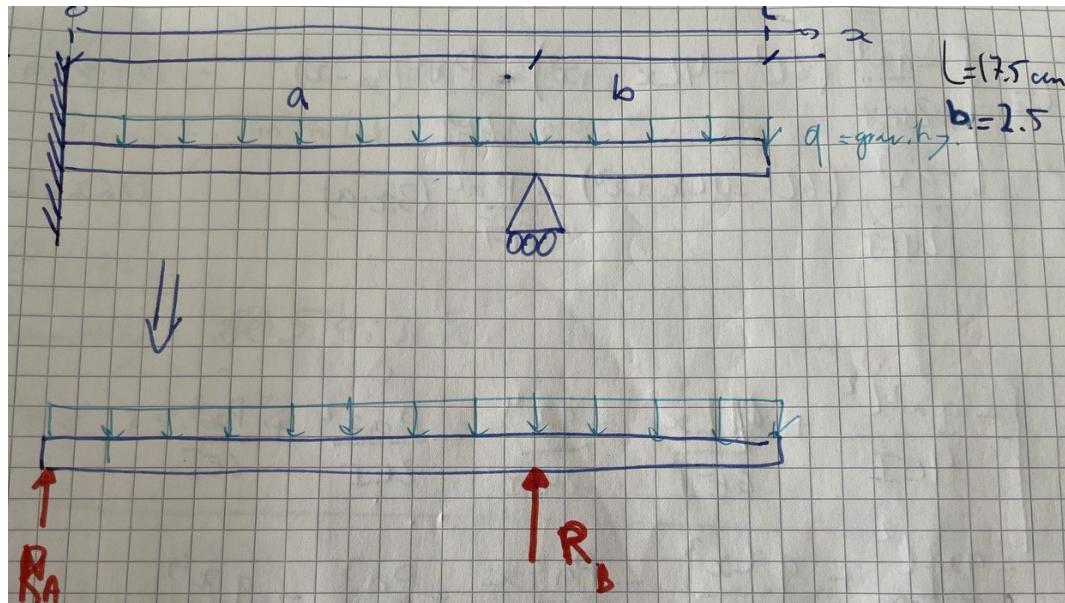


Figure 9: This is the force diagram being applied to the shaft. The Red forces are punctual forces, the blue force is the distributed force of gravity applied to the rod.

law of Newton and the sum of the moments at  $x=0$ :

$$\sum \vec{F} = \vec{0} \Rightarrow R_a + R_b = \int_0^L q(x) dx = m_{beam} g \quad (8)$$

$$\sum \vec{M}_a = \vec{0} \Rightarrow R_b \times a = \int_0^L q(x) x dx = \frac{m_{beam} g L}{2} \quad (9)$$

Thus we can therefore express the forces applied as the superposition of two simpler systems: the distributed forces and the punctual forces. We know how to compute their respective deflection using the differential relation:

$$\left\{ \begin{array}{l} \frac{d^4}{dx} w(x) = -\frac{q(x)}{EI} \\ \frac{d^3}{dx} w(x) = \frac{V(x)}{EI} \\ \frac{d^2}{dx} w(x) = -\frac{M(x)}{EI} \\ \frac{d}{dx} w(x) = \int_0^L w'(x) dx \\ w(x) = \int_0^L w'(x) dx \end{array} \right.$$

The total deflection is the sum of the deflections of the two subsystems. In order to find the maximum deflection, we compute the first derivative and find the values in which the derivative is zero. The deflection is computed for those positions, the largest value is defined as the largest value. A reasonable maximum deflection for the MEW process is 100 microns. The position  $x=0$  is defined as the position in which the shaft is held in place. So  $x=175$  [mm] is the free end of the shaft.

Shaft radius [mm]	Deflection [ $\mu m$ ]	Location of the maxima [mm]
1.5	18.1	175
1	40.1	175
0.65	96.6	175
0.5	163.2	175

Table 3: this table shows the maximum deflection for shaft of a given radius.

## 7 Conclusion

This work has shown that a tubular collector is possible to use as a research tool. Once the printer was operational, it was used to produce samples using a new material to expand the catalogue of the possible materials for MEW. The fibres are between 30 and 40 microns wide, which is far from the performance of PCL (5-15 microns) [13].

The next steps would be to lower the fibre diameter as well as optimising one of the patterns for a given application. This could be tissue engineering, photonics or microfluidics. Another possibility, would be to explore the limits of the tool such as the minimum feature size achievable or printing geometries (printing on 2D curved surfaces). The possibilities offered by new materials are extremely large and could result in new understanding of the MEW technology.

## References

- [1] Ander Reizabal et al. “MEWron: An open-source melt electrowriting platform”. In: *Additive Manufacturing* 71 (2023), p. 103604. ISSN: 2214-8604. DOI: <https://doi.org/10.1016/j.addma.2023.103604>. URL: <https://www.sciencedirect.com/science/article/pii/S2214860423002178>.
- [2] *Voron design - Voron V0.1*. URL: <https://vorondesign.com/voron0.1> (visited on 08/23/2023).
- [3] *Voron design - software installation*. URL: <https://docs.vorondesign.com/build/software/> (visited on 08/23/2023).
- [4] *STM32CubeProgrammer - STMicroelectronics*. URL: <https://www.st.com/en/development-tools/stm32cubeprog.html> (visited on 08/28/2023).
- [5] *Fysetc spider bootloader*. URL: <https://github.com/FYSETC/FYSETC-SPIDER/tree/main/bootloader> (visited on 08/28/2023).
- [6] *WinSCP software*. URL: <https://winscp.net/eng/download.php> (visited on 08/28/2023).
- [7] *GNU Affero General Public license*. URL: <https://www.gnu.org/licenses/agpl-3.0.de.html> (visited on 08/29/2023).
- [8] *OctoPrint-Enclosure plugin*. URL: <https://plugins.octoprint.org/plugins/enclosure/> (visited on 08/29/2023).
- [9] *Simple Emergency Stop plugin*. URL: <https://plugins.octoprint.org/plugins/simpleemergencystop/> (visited on 08/29/2023).
- [10] *M112 G-code on RepRap*. URL: [https://reprap.org/wiki/G-code#M112:\\_Full\\_.28Emergency.29\\_Stop](https://reprap.org/wiki/G-code#M112:_Full_.28Emergency.29_Stop) (visited on 08/29/2023).
- [11] Alessandro Zedgitt. “Melt Electrowriting (MEW) G-code generator for tubular collector”. In: (2023). URL: [https://github.com/azedgitt/MEW\\_G-code\\_generator/tree/main#melt-electrowriting-mew-g-code-generator-for-tubular-collector](https://github.com/azedgitt/MEW_G-code_generator/tree/main#melt-electrowriting-mew-g-code-generator-for-tubular-collector).
- [12] *Manual Stepper Klipper Documentation*. URL: [https://www.klipper3d.org/G-Codes.html#manual\\_stepper](https://www.klipper3d.org/G-Codes.html#manual_stepper) (visited on 08/14/2023).
- [13] Taveet Kangur. “Developing Melt Electrowriting on an Open-Source Fused Filament Fabrication Platform”. In: (2023).
- [14] Andreas Götz et al. “General image fiber tool: A concept for automated evaluation of fiber diameters in SEM images”. In: *Measurement* 177 (2021), p. 109265. ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2021.109265>. URL: <https://www.sciencedirect.com/science/article/pii/S0263224121002736>.

- [15] Jennifer Huling et al. “GIFT: An ImageJ macro for automated fiber diameter quantification”. In: *PLOS ONE* 17.10 (Oct. 2022), pp. 1–13. DOI: 10.1371/journal.pone.0275528. URL: <https://doi.org/10.1371/journal.pone.0275528>.