POLITECNICO DI TORINO

Corso di Laura Magistrale
in Mechatronic Engineering

Tesi di Laurea Magistrale

Artifacts in FDM 3D Printing, Extrusion Inconsistency Problem and Weight Reduction

Relatore
prof. Massimo Violante

.....

Candidato
Francesco Benasso

.....

Anno Accademico 2017-2018

ABSTRACT

The purpose of this thesis is to and explain and to summarize, in a logical manner, all the experiments, the procedures and results obtained during a 5 months' internship in PrusaResearch. Extrusion inconsistency artifact was investigated by means of the creation of a testing machine. The purpose of the testing machine is to collect data about the force and the speed supplied by the feeding mechanism in order to have a better understanding of the causes of the phenomenon. An attempt to control the pressure inside the nozzle is made by implementing a closed loop control. A geared extruder is investigated in order to have a better extrusion and to reduce weight. In conclusion, some proposals of sensor implementations are given to close the loop.

The thesis is composed by the following chapters:

- **Chapter 1**: Introduction and a bit of history of 3D printing and additive manufacturing field, followed by a quick description of the hosting company;
- **Chapter 2**: 3D printing workflow is explained. All the necessary step to print an object are explained, from the idea to the finished object. CAD software, STL files, Slicer software, G code, 3D printer and post-processing are explained;
- **Chapter 3**: Limitations of FDM technology and Accuracy in positioning of the TCP is discussed;
- **Chapter 4**: Artifacts are introduced, and few examples are presented;
- **Chapter 5**: Extrusion Inconsistency is explained;
- **Chapter 6**: Description of the testing machine is given in terms of mechanics, electronics and software;
- **Chapter 7**: Explanation of the tests performed, and the results obtained in the investigation on the effects of the slicing software related to Extrusion Inconsistency;
- **Chapter 8**: Explanation of the tests performed, and the results obtained in the investigation on the effects of Marlin Firmware related to Extrusion Inconsistency;
- **Chapter 9**: Explanation of the tests performed, and the results obtained in the investigation on the effects of the Mechanics and actuators related to Extrusion Inconsistency;
- **Chapter 10**: Design of the Extrusion System Model in Simulink;
- **Chapter 11**: Implementation of PID controller and tuning by model-based approach and empirical approach;
- **Chapter 12**: Proposal of sensors to provide the feedback;
- **Chapter 13**: New dimensioning for the stepper motor is investigated with the aim of having a better extrusion and to reduce weight.
- **Chapter 14**: Conclusion and further advancement.

CHAPTER 1: Introduction to 3D printing

In this chapter an introduction to 3D printing world is given and the main techniques are introduced. A bit of history of 3D printing follows from the first patents, to the birth of the open source community around RepRap Project until present days. Finally, a brief presentation of PrusaResearch is reported.

DEFINITION OF 3D PRINTING AND MAIN TECNIQUES

According to Cambridge Dictionary, 3D printing is "the process of printing a solid object from a digital model by printing many separate layers of the object". This can be achieved with a 3D printer, which is a device that can put together material, like plastic polymers, resins or metals, in layer fashion for creating an object from scratch. This technique differs from traditional manufacturing methods, which are based on the removal of material, for its additive nature.

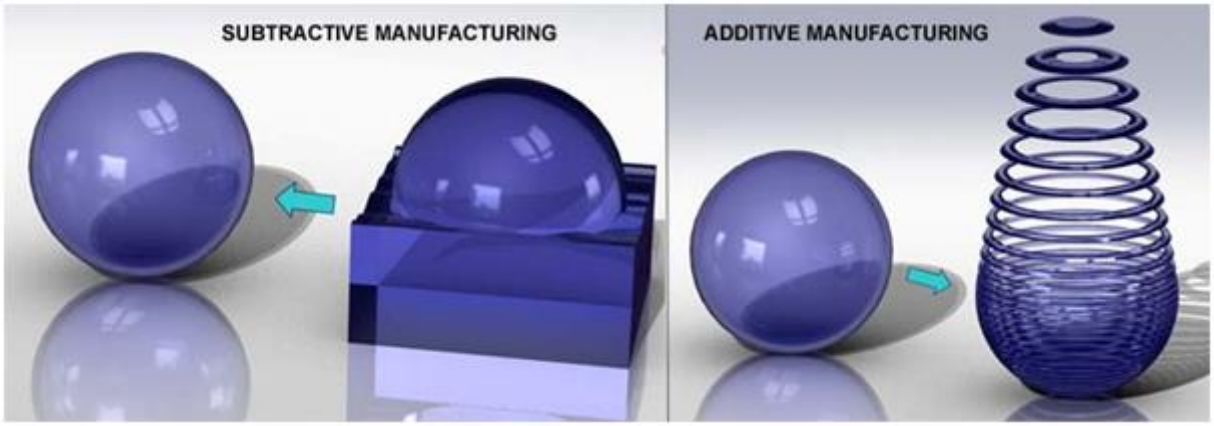


Figure 1: Subtracting Manufacturing and Additive Manufacturing

Several technologies are available to 3D-print objects and the main difference between them is the way material is aggregated.

- **FDM** (Fused Deposition Modeling) technique is based on a nozzle, moving in space and depositing fused polymer, layer by layer, in order to print the desired geometry. With this technology, the most printed materials are plastic polymers such as: PLA, ABS, PET, Nylon, Polycarbonate, etc.

- **SLS** (Selective Laser Sintering) technology can print both metals and polymers which are in a powder state. Every time a new layer is printed, the whole print surface is covered with a thin layer of powder, which is then melted with a precise and powerful source of energy, such as a laser or an electron beam.

- **SLA** or **STL** (STereoLithography Apparatus) technology is based on the curing of liquid material. Photopolymer resins are mainly printed with this technology and cured by UV light exposure.

HISTORY OF 3D PRINTING

3D printing technologies are not a brand-new idea: the first scientific articles appear in the late ’70s, while the first patents are published in the ’80s.

In 1986 Charles W. Hull patents the first SLA machine (US Patent 4575330) [1], starting the company, 3D System. He is also the father of STL file format. Quoting the patent:

“ In recent years, very sophisticated techniques have been developed for generating three-dimensional objects within a fluid medium which is selectively cured by beams of radiation brought to selective focus at prescribed intersection points within the three-dimensional volume of the fluid medium. “

In 1989 S. Scott Crump patents the first FDM machine ( ) [2], founding the market leading company Stratasys Inc.

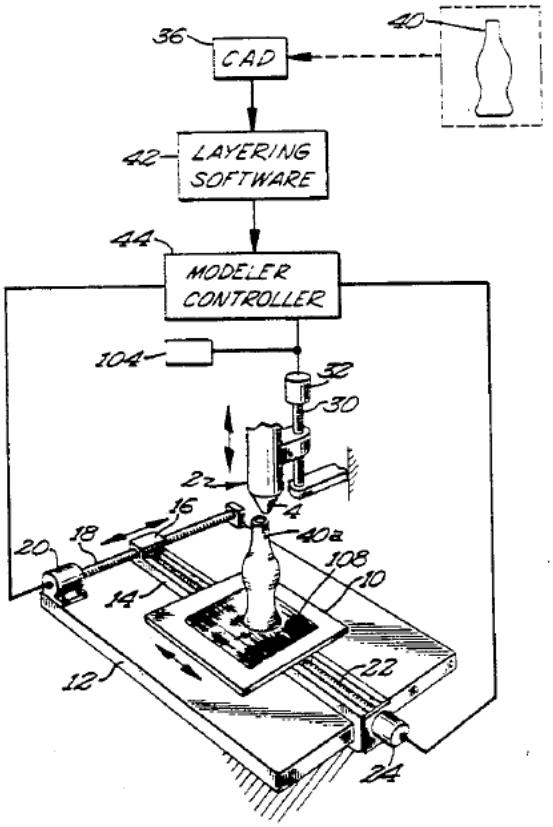


Figure 2: Image from patent US 5121329A

During the following 20 years of the patents‘ duration, 3D printing is mostly confined to prototyping field and ultra-customized, small volume production. The main characters on the market are professional printers made with proprietary technology.

The cost for each machine is very high, making it affordable only for structured corporations and limited to their offices. 3D printing in barely known out of this context and major improvement are slow to come, due to the lack of competitors.

It is only in the first years of the 21th century, that the need for a more affordable machine, suitable for scientific and household environment and budget, is satisfied.

In 2005, in England, at the University of Bath, Dr Adrian Bowyer, senior lecturer in mechanical engineering, founds the RepRap Project. It is an initiative to develop a low-cost and open-source 3D printer capable of replicating itself. In other terms the design of the printer is developed with the aim of maximizing the number of parts which can be directly printed by another RepRap 3D printer. In addition, the remaining components are required to be inexpensive and easily findable in retail hardware store. At the beginning, electronics (PCBs) as well, are thought to be 3D printed, however, nearly at the same time, Arduino environment is catching on and soon after, it is widely adopted by the project. In Arduino, the RepRap project can find a cheap and reliable source of open-source electronics. The developed printers are based on FDM technology, which is, among all the other methods, the cheaper and the more compatible with a non-industrial environment.

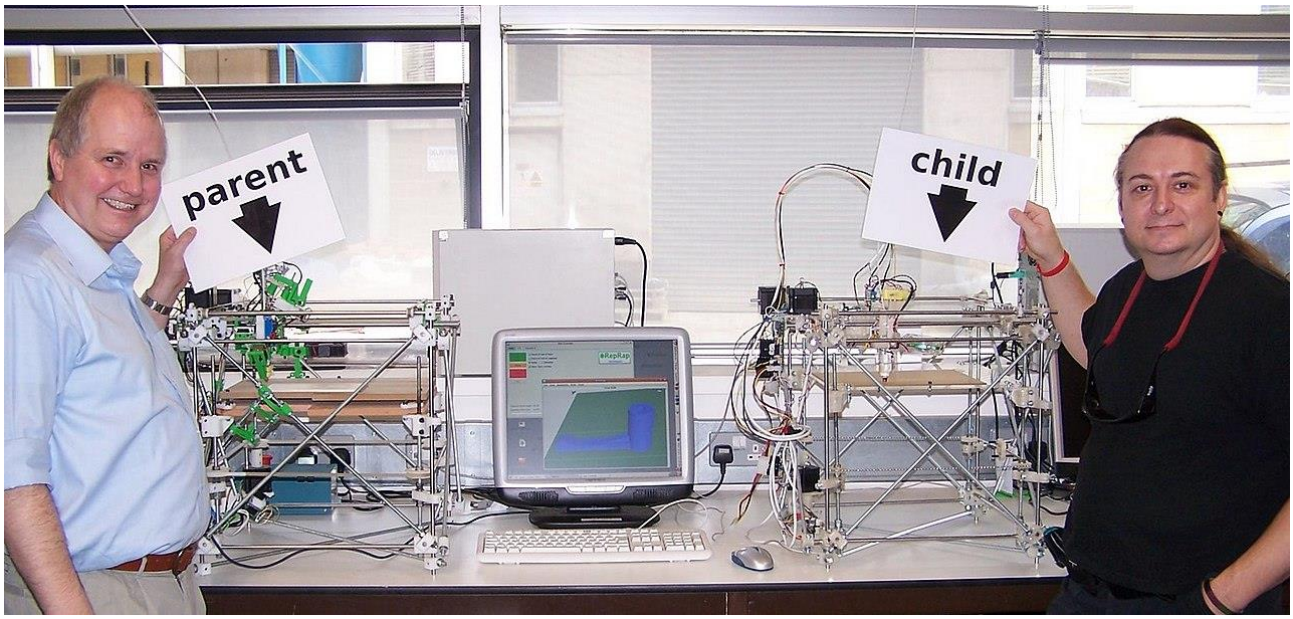


Figure 3: All the plastic parts for the machine on the right are produced by the machine on the left. Adrian Bowyer is on the left. From RepRap website

Objects are printed with plastic filament, stored and shipped in reels. No special precaution must be observed for storing and handling the filament, since these thermo-plastics are not a safety hazard. Only few polymers (such as ABS, PLA or PVA) are hygroscopic, which means that they should be stored in dry environment, otherwise moisture absorption can take place, leading to a performance decrease, but no safety issues. No harmful fumes are produced during the printing process, letting the printer to work at home or in office, without the need of any air filtering system. No electrical upgrades must be done, since low power consumption is a winning feature of the printing process: the most energy-consuming components are usually the heaters which in the worst-case scenario are comparable in power consumption with a refrigerator. Furthermore, low complexity of the process makes it compatible with Arduino 8 bits architecture.

From 2005 to 2009 RepRap project is a collection of best practice for building a 3D printer at home. No commercial activities grow from it: the US patent 5121329A is still valid and granted.

However, in 2009 the patent expires, and the first commercial printer are available for the hobbyist's market, sold in DIY kit by MakerBot, which after few years were sold to Stratasys.

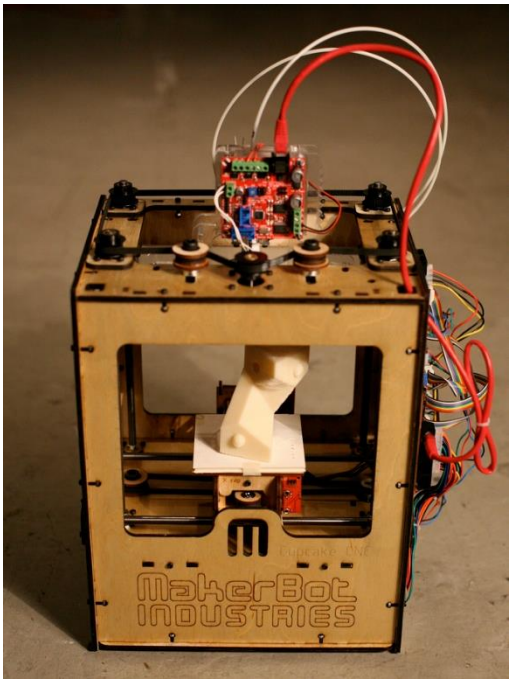


Figure 4: The first 3D printer for DIY market, MakerBot Cupcake CNC.

After 2009, hundreds of companies flourish, offering the market new 3D printers, at always more and more competitive costs. This increase of demand and availability for consumer's printers leads to the formation of active communities of users, who, empowered by the open-source hardware and software environment can work, mod and develop the technology.

Wohlers Associates is recognized as one of the preeminent 3D printer experts in the world, writing from 1993 to today an annual report on 3D printing and additive manufacturing state of the industry. According to Wohlers Report 2016: "More than 278,000 desktop 3D printers (under $5,000) were sold worldwide last year (2015), according to Wohlers Associates". While on Wohlers Report 2018, it is claimed that 528,952 desktop 3D printers were sold worldwide in 2017.

THE HOSTING COMPANY: PrusaResearch

PrusaResearch, founded in 2012 and based in Prague (Czech Republic), designs and produces 3D printers for the consumer's market, based on FDM technology.

PrusaResearch is fully compliant with RepRap Project guide lines, which means that printers' design is Open Source and that they are self-replicating.

Being an Open Source company means that every piece of information needed to build and reproduce the printers is shared online. In fact, it is possible to find all the technical information about the mechanics, the electronics and the firmware on Prusa's Github page.

QR Code 1: Link to Prusa's Github Page

Self-replicating means that the printers are designed to be capable of reproducing them self. This capability is limited to the plastic parts of the printer. This feature leads to the unique way the company chooses to manufacture all the plastic components: the farm.

The farm is composed by several 3D printers (360 on July 2018), disposed in multi-level racks, in the same room. The farm works 24/7, in order to fulfill the production level of around 6000 printers/month (July 2018). On average 27 hours are required to print all the components needed for one printer.

QR Code 2: Link to YouTube video on the Farm

CHAPTER 2: 3D Printing Workflow

In this chapter the necessary steps to print an object with a 3D printer are discussed. Particular attention is given to FDM printers and open source solutions.

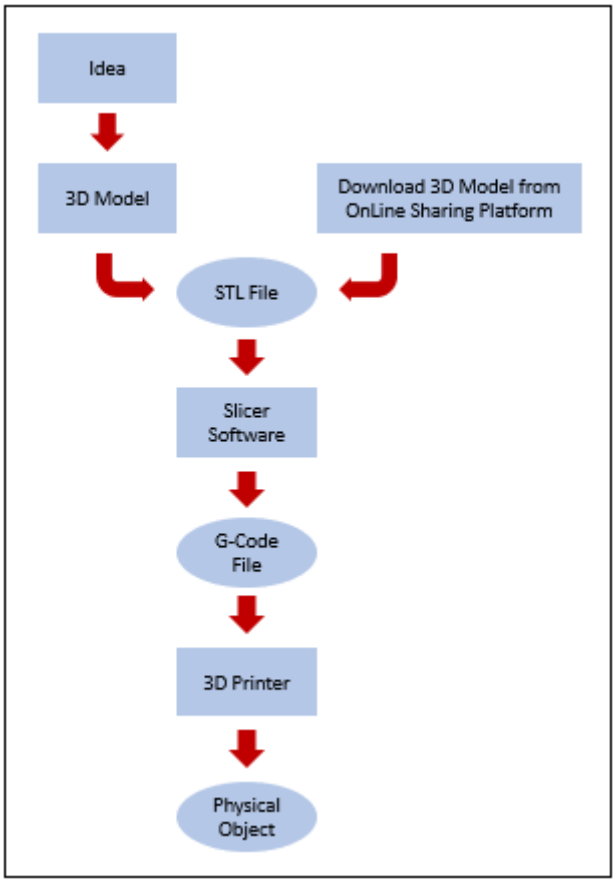


Figure 5: 3D printing workflow

THE 3D MODEL

The starting point of the whole process is the digital model of the object to be printed. The model of the object can be realized in CAD environment or downloaded from online sharing platform.

CAD (Computer-Aided Design) is a software tool which allows the creation and modification of 3D models, analysis and optimization can also be performed. Several tools can be found on the market, open source (OpenScad, FreeCad, Google Sketckh Up, TinkerCad, Blender) or proprietary (AutoDesk, Rhinoceros, Catia, SolidWorks).

3D models can be also downloaded from internet, from sharing platform filled by user's projects and designs. This is a handy shortcut which opens 3D printing to amateur users. Popular web sites are: Thingiverse and GrabCad.

STL FILE FORMAT

When the object to be printed is described in a 3D model within a CAD software, it can be exported or converted in STL file format. In this format, objects are described only by their surface geometry, without any other additional data such as color or texture. Surfaces' geometry is described as a tessellation of triangles in a Cartesian coordinate system. Each triangle is represented with the coordinates of its 3 vertices in the Cartesian space and the normal unit vector to the surface, so that there is a total of 12 numbers stored for each triangle. Newer versions of STL format include color information for multi-material printing.

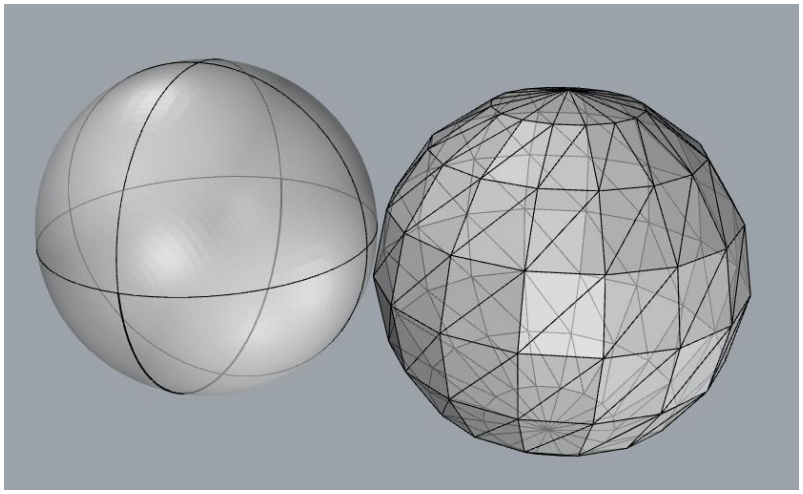


Figure 6: On the left, a sphere's 3D model in CAD, on the right, a low resolution STL file of the same sphere

There are two style of STL format: ASCII STL or Binary STL. The ASCII STL format is mainly intended for testing, since the large size of its files makes it impractical for general use. Information about each triangle is represented as follows:

```
Facet normal n_x n_y n_z
     outer loop
          vertex v_1x v_1y v_1z
          vertex v_2x v_2y v_2z
          vertex v_3x v_3y v_3z
     endloop
endfacet
```

Increasing the resolution, STL files can grow their size dramatically, so a more compact version is represented by Binary STL. At the beginning of Binary STL file there is a 80-bytes header that can be interpreted as a comment and is usually ignored by the vast majority of STL file reader.

The following 4-bytes long integer gives the total number of facets in the file. What follows is the description of each triangle, one after the other. As in ASCII STL, triangles are represented by the normal and the coordinates of each vertex. Coordinates and components of the normal are represented as a 4-bytes floating point number. There is also a 2-byte spacer at the end of each facet data. The result is that each triangle is represented by 50 bytes: 12 for the normal, 36 for the 3 vertices, and 2 for the spacer.

THE SLICER

The slicer software receives as input the STL file of the object to be printed and gives as output the G-code to feed the printer with. It performs the computation to cut the object in layers and then for each layer, computes the tool path to be executed by the printer.

The number of layers in an object and the related layer height define the resolution in the Z axis of the printer. The thinner is the layer height, the higher will be the number of layers needed to print an object, the better will be the quality. However, increasing the number of layers leads to an increase in the printing time. The need of finding a good compromise between number of layers, printing quality and printing time, is shared by nearly every additive manufacturing technology. In FDM 3D-printing this compromise is generally achieved with a layer height of 0.2 mm, even though thinner layers are reachable by the technology.

In order to output the G-code, the slicer has to implement a multiple-steps procedure which cannot be completed without knowing few parameters of the printer executing the instructions.

As first thing the STL file is loaded in the slicer, and the model is inserted in a cartesian reference frame. The positioning of the 3D model is then chosen by the user in order to maximize the contact surface with the printing bed, which lays in the XY plane. This is done to enhance the adhesion of the object to the printing bed. This is the starting point to define the cutting planes which are parallel to the printing bed and therefore perpendicular to the Z axis. Afterwards, the intersection between every cutting plane and the model is computed, with the following general algorithm:

```
For each(triangle in the STL file):
     For each(cutting plane):
          Compute intersection between cutting plane triangle;
          If(the result = 2 points) then:
               add the line connecting the 2 points to the list of
               line segments for current cutting plane;

For each(cutting plane):
     Assemble the list of line segment as a set of continuous line
```

The continuous lines obtained from the previous step are the starting point for defining the structure of each layer. Each layer is composed by one or more perimeters, corresponding to the previously mentioned continuous line, and by the infill, which can be at full or partial density. Those components are then translated in the path the printer's tool must follow. Finally, the paths are translated into G-code, which can be interpreted by the printer.

THE G-CODE

The G-code also known as RS-274, is the most widely used numerical control programming language, used in CAM (computed-aided manufacturing) to control automated machine tool. It firstly appeared in the '50s, and it has been developed and enhanced till the present days. Many dialects exist, since each producer has customized it, but it is also possible to refer to the international standard ISO 6983.

Open-source 3D printing birth occurs in a period in which G-code language is stable and wide-spread, so the communities embraced it. In this field, there is no recognized standard, albeit there is no significant difference with respect to the CNC G-code style. Hence different flavors and implementations are possible, which are dependent both to the executing printer and to the firmware.

G-code, which stands for "geometric code", instructs the machine or the printer where and how to move the tool or the extruder. The code is read and executed one line after the other, from the top of the file to the bottom. It usually follows some variation of the following alpha numeric pattern:

N## G## X## Y## Z## F## S## T## M##

In CNC Application:

N: Line number
G: Motion
X: Horizontal position
Y: Vertical position
Z: Depth
F: Feed rate
S: Spindle speed
T: Tool selection
M: Miscellaneous functions

In 3D-Printing Application:

G: Motion
X: Horizontal position
Y: Vertical position
Z: Depth
E: Extruder Position
F: Feed rate
M: Miscellaneous functions

It is possible to notice how open-source 3D printing is using a sub-set of traditional G-code commands. A collection of all the commands and the implementing firmware can be found on RepRap Project website.

QR Code 3: Link to RepRap website about G-Code Commands

The slicer, to generate a G-code file for a 3D printer, usually needs few data about the printer in which it will be executed, for better performance, fast motion and filament managment. At the minimum, the location of the Cartesian reference frame must be communicated to the slicer. A G-code file is usually is composed by 3 parts: the start, the printing of the object and the conclusion.

In the first part, some commands are used to initialize the printer firmware, switching on the heaters, homing all the axis, sensing the bed for calibration and priming the extruder. These commands can vary from one printer to the other and are generally dependent on the printer. For example, Prusa's Slicer, customization of the open-source Slic3er, generates the starting commands in Figure 7.

The second part is the main part of the file. All the motions for printing the object are store in this portion of the file. Here commands are packed together layer by layer. An example of a piece of code can be found in figure 8.

In the conclusive part, few commands are used mainly to switch off the heaters and fans, to disable the motors and to park the extruder in homing position to allow a safer collection of the printed object by the user. An example of this portion of code can be found in figure 9.

```
M201 X9000 Y9000 Z500 E10000 ; sets maximum accelerations, mm/sec^2
M203 X500 Y500 Z12 E120 ; sets maximum feedrates, mm/sec
M204 P1500 R1500 T1500 ; sets acceleration (P, T) and retract acceleration (R), mm/sec^2
M205 X10.00 Y10.00 Z0.20 E2.50 ; sets the jerk limits, mm/sec
M205 S0 T0 ; sets the minimum extruding and travel feed rate, mm/sec
M107 ;Fans Off
M115 U3.3.1 ; tell printer latest fw version
M201 X1000 Y1000 Z1000 E9000 ; sets maximum accelerations, mm/sec^2
M203 X200 Y200 Z12 E120 ; sets maximum feedrates, mm/sec
M204 S1250 T1250 ; sets acceleration (S) and retract acceleration (T)
M205 X8 Y8 Z0.4 E1.5 ; sets the jerk limits, mm/sec
M205 S0 T0 ; sets the minimum extruding and travel feed rate, mm/sec
M83  ; extruder relative mode
M104 S215 ; set extruder temp
M140 S60 ; set bed temp
M190 S60 ; wait for bed temp
M109 S215 ; wait for extruder temp
G1 E-3 ; Retraction for avoiding filament stringing  by_fraBenasso
M73 Q0 S123
M73 P0 R123 ; Set build percentage
G28 W ; home all without mesh bed level
G80 ; mesh bed leveling
G1 Y-3.0 F1000.0 ; go outside print area
G1 E 3 ; ReExtruding Filament back to Initial State by_fraBenasso
G92 E0.0
G1 X60.0 E9.0  F1000.0 ; intro line
G1 X100.0 E12.5  F1000.0 ; intro line
G92 E0.0
M221 S95
M900 K30; Filament gcode
G21 ; set units to millimeters
G90 ; use absolute coordinates
M83 ; use relative distances for extrusion
;BEFORE_LAYER_CHANGE
G92 E0.0
;0.2
```

Figure 7: Starting G-code commands generated by Slic3r Prusa Edition

```
;AFTER_LAYER_CHANGE
;0.35
G1 X100.226 Y80.271 F7200.000
G1 Z0.350 F7200.000
G1 E0.80000 F2100.00000
M204 S800
G1 F900
G1 X100.226 Y80.226 E0.00117
G1 X149.774 Y80.226 E1.29101
G1 X149.774 Y129.774 E1.29101
G1 X100.226 Y129.774 E1.29101
G1 X100.226 Y80.331 E1.28828
M204 S1000
;BEFORE_LAYER_CHANGE
;0.5
```

The nozzle is positioned close to the point in which the extrusion must begin, then it is lowered to the correct layer Z-height, and then the filament is pushed in the nozzle of 0.8 mm to start to fill it.

Acceleration and velocity are decreased

The nozzle is moved to the right position

Each side of the square is printed with a single motion. For every side, the same amount of plastic is extruder, except for the last side, in which less plastic is extruded. In fact, the extruder is **coasting**, for preventing the formation of a blob of plastic in the corner of the cube.

```
G1 F5760
G1 X100.226 Y80.226 E-0.03637
G1 F5760
G1 X102.315 Y80.226 E-0.72363
G1 E-0.04000 F2100.00000
G1 Z0.950 F7200.000
```

Layer Height : 0.15 mm

Before printing the following layer, no debris or burrs must be left. In order to prevent that, acceleration and velocity are increased and the extruder is moved, of around 2 mm, over an already printed area while retracting the filament of 0.8 mm. This motion is know as **wiping.**
In the end the extruder is lifted of 0.5 mm, performing a motion called **hop**.

Figure 8: G-code for one layer of an empty cube 50x50 mm, generated with Slic3r Prusa Edition

```
; Filament-specific end gcode
G4 ; wait
M221 S100
M104 S0 ; turn off temperature
M140 S0 ; turn off heatbed
M107 ; turn off fan
G1 X0 Y200; home X axis
M84 ; disable motors
```

Figure 9: Ending part of G-code file

THE 3D PRINTER

3D printers can vary immensely in technology, shape, mechanics, electronics, number of extruders and so on. From this moment on only FDM technology will be only discussed.

From the mechanical point of view, a 3D printer is a robot with 3 prismatic joints with a not-redundant kinematic chain since, the task space only requires 3 degrees of freedom: the motion on axes X, Y and Z (orientation of the extruder is not required). At the end of the kinematic chain there is the TCP (Tool Center Point), which in this case is the extruder's nozzle. By moving the hot nozzle in the workspace, while the plastic filament is pushed in, layers are printed. Most 3D printers implement the following geometries: cartesian, coreXY, delta and scara. A key factor to highlight is the positioning of motors, which can be the bigger source of inertia in each axis: the ultimate goal is to place them, not on moving elements but in fixed position.

Cartesian 3D printers are the most common 3D printers on the market. This geometry is realized with 3 orthogonal axes which generate a workspace, or a printing volume, with a rectangular parallelepiped shape. With this geometry, different arrangements and motor's positioning are possible, but all of them have in common the following inverse kinematic:

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} k_x X_n \\ k_y Y_n \\ k_z Z_n \end{bmatrix}$$

Where X_n, Y_n and Z_n are the coordinates of the nozzle in the cartesian workspace, q_x, q_y and q_z are the joint variables, which describe the rotations of the motors for the 3 axes, and k_x, k_y and k_z are the coefficients which transform the rotation into linear movement. The coefficient k can depend on the radius of a pulley on a motor or it can depend on the pitch of a lead screw. k can also include the gear ratio coefficient. From the computational point of view, this Cartesian geometry leads to an easily computable inverse kinematic, making it compatible with the basic electronics, 3D printers are usually equipped with, that is the 8-bits AVR microcontroller ATmega2560.

3D printers are generally adopting stepper motors to realize the motions for the 3 axes. Stepper motors, which we'll be analyzed in detail in the following chapters, are chosen since they are cheap and enough precise for this application. Quoting Paul Acarnley in 'Stepping Motors a guide to theory and practice': "The essential property of the stepping motor is its ability to translate switched excitation changes into precisely defined increments of rotor position ('steps')". This capability enables the motor to run in open loop, reducing complexity and cost since no encoder for feedback is needed.

Also, the relative ease of controlling stepper motors must be considered, making it a compatible choice with a not-powerful processor.

This thesis focuses on the analysis of some phenomena occurring in Prusa's 3D printer: Original Prusa i3 MK3 (from this moment on, referred as MK3). So MK3 will be presented as an example of cartesian 3D printer, as follows:

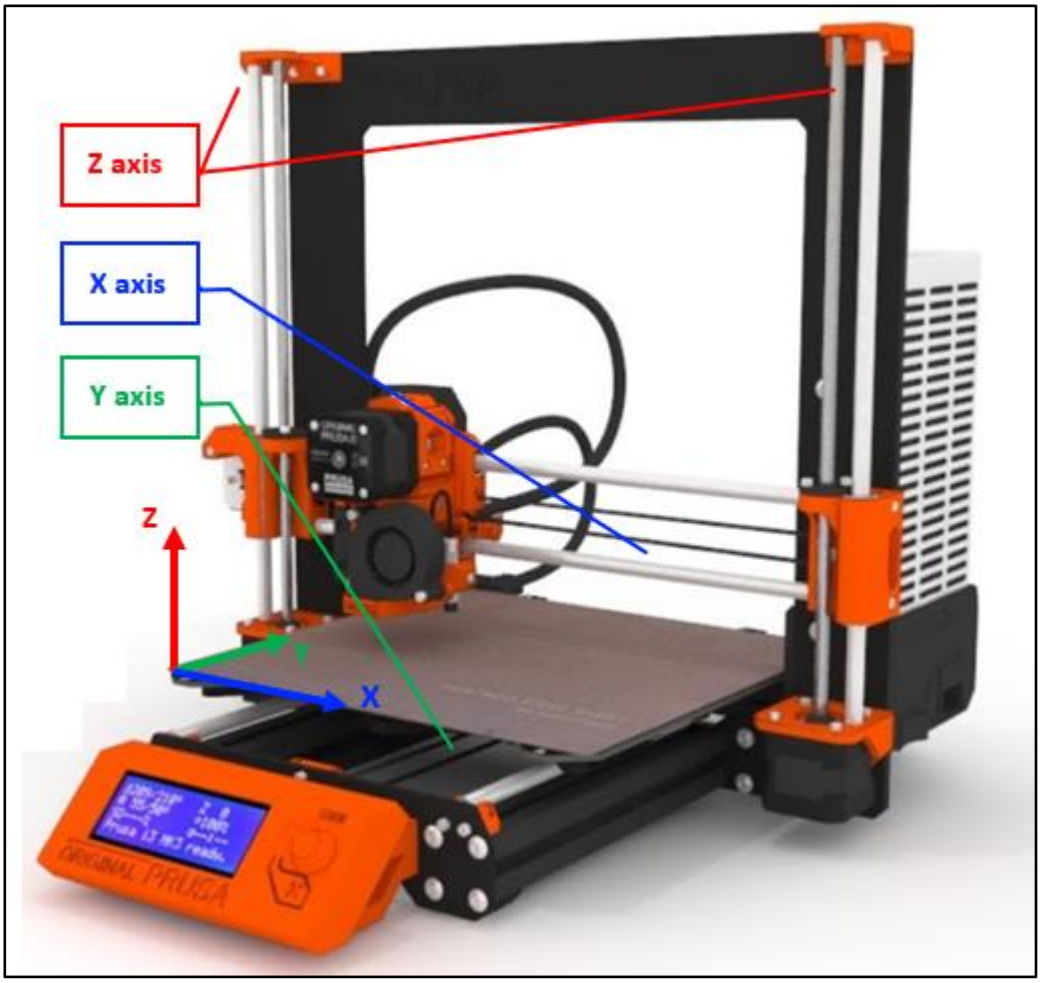


Figure 10: Original Prusa i3 MK3

MK3 is equipped with 4 stepper motors which realize the linear motion on the 3 axes. They are placed as shown in figure 11. The Y axis moves the printing bed while the X axis moves the extruder. Not every motor is in a not-moving position, in fact X axis stepper lifts along the Z axis. This positioning, however, has a negligible impact on performance for 2 main reasons:

- the additional increase of mass due to the motor on the Z axis is shared by 2 stepper motors;
- the motion on Z axis is not frequent since it mainly performs hops or layer changes. Hops are fast vertical motions needed to move the extruder between points of the same layer without

printing. Layer changes are necessary motion performed every time a layer has been deposited and a new layer must be printed;

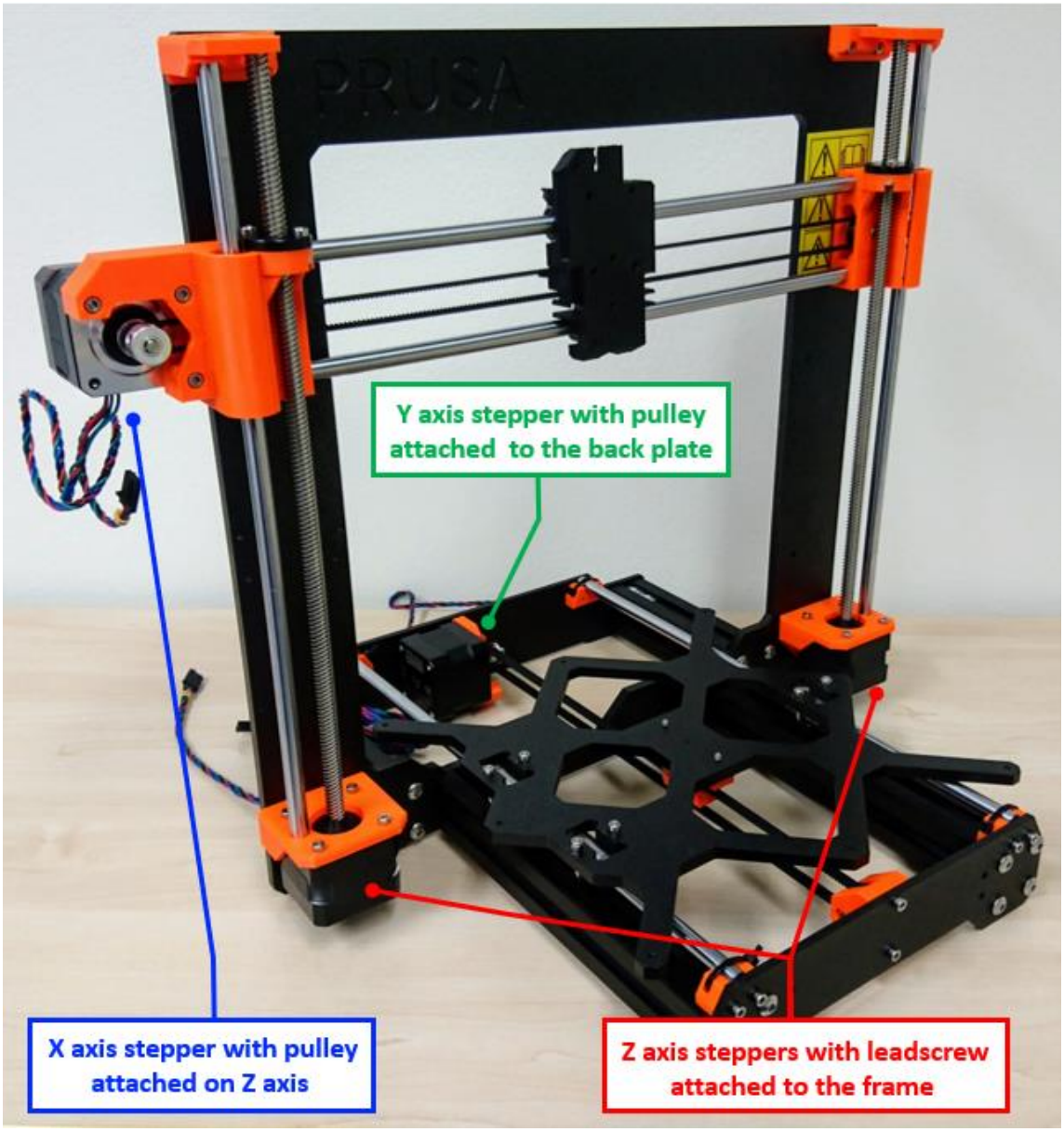


Figure 11: Partially assembled MK3 in which is possible to see motors positioning

The *k* parameters for the inverse kinematic in MK3 are defined by the number of teeth in the pulleys and by the pitch of the leadscrew. However, in this application it is more common to define how many steps are needed to move of 1 mm.

GT2 belts are used on X and Y axes, which are characterized by a pitch p_{belt} of 2 mm. Pitch in belts is defined as the distance between two adjacent teeth. X and Y stepper motor are equipped with pulleys with n_{teeth} 16 teeth. While in Z axis leadscrew is used, with pitch $p_{leadscrew}$ of 8 mm.

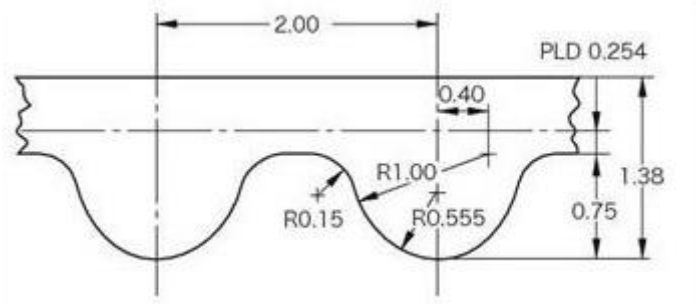


Figure 12: GT2 Belt Dimensions

In MK3, stepper motors have 200 steps per revolution S_{motor} and are operated with drivers which allow different micro-stepping configurations. In MK3 the micro-stepping S_{driver} is configured at x16.

$$\begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \begin{bmatrix} spm_x * X_n \\ spm_y * Y_n \\ spm_z * Z_n \end{bmatrix} \xrightarrow{for\ MK3} \begin{bmatrix} spm_x \\ spm_y \\ spm_z \end{bmatrix} = \begin{bmatrix} \frac{S_{motor} * S_{driver}}{p_{belt} * n_{teeth}} \\ \frac{S_{motor} * S_{driver}}{p_{belt} * n_{teeth}} \\ \frac{S_{motor} * S_{driver}}{p_{leadscrew}} \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{200\ [step] * 16}{2\ [mm] * 16} \\ \frac{200\ [step] * 16}{2\ [mm] * 16} \\ \frac{200\ [step] * 16}{8\ [mm]} \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \\ 400 \end{bmatrix} in \left[\frac{steps}{mm}\right]$$

Where *s* parameters are the number of steps needed to move of displacement $[X_n, Y_n, Z_n]$ in millimeters and *spm* (Step Per Millimeter) is the coefficient which links the previous variables.

So far, the positioning system has been discussed, though in this field it is universally recognized that having the best extruding technology is crucial. The extrusion is realized by having a stepper motor pushing the plastic filament inside the hotend in which it gets fused. Within the open-source solutions many configurations are possible, a first distinction can be made on the stepper motor positioning: Direct-Drive Extruder (DDE) or Bowden Extruder (BE).

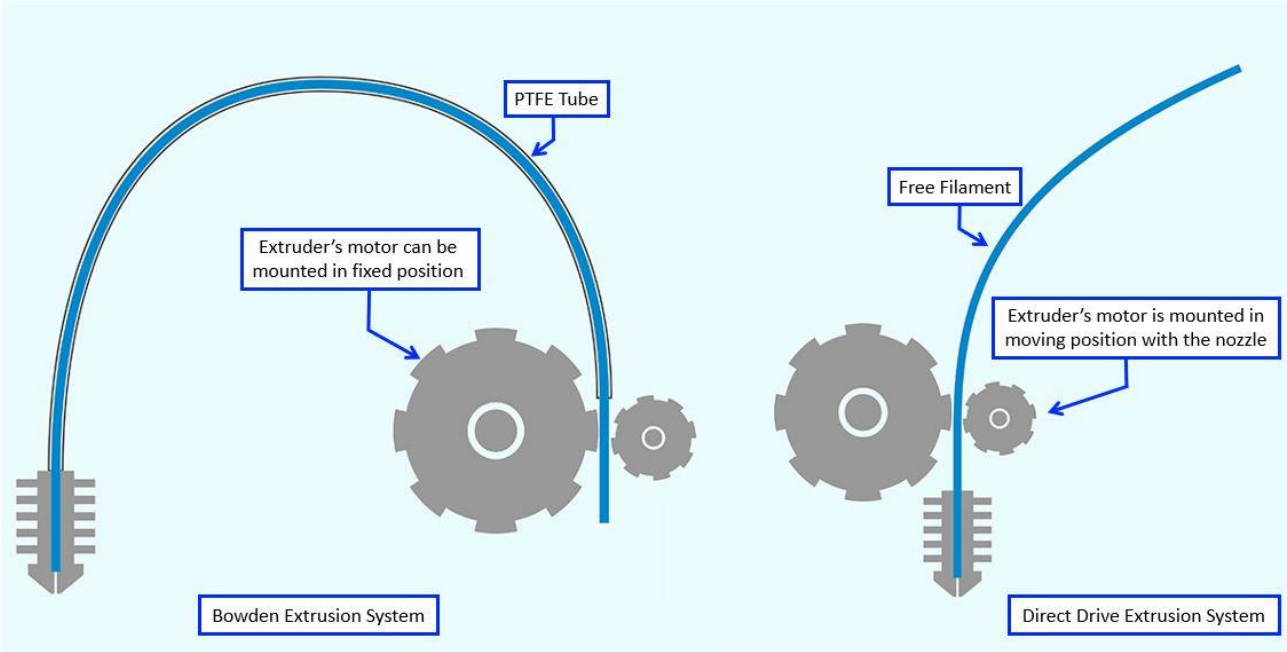


Figure 13: Bowden Extruder vs Direct-Drive Extruder

In DDS the stepper motor, which is pushing the filament in the hot nozzle, is the closest to it so that the length between where the filament is pushed and where the filament is extruded is reduced. This setup leads to a better control on the extrusion and shorter retractions. But nothing comes free, so in case the nozzle is moved (and not the bed) the motor itself participates to the motion.

In BE the stepper motor is attached to a fixed position, usually on the frame, the filament is then pushed inside the nozzle through a Bowden cable which is usually made of PTFE polymer which is chosen for it is heat-resistant, up to 300°, and has a very low friction coefficient (0.05 - 0.1 against steel). This solution allows major mass reduction on the extruder but leads to a more complex control on the extrusion and longer retractions. In MK3 a Direct-Drive system is used, and it moves on X axis. The distance between the point where filament is pushed and where the filament is extruded 83mm with a 19mm of melting zone.

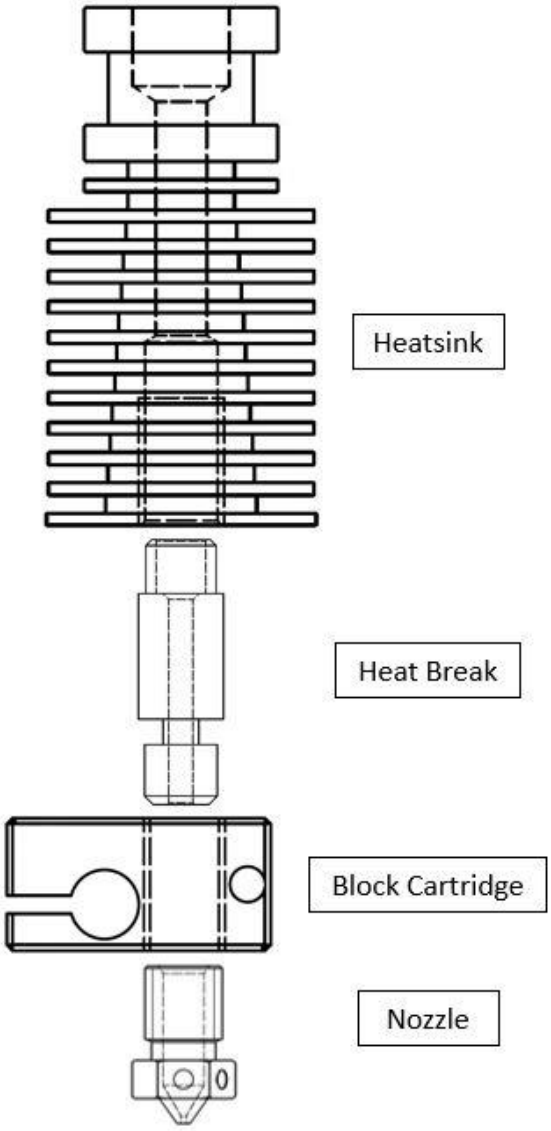


Figure 14: The market standard, V6 hotend from E3D

Not only the way the filament is pushed is critical for good results but also where the filament is fused: the hotend. The hotend purpose is to manage the fusion of the plastic filament and its extrusion. Plastic at first enters the hotend and goes through the heatsink, which usually maintains its temperature not over 45°, thanks to the fan and high surface dissipators. Then it goes through the heat break, whose duty is to connect the heat sink to the block Cartridge and to avoid heat to pass between the two previous components. In high quality printers the heat break can be made from titanium alloy to provide both stiffness and low thermal conductivity. After that, the filament enters the block cartridge, where it is fused. The block cartridge houses the heating cartridge and the temperature sensor; a closed loop is implemented so that together they provide a reliable and stable source of heat which can reach up to 450°. The aluminum block cartridge is a relatively big component, but this is done, so that it behaves like a heat tank in case some sudden heat loss occurs. Finally, the plastic is fused and can be extruded through the brass nozzle. It can have different hole's size from 0.15mm to 1mm; standard size is 0.4mm.

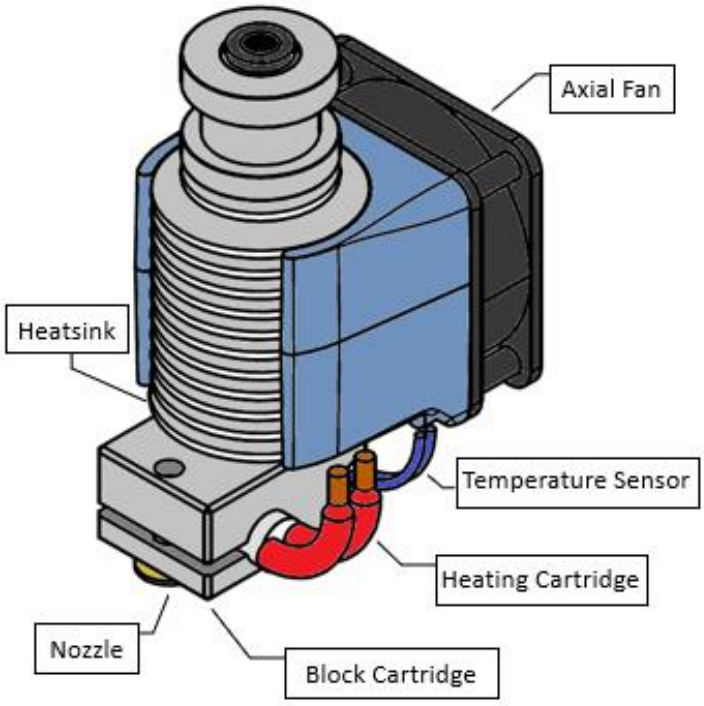


Figure 15: V6 hotend from E3D

A rule of thumb for good quality prints states that the transition of the plastic filament form solid to low viscosity must be the quicker possible or in other terms the melting zone in the hotend must be the shortest possible. The reason for this is that a long melting zone behaves like a tank of molten plastic (in bond graph terms a compliance), making the extrusion control harder.

In MK3, V6 hotend from E3D is used with custom housing and different fan.

After the previous mechanics overview, attention should be focused back on 3D printing work flow. When a G-code file is ready it can be executed, starting the printing process. The file can be transferred to the printer or via serial or via SD card. The second method usually ensure better time performance. When the file is accessible by the printer, the firmware deals with importing the G-code, inserting it in a circular buffer and then line by line it is executed. The firmware as well, is generally open source. The most used and famous one is Marlin. Quoting Marlin firmware web site: " *First created in 2011 for RepRap and Ultimate, today Marlin drives most of the world's 3D printers. Reliable and precise, it delivers outstanding print quality while keeping you in full control of the process. As an open source project hosted on Github, Marlin is owned and maintained by the maker community* "

Marlin, which is adopted in MK3 as well, deals with different tasks, the most important are:

- communication via serial port;
- reading G-code from SD or Serial port;
- implementing circular buffer;
- computing inverse kinematics;
- driving stepper motors drivers;
- temperature PID for bed and hotend;
- crush detection, power failure recovery, filament detection;
- driving HMI: LCD screen and encoder with push button;

POST PROCESSING

When an object has been printed it is not usually ready to be used or sold, since surface finishing is not usually acceptable. A further step of refinement must be undertaken: the printed object is then deburred, sanded, sprayed with multiple products (filling primer and paint). All these steps allow the print to look perfectly, how ever in other AM technology or in traditional technology such as injection molding these last steps are not necessary since surface finishing doesn't require further work.

It is important to highlight that post processing operations are one of the reasons that make 3D printing not suitable for industrial production and for production in retail shops, since post-processing is a labor-intensive task and a dedicated environment must be provided.

CHAPTER 3: Limit and accuracy of FDM technology

In this chapter the main limits of FDM technology are shown. Accuracy and control resolution of MK3 are discussed.

LIMIT OF FDM TECHNOLOGY

The main limitation in FDM technology is embedded in its core: the layers. To print a layer, a support structure is needed. Most of the time, the printing bed or previous layer support the new layer to be printed. However, printing some geometries can be really challenging since some layers can be required to be lay down in mid-air, like in figure 16 below.

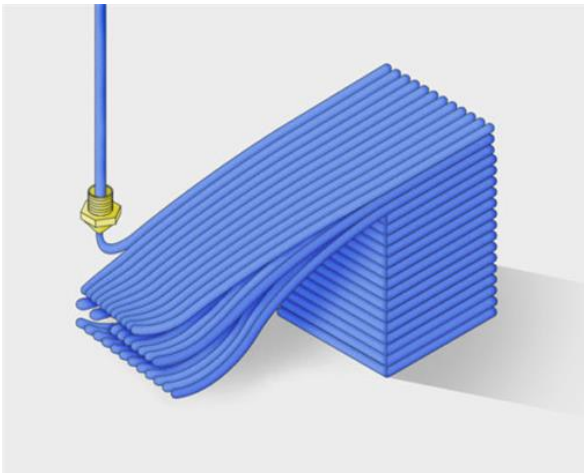


Figure 16: Object with layers in mid-air

This issue can be solved in different ways: the object can be sliced orienting the layers with a different angle, removable support structures can be created or the geometry itself must be conceived in order to avoid this problem.

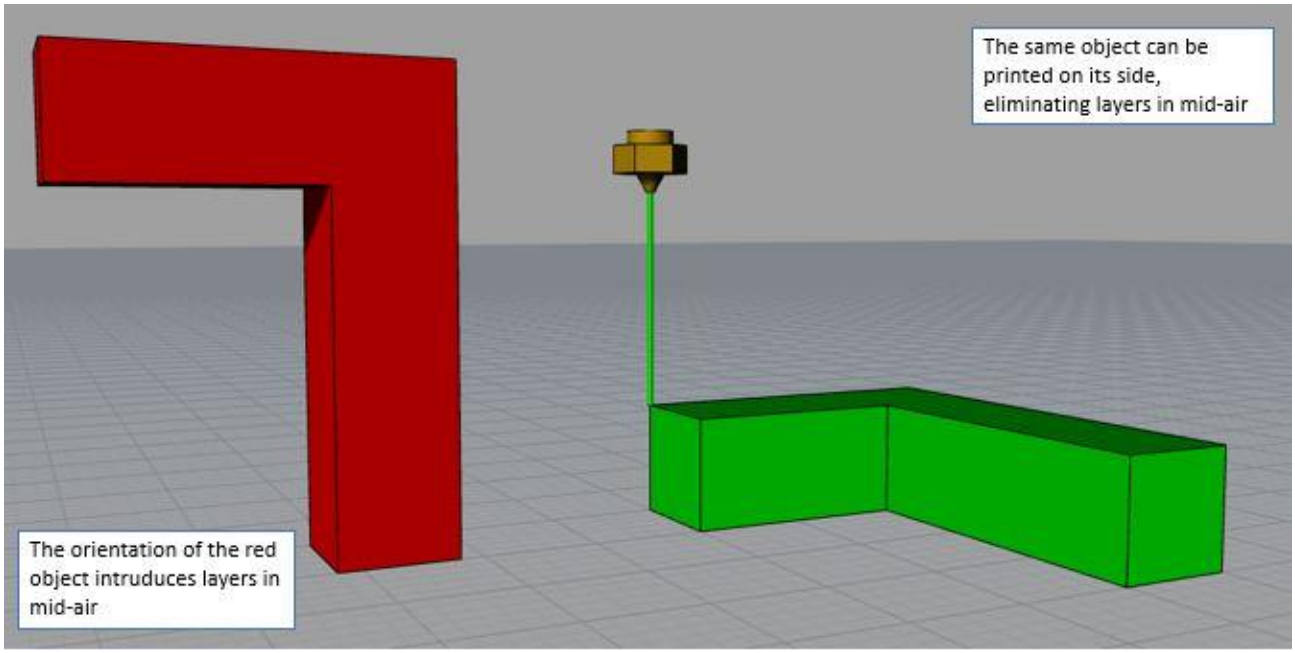


Figure 17: The orientation of the object can determine its printability

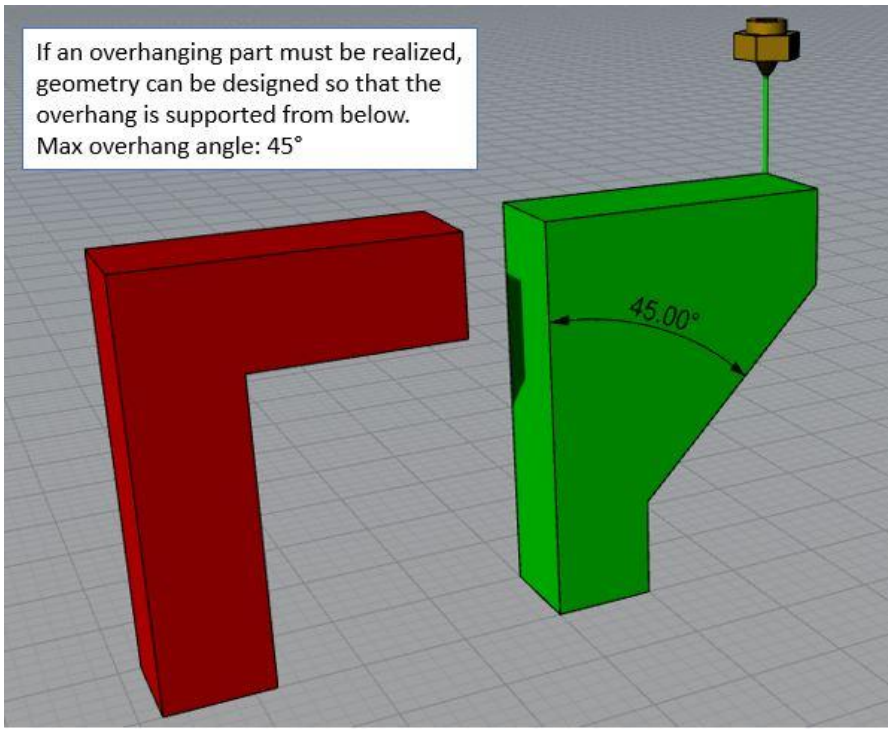


Figure 18: In the design phase overhangs should be avoided

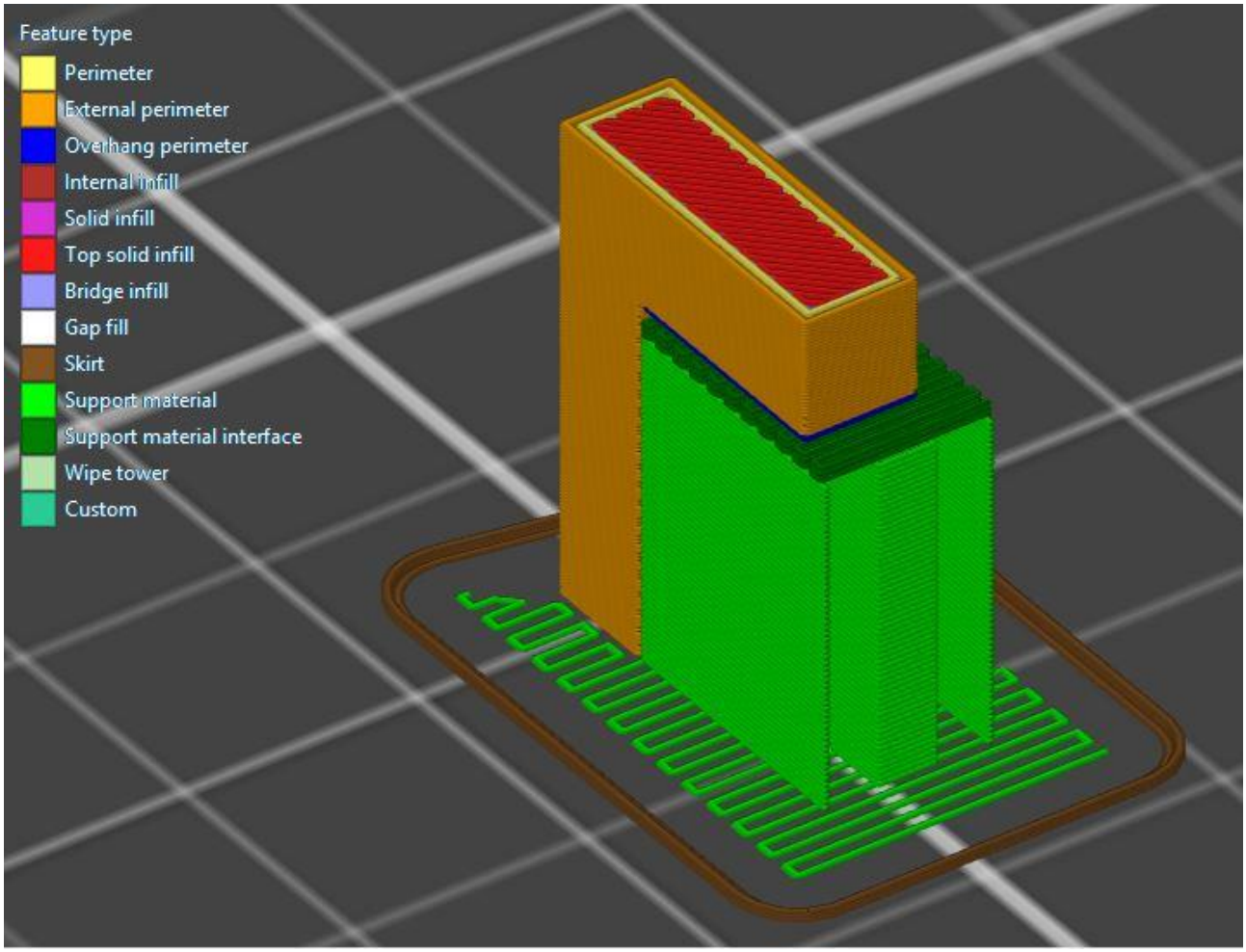


Figure 19: Preview from Slic3r of the L-shaped object; printed in this orientation, support structure is added by the slicer to guarantee printability of the overhanging part

PRECISION AND ACCURACY

The precision in positioning system can be also a limiting factor and must be taken in account; Three measures of precision can be defined: control resolution, accuracy and repeatability.

Control resolution refers to the control system's ability to divide the total range of the axis movement into closely spaced points that can be distinguished by the firmware. Control resolution is defined as the distance separating two adjacent addressable points in the axis movement. Addressable points are locations along the axis to which the worktable can be specifically directed to go. It is desirable for control resolution to be as small as possible.

Limitations to the control resolution are imposed by the electromechanical components of the positioning system and/or the number of bits used by the controller to define the axis coordinate location.

In MK3 case, from *spm* we can derive the control resolution for each axis as:

$$\begin{bmatrix} CR_x \\ CR_y \\ CR_z \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.01 \\ 0.0025 \end{bmatrix} \; in\,[mm]$$

The precision with which an addressable point is reached at its exact location is limited by several mechanical factors: play between screw and worktable, backlash in the gears, deflection of machine components, stiffness in belts, etc. The mechanical errors are assumed to form an unbiased normal statistical distribution about the control point with mean $\mu = 0$ and constant standard deviation σ along the axis of interest. Under these assumptions, nearly all the mechanical errors (99.73%) are contained within $\pm 3\sigma$.

Accuracy is defined under worst case conditions in which the desired target point lies in the middle between two adjacent addressable points. The accuracy of an axis of a positioning system is the maximum possible error that can occur between the desired target point and the actual position taken by the system. So that we have: $Ac = \frac{CR}{2} + 3\sigma$

Repeatability refers to the ability of the positioning system to return to a given addressable point. This capability can be measured in terms of the location errors encountered when the system attempts to position itself at the addressable point. Location errors are the manifestation of the mechanical errors of the positioning system, which follow a normal distribution as previously assumed. From above, it is possible to write: $Re = \pm 3\sigma$

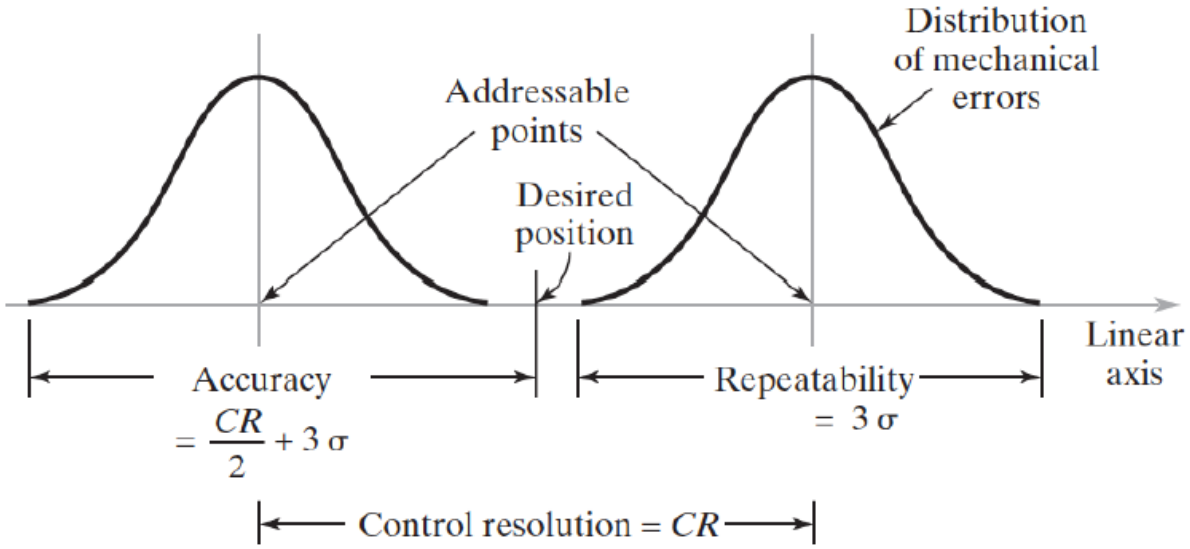


Figure 20: Precision in positioning systems

CHAPTER 4: Artifacts

In this chapter a definition of artifacts is given, artifacts are then divided with respect to their cause. Eventually few examples are shown.

ARTIFACTS

Artifacts are undesired effects in 3D printed objects in terms of surface quality, geometry or tolerances. Artifacts can be subdivided in few categories as follow:

- Artifacts due to poorly tuned or badly assembled 3D printers or caused by failed parts of the device;
- Artifacts causes by a violation of the limitations imposed by FDM technology;
- Artifacts due to phenomena which are not controlled yet by the available technology or are caused by non-idealities which, with the perfecting of the technology, become visible;

While artifacts in the first and second category are not interesting at all, since few limitations are not possible to be overcame and many others are user related, in the third category there is room for investigation and improvement. A detailed description of all the known artifacts in category 1 and the relative solutions can be found at the web site on linked by the QR Code below.

QR Code 4: Link to TroubleShooting Guide by Simplify 3D

STRINGING

A very common issue which can affect a print is stringing. The printed object shows some very thin strings of plastic which connects various points of the objects. It is usually a minor effect which does not affect quality or shape and it is easily solvable just by treating the object with a heat-gun, which basically burns the strings. Stringing can be caused by various reasons such as too high printing temperature or bad quality filament, but the main cause is a non-optimal management of retraction. When a hop motion must be performed the filament needs to be retracted, to avoid depositing material during the travel. However, if the retraction distance or the retraction speed is too small a bit of

material stays attached to the tip of the nozzle. The nozzle then travels to its target location and pulls the bit of plastic which gets thinner and becomes a string.

GHOSTING

Another artifact which is commonly affecting 3D printer is ghosting or ringing. It affects the surface quality of printed objects causing the surfaces of models to display echoes of previous features. Rapid motions of the printhead translate into vibrations of the frame which bends slightly and resonates at its natural frequency introducing inaccuracies. For example, the printhead is moving fast on the X axis, then suddenly it stops, and the Y axis starts to move. The moment of inertia of printhead is thus transferred to the frame which dissipates it with dampened vibrations which affects the print surface. Therefore, reducing print speed, jerk and acceleration can positively affects print quality. From the designer point of view, reducing the masses in motion and choosing stiffer components for the frame, can decrease the effects of this artifact.

WARPING

Warping is an artifact related to material and temperature. When printing, particularly with ABS, if an uneven cooling of the printed layers occurs, tensions inside the objects rises and bends upwards the bottom layer of the model. Warping in the most severe cases leads to an incorrect shape of the model and can severely affect the adhesion between the first layer and the printing bed. It can cause the printed object to detach from the printing bed, leading to a failed print. Warping can be solved by employing a heated bed capable of heating up to 100° and by housing the printer in an enclosure since a mild breeze can induce uneven cooling.

CHAPTER 5: Extrusion Inconsistency

In this chapter, first an idea of how the perfect polymers extrusion should happen, then, extrusion inconsistency is introduced.

IDEAL EXTRUSION

Ideally, when printing the nozzle should deposit a string of plastic with always the same cross section, at any speed the printing head is moving, like in figure below. Actually, the slicer generates G-code assuming these conditions to be true, with, in most common configuration, string dimensions like in figure below. In an ideal wall, like below, at a visual inspection, different layers would be slightly

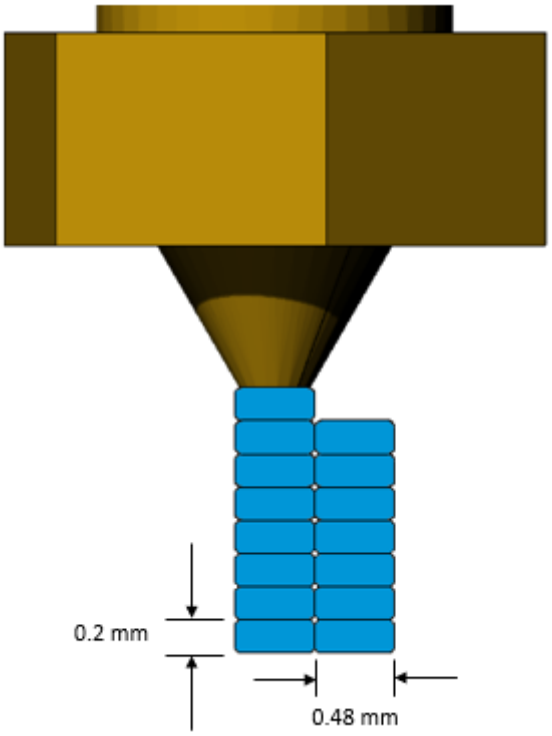


Figure 21: Ideal plastic deposition. The 0.4 mm nozzle is travelling in a direction perpendicular to the page.

recognizable despite the constant cross section, but beyond that, the wall would be flat. In case this esthetic effect must be avoided layer-height can be reduced at the cost of increased printing time.

EXTRUSION INCONSISTENCY

Extrusion inconsistency (EI) is an artifact which involves the surface finishing of vertical walls. This effect is visible on printed object when a light beam hits the surface in parallel, highlighting the superficial roughness. In other words, from one layer to the other, albeit having the same G-code and shape, the outer perimeters don't realize smooth surfaces, but very rough ones. This is caused by an inconstant cross section in the extruded plastic string, which cannot expand vertically, since it is

pushed down by the nozzle, also aided by gravity, and therefore it follows the lower resistance path, expanding laterally. The string cross section changes are due to an incomplete control of the extrusion which leads to a variation in the amount of material flowing out of the nozzle.

This artifact not only decreases dimensional accuracy but most importantly decreases esthetic quality of the printed object. This effect is partially responsible for the need of the post-processing of printed objects.

The 5th of April 2018, an issue named " Extrusion Inconsistency #602 " is opened on GitHub. Users from Prusa's community complain poor surface quality in printed object. This issue seems trivial but has both economic and performance consequences, since it goes "viral" in the community and is perceived as a major issue by users. This pushes PrusaResearch to start an investigation about causes and possible improvements. It is important to underline how this artifact is not something related exclusively to Prusa's MK3 but is something which every 3D printer in the consumer market is affected by.

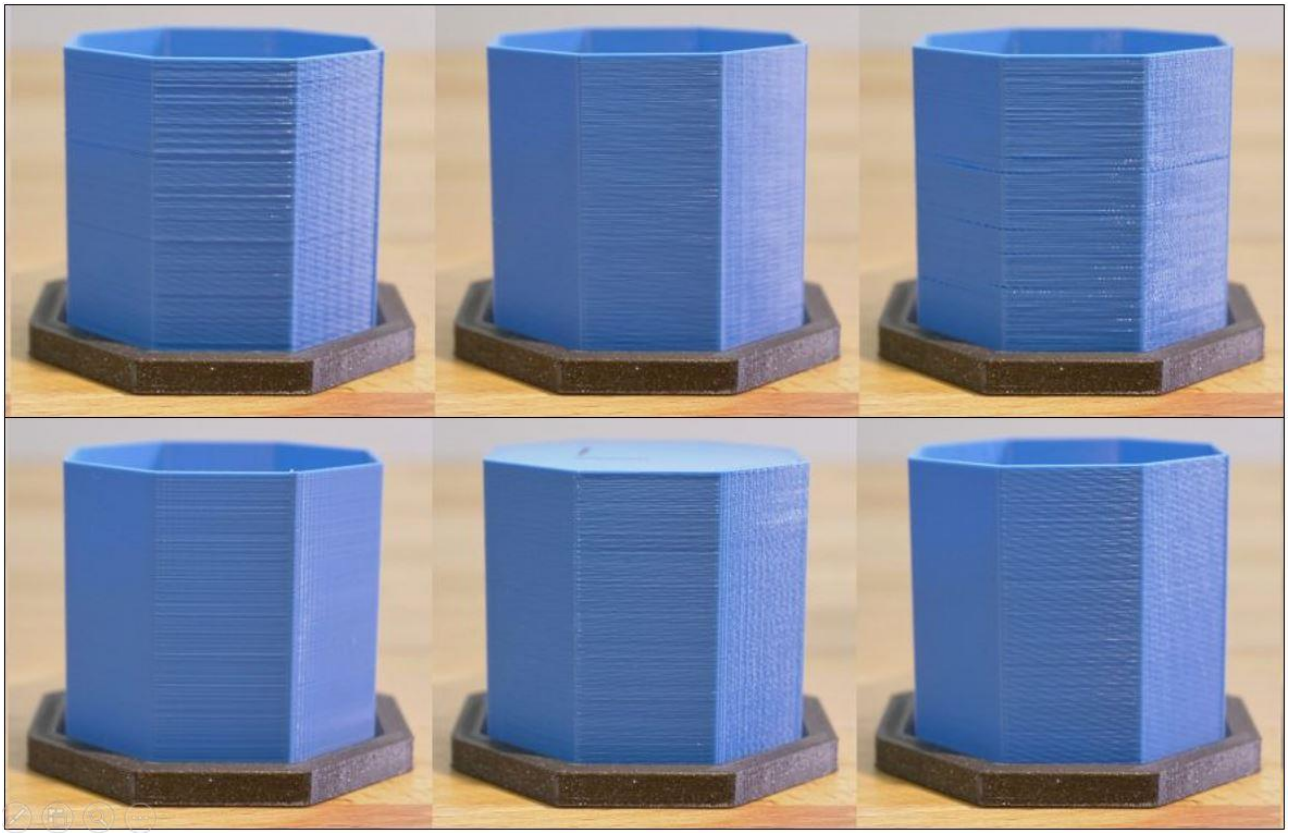


Figure 22: Extrusion Inconsistency Effects on surface quality. From PrusaResearch website

CHAPTER 6: The Testing Machine

The testing machine designed to perform tests on EI is described in terms of mechanics, electronics and software.

THE TESTING MACHINE

In order to further investigate extrusion inconsistency a testing machine (TM) is designed to understand what happens inside the melting chamber. Since uneven amount of plastic is causing EI, the TM aims to collect data about pressure inside the nozzle, however, since the melting chamber is small, no direct data can be taken directly from it. It is not possible to find any off-the-shelf sensor small enough to fit inside the melting chamber without drastically changing the system. Furthermore, it would need to work at high temperature (200° - 260°) which would make things even harder in terms of tolerances and sealing. Then a different approach is adopted: the force which the filament, pushed by stepper motor, exerts on the inner bottom walls of the nozzle, is measured. This is accomplished by cutting in half the standard extruder and by placing in the middle 2 load-cells to sense the force.

For this reason, the 3D printed parts of the extruder are redesigned to house the 2 load-cells. It is necessary to allow the load-cells to deform to let them work and it is important to strengthen the structure to avoid corrupting the measurement due to its deformation. In order to enhance the stiffness of the parts which are not supposed to deform two, 3mm thick, aluminum plates are adopted as shown in figure 23.

Load cells were chosen with a maximum load of 5kgf, since it was already known that the maximum force needed to extrude filament was around 5kgf. The output of a load cell is an analog signal proportional to the deformation and hence proportional to the applied force. The signal is acquired through the Analog-to-Digital Converter HX711 which reads the voltage difference in the Wheatstone bridge on the cells and then converts it in a 24 bits digital word which can be acquired by a microcontroller via serial communication. Data are sampled at a frequency of 80 Hz.

For further understanding of the phenomenon the angular speed of the extruder's stepper motor is also desired. This data is acquired by placing the magnetic digital encoder AS5048 on the back of the stepper, which reads the angular position of the shaft; The angular speed is then computed by derivation in time.

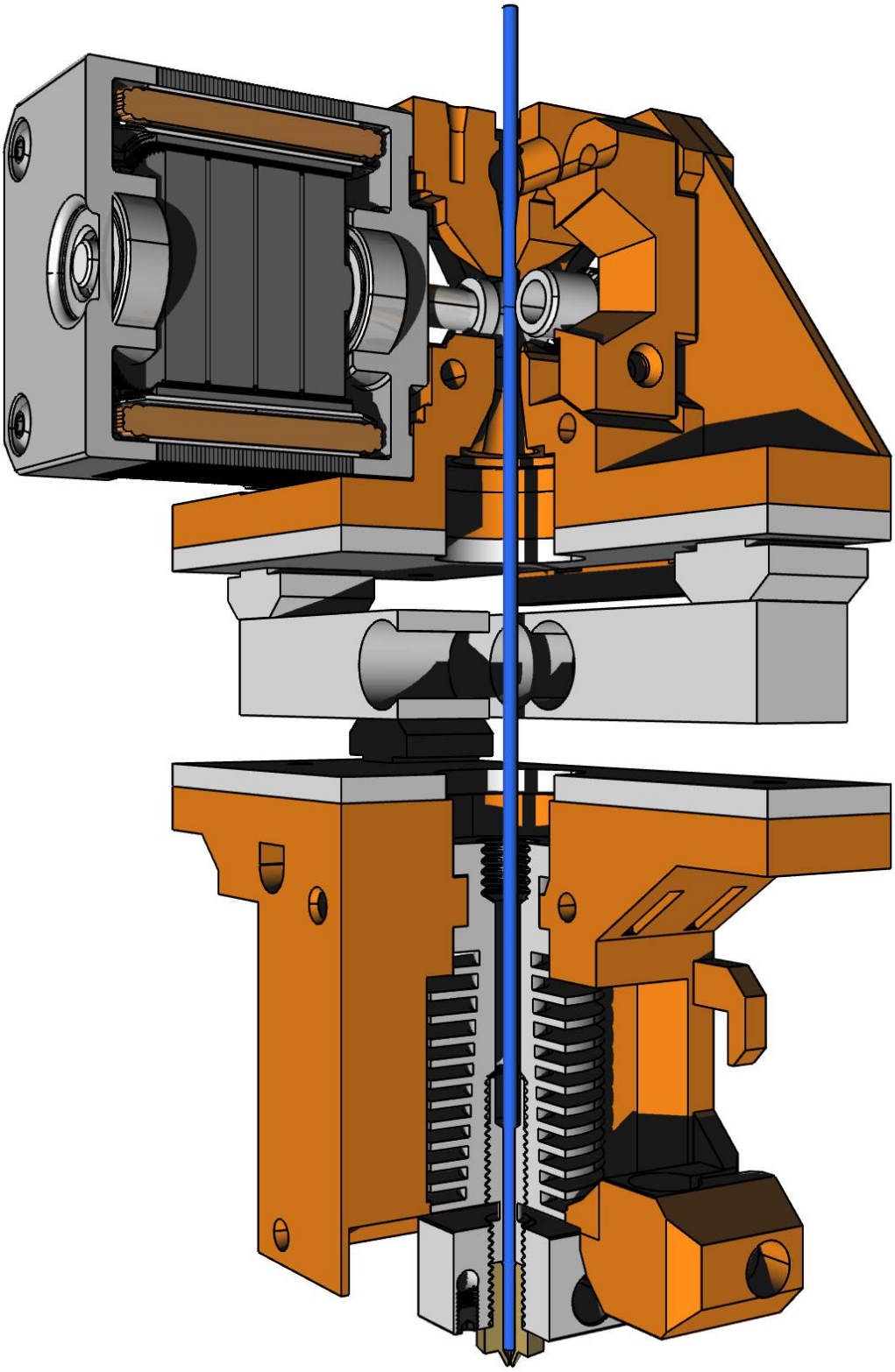


Figure 23: Section of the testing machine

A piece of firmware is written for the micro-controller (uC) Atmega2560 to collect samples from the load cells and the encoder and then send them via serial to PC. In the uC data is not manipulated to avoid any approximation error, due to the 8bits architecture. Serial port is then read by a Matlab script which stores, refines end finally plots graph with the acquired data. With this equipment, several tests are run to understand the cause of EI, minor modification are needed for each test.

Before starting testing, the conversion factor for each load cell must be found. The conversion factor transforms the 24 bits digital word from the ADC in the number representing the force measured in gram-force. Each load cell is loaded with a known weight in the same spot the force will be applied in the real tests, in steady state conditions, to avoid reading disturbance due to forces caused by vibration. The output of the ADC is acquired for few seconds in Matlab via the uC, storing 1000 samples. Finally, the conversion factor is computed as follows for each sample and then a final value is obtained by computing the mean value:

$$[gramsForce]_i = \beta_i * [24\ Bits\ digital\ word]_i \rightarrow \beta_i = \frac{[gramsForce]_i}{[24\ Bits\ digital\ word]_i}$$

$$\beta = mean(\beta_i)$$

The accuracy of the readings of the scales are under 1 gram-force making it suitable for the application. Finally, it is important to underline that for every test the load cells must be initialized, which means that whenever a test is started the condition of zero load must be found, to have reliable data. Changes in temperature or weight of the hanged mass to the load cells can occur so an automatic routine is executed at the beginning of every test to achieve a zero-load condition.

The encoder on the other side needs no initialization, it returns the angular position in terms of a 14 bits unsigned integer a simple conversion can be directly implemented, furthermore no other electronics is needed since the value of the angular position can be acquired via I^2C interface.

A further step, before starting testing is to establish the range in which the extruder works. Different units are used in the slicer and in the firmware to describe and limit the extrusion. Extrusion can be described in terms of Flow Rate $[mm^3/s]$, which means the amount of plastic in mm^3 extruded per second, or feed rate or linear velocity of the filament in $[mm/s]$. It is also important to convert those previous unit in angular velocity for the stepper motor to be able to driver the motor at the proper speed during the experiments.

From the firmware the maximum velocity for the extruder stepper motor is set to 120 mm/s, which is not relevant in our case since this value refers to the retraction speed. From the slicing software, it

is possible to infer the fastest extruding speed required to print an object for MK3, which is expressed in term of flow rate of plastic. Considering a filament with 1.75 mm of diameter, the equivalent radius of the gear gripping the filament of 3.6378 mm, a stepper motor with 200 steps per revolution and a 32 micro-steps per step, the maximum extrusion is achieved at:

	Flow Rate $[mm^3/s]$	Feed Rate $[mm/s]$	RPM	RPS	$steps/s$
Absolute Maximum for PLA at 210°	15	6.236	16.370	0.2728	1746.170
Absolute Maximum for PET at 240°	8	3.326	8.731	0.146	931.291
Absolute Maximum for ABS at 255°	11	4.573	12.005	0.200	1280.525

More than 10 different tests are run with the previously explained equipment. Every step of the work flow is considered and analyzed, since at the starting of the investigation both slicing software, Marlin and mechanics are considered possible causes of the artifact. In the following chapters a summary of the most important tests and their results is reported (the order in which tests are reported is not necessarily the order they were executed).

CHAPTER 7: Investigation on the slicing software

In this chapter the modification to the TM, the MatLab script and the results produced are shown. The aim is to test if, and to which extent, the slicing software is the cause of EI.

THE POSSIBLE CAUSES TO TEST

For what concerns the slicing software, the major concerns are about the motion on the E axis, that is the stepper motor on the extruder. The Gcode which is executed by the printer could be wrongly computed or unnecessary lines might introduce unwanted behavior. It is interesting to synchronize the data from the load cells and the encoder with the execution of Gcode instructions, so that it is possible to check what is the impact of each instruction on the data collected by sensor.

CHANGES IN THE TESTING MACHINE

The idea is to print an object with an easy geometry so that it is easy to follow, interpret and analyze the Gcode. During the print, Marlin notifies the uC, reading the sensors, that a Gcode instruction has been successfully executed. This feature is implemented considering that in the firmware, Gcode instructions are read from the SD card and inserted in a circular buffer. Motion instructions are then read by the motion planner which has a buffer as well. Every time the motion corresponding to a Gcode line is executed, it gets discarded from the buffer. In the function `plan_discard_current_block()` is then added a line of code which toggles an unused pin. The pin is read by the uC with an interrupt which consequently writes on the serial bus the status of the pin.

A Matlab script in then responsible for collecting in real-time the data from the buffer of the serial port. The same script also supervises the correct initialization of all the procedures needed to collect data, all the implemented steps are shown below:

1. The OS on the laptop is instructed to disable standby mode, since data collection can last up to 2 hours.
2. The Gcode file which is printed is analyzed to understand how much memory to allocate knowing the 80Hz acquisition frequency. Memory is then allocated to speed up the process and to avoid buffer saturation.
3. MK3 Serial Port is opened to start to communicate with the printer.
4. Heaters in the printer are turned on.

5. When the heating process is finished, the filament is retracted of 5 mm
6. uC Serial Port is opened
7. A command via serial port is sent to uC to start a routine to find the zero-load condition for the load cell
8. 24 standard Gcode instructions are sent via serial port to MK3 to initialize it and start the print
9. G4 S5 is sent to MK3 to force it to wait 5 seconds
10. A Gcode command (M23 *'Name_of_file'*) is then sent to MK3 to open the file in the SD with the name specified in the command. It is followed my an M24 command to start printing.
11. The uC is then waken up to start the samples collection and the following forward via serial to Matlab.
12. Serial port is scanned multiple times a second to collect the samples.
13. Collection is over, serial ports are closed, the OS is allowed to go in standby mode

Once the data collection is over, a script in Matlab, knowing the executed Gcode and which line is inserted in the motion planner buffer and consequently discarded and acknowledged can synchronize commands with the data.

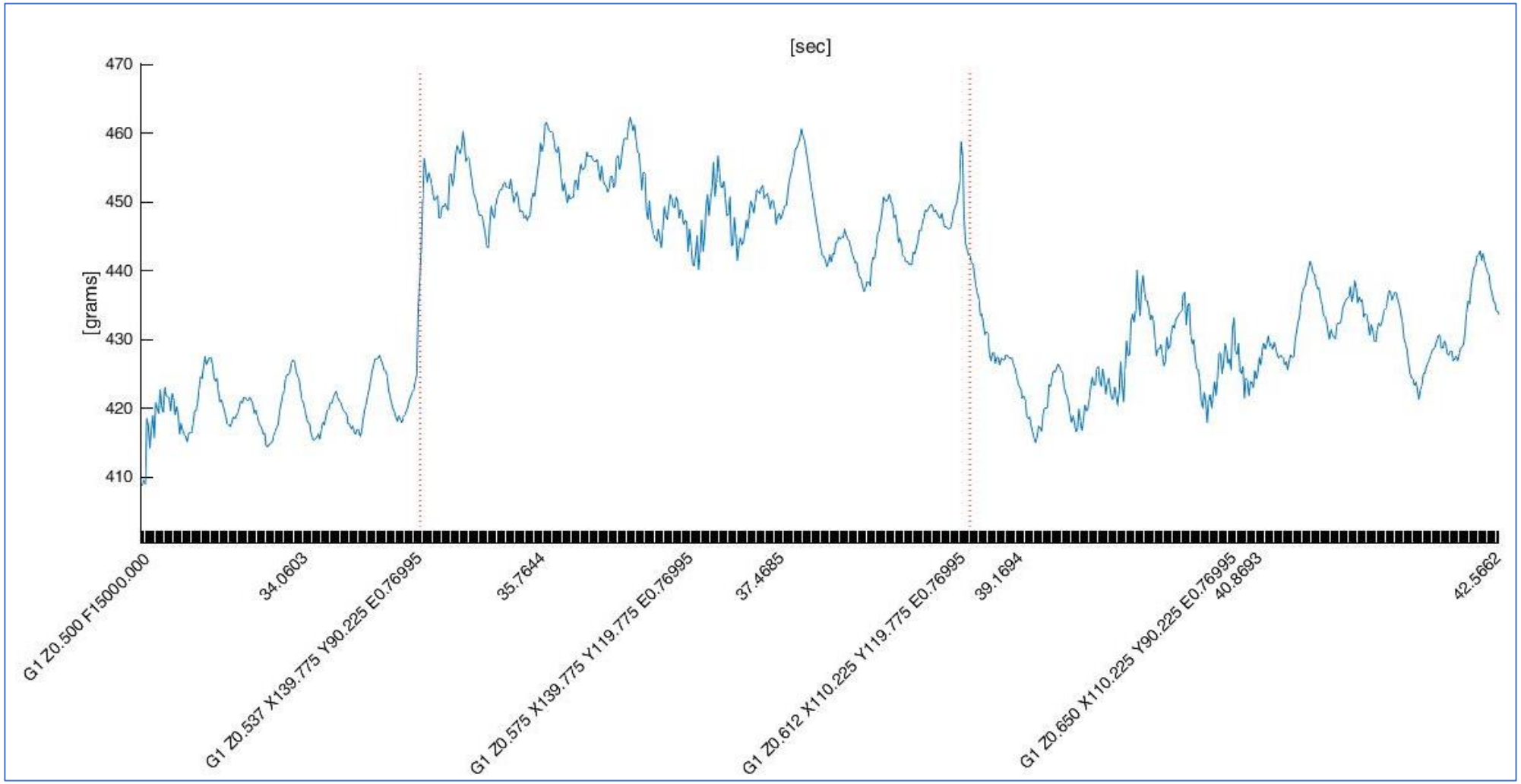
THE TEST AND THE RESULTS

For this test, two main geometry are chosen: a cube and a cylinder. Both the geometries are printer with a single outer perimeter, no infill and no bottom and upper layers. Two different printing style are chosen, the standard one with whipping and coasting and the vase mode. Vase mode is a printing style where only one outer layer is printed, and each layer is not at constant height, but it describes a spiral.

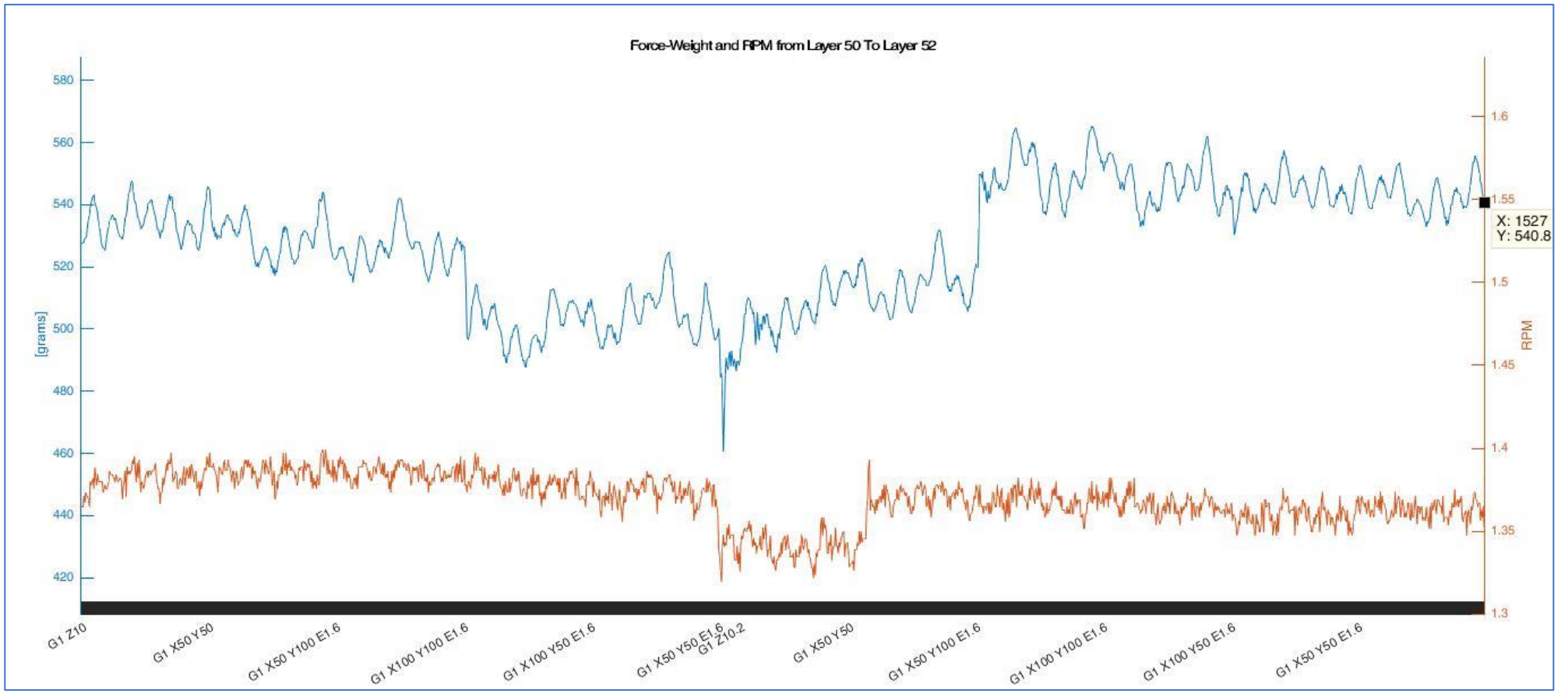
The cube has an easy Gcode with few lines for each layer making it very easy to understand, but nothing comes free, since the sudden direction change affects the readings of the load cells. In fact, the sudden change in direction needed to print every side of the cube leads to a complete stop of an axis and the start of the motion on the other. The forces needed to accelerate and decelerate can be picked up by the load cells which then results in an increase or a decrease of the signal as shown in the following Graph 1. Knowing that, they can be discarded when analyzing the data.

Acceleration effects can be neglected in the case of a cylinder since the X axis motion is much smoother, in fact, when printing a cylinder, each layer is a circle, which, if nozzle printing speed is constant, leads to a sinusoidal motion on the X axis.

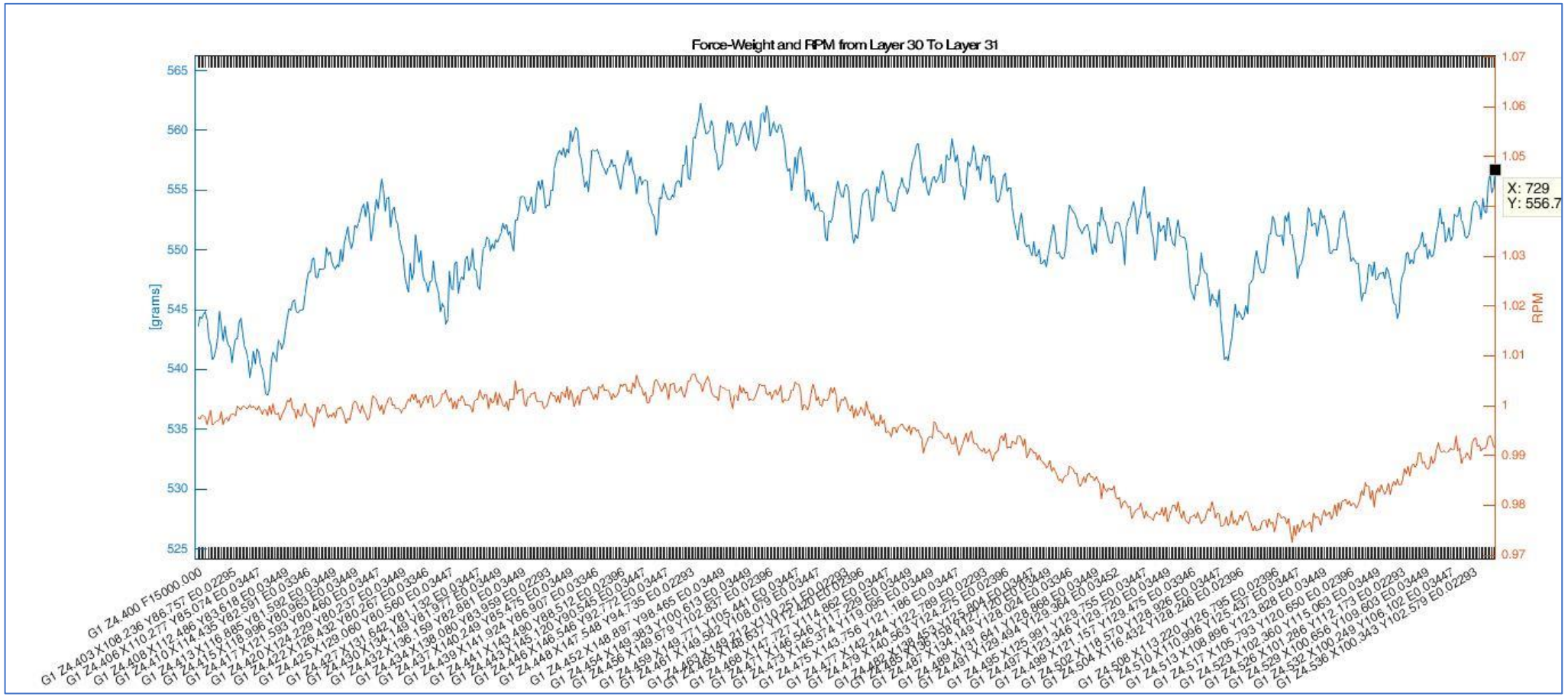
In the following Graph 2 and Graph 3, data from a single layer of a cube and a cylinder respectively, are shown. The cube is 50 mm by 50 mm printed in normal mode while the cylinder is printed in vase mode. It is possible to observe that no out of ordinary behaviors occur which can be directly caused by wrongly computed Gcode instructions. From the tests performed with this setup it is possible to conclude that EI is not caused by the slicing software.



Graph 1: Load-Cells Signal in time with effect of the acceleration on the x axis, data are subdivided layer by layer. Vase mode is used.



Graph 2: Two consecutive layers of a 50x50 mm Cube printed in normal mode.
The graph display 1527 samples which means approximately 19 seconds



Graph 3: One layer of a Cylinder printed in Vase Mode.
The graph displays 729 samples which means approximately 9 seconds

CHAPTER 8: Investigation on Marlin Firmware

In this chapter an overview of the test conducted con Marlin is presented. Mainly tests focuses on the possible errors in the timing for pulse to the stepper driver

MARLIN TASKS AND PULSES GENERATION

While printing, the firmware deals in real time with multiple tasks such as the steps signals for the 4 axes, the PIDs for the heated bed and the hotend, the Gode reading from the SD card, the inverse kinematic computation to go from the Gcode to the pulses for the stepper motors. The microcontroller on which Marlin runs is an 8 bits ATmega2560 with 16Mhz frequency of master clock. At the time the investigation was carried out, one of the major concerns was that the uC could be overloaded resulting in a bad timing in the pulses for the stepper motor driver.

Stepper motors are moved with drivers which interface with the uC through multiple registers, accessible via SPI and 3 pins for every driver: Step, Direction and Enable. The more crucial one is the step pin which must receive a voltage pulse to cause the motor to rotate one step. The pin connected to step pin of the driver is set high and low with a t_{ON} of at least 1 µs every time a step must be performed. This procedure is done in an Interrupt Service Routine (ISR) which is called according to the desired position and velocity profile and acts, at the same time, on the pins related to every motor. The ISR triggers when a dedicated timer overflows and sets a flag which consequently generates an interrupt. The ISR is executed with variable frequency which is computed every time in the ISR itself. This is done by setting, time by time, the proper number from which the timer starts to count from. The frequency computation must take into consideration that 4 different axes (Z axis has 2 stepper motors which rotate identically) must move at the time and thus pulses must be provided in real time for all the motors. Therefore, the ISR is triggered, time by time, according to the time imposed by the fastest rotating motor which requires the highest pulse frequency and thus the shortest time interval between two consequent ISRs. This approach leads to the fact that slower rotating motors receive a pulse not exactly at the ideal pulse time but in the ISR which minimize the error between the ideal pulse time and the time imposed by the fastest motor. So, this means that a discretization in time takes place and thus an error is introduced. The previous strategy for the generation of pulses can be addressed as an implementation of a Bresenham algorithm. For clarifying the previously explained phenomena, two tests were carried out.

PULSES PRODUCED BY FUNCTION GENERATOR

In a first test the E axis (extruder motor) is driven independently from the X and Y axis in order to exclude that the possible cause of EI, is the discretization error introduced by the stepping algorithm. To do so, the driver corresponding to the E axis receives as input a pulse train produced by a function generator, which can be assumed to be perfectly shaped. Since the frequency of the function generator is set manually, it can't be changed runtime accurately and with proper timing and thus it is set to a constant value. Therefor a constant linear speed of the nozzle is required during the printing process. This requirement entails that a cylindrical shape is chosen to be printed. Furthermore, in order to have a continuous trajectory of the nozzle, the cylinder is printed in vase mode, where the desired geometry is realized as a spiral. The cylinder, obtained with the changes in the printer previously exposed, showed the same issues of any other printed objects, suggesting that the timing on the E axis is not the cause of the EI.

SWAPPING MARLIN WITH KLIPPER

The first test doesn't exclude that the discretization and the possible faulty timing of the pulses' train on X and Y axes, can be the cause of EI. Therefore, a second test is performed and consists in changing the firmware the printer executes: Marlin is substituted with Klipper firmware. Klipper is made by two parts: one piece of software runs on a host machine and the other piece of firmware runs on the microcontroller of the printer. The host machine can be a PC running Linux or, like in the current test, a Raspberry Pi. The host software reads and interprets the Gcode, computes the inverse kinematics, generates the printer movements and build a schedule of events which consists of when and which motor has to step, without implementing any Bresenham algorithm. In fact, the uC since no kinematic computation has to be performed, can avoid steps discretization. Finally, the host software compresses those events and transmits them to the uC. The firmware running in the uC, just receives the events, and can simply execute each event at the requested time.

With the previously described changes, few test prints are produced. After a visual inspection of the printed geometries no major changes are noticed, since it is still possible to see EI artifacts. In conclusion the firmware and any issue related to it can be excluded to cause EI.

CHAPTER 9: Investigation on Mechanics

In this chapter a summary of all the tests focusing on mechanical components involved in the extrusion is reported. At first, a list off all the tests carried out is reported, followed a detailed description of the most important results.

TESTS ON MECHANICS

A list of all different tests carried out can be found below, it is then followed by an explanation of most significant tests with their results.

- **TEST 1-3:** PLA, PET and ABS are extruded from 0 to 19 rpm, above their maximum extrusion speed. No modifications are made to the testing machine (TM). Results are reported below;
- **TEST 4:** ABS is extruded from 0 to 19 rpm, above its maximum extrusion speed. TM has no changes except for the removal of teeth on the driven gear. Results are reported below.
- **TEST 5:** ABS is extruded from 0 to 19 rpm, above its maximum extrusion speed. Different way to tighten the gears on the filament are tested. It is possible to conclude that gear's tightening has no effect on EI.
- **TEST 6:** ABS is extruded from 0 to 19 rpm, above its maximum extrusion speed. The TM is equipped with high end heatbreak, with low friction coating in the inner wall used to prevent clogging. It is possible to conclude that it doesn't affect EI.
- **TEST 7-9:** PLA, PET and ABS are extruded from 0 to 19 rpm, above their maximum extrusion speed. The TM is modified by doubling the length of the melting zone. It is possible to conclude that different melting zone lengths don't affect EI.
- **TEST 10:** PLA is extruded from 0 to 19 rpm, above its maximum extruding speed. The extruder motor of the TM is swapped with a stepper motor equipped with a 5.18 gear ratio. Teeth on the driven gear are removed making it an idler. Small Oscillations still occurs, but with different frequency.
- **TEST 11:** A specially prepared spool of filament with ±0.015 mm tolerance is extruded from 0 to 19 rpm. The TM uses the same components of the previous test. No changes can be seen in EI albeit the use of the special filament.
- **TEST 12:** A specially prepared spool of filament with ±0.015 mm tolerance is extruded from 0 to 19 rpm. Extruding temperature is raised from 215° to 230°. The TM uses the same components of the previous test. No changes can be noticed in EI.

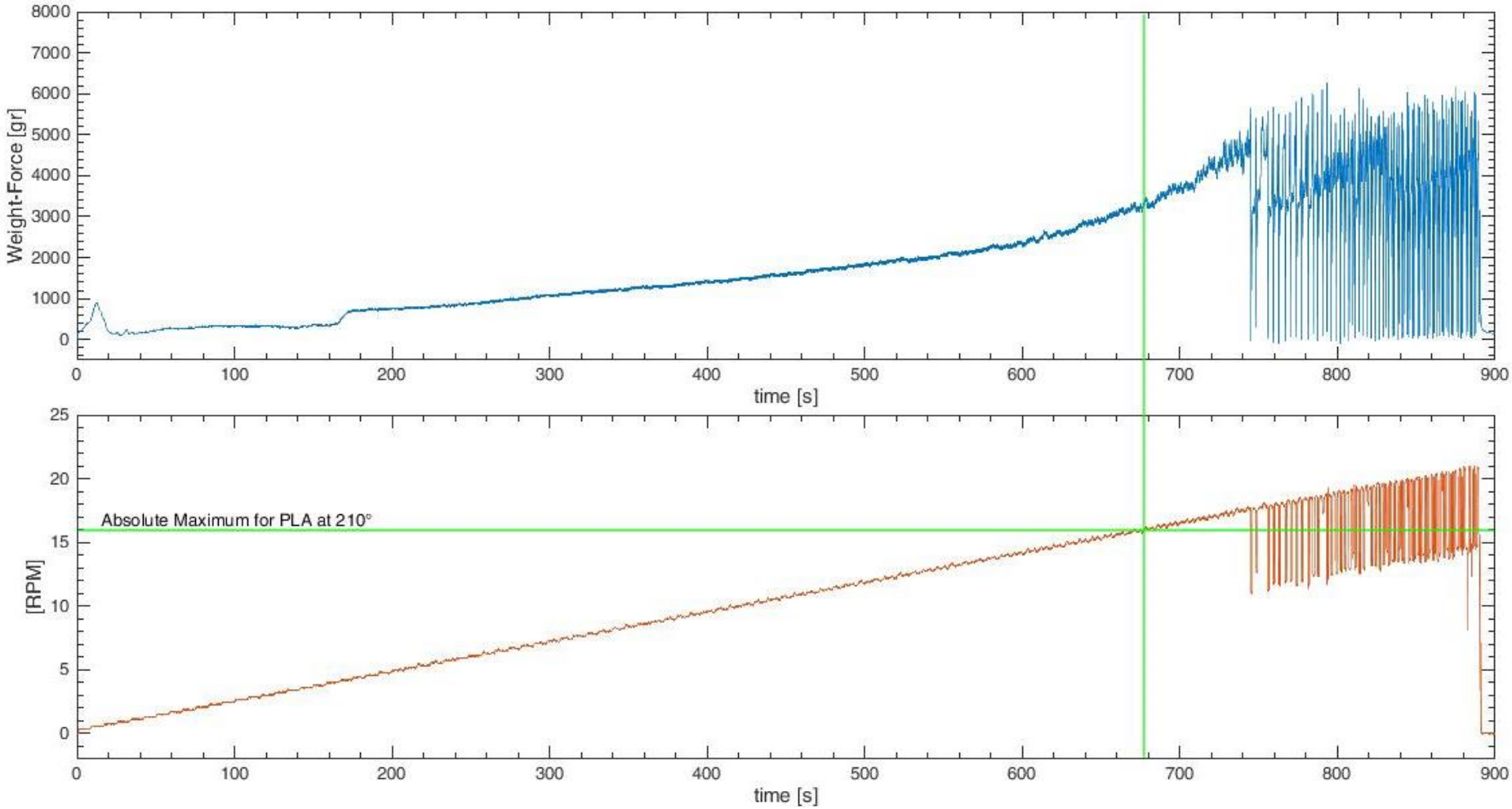
- **TEST 13:** A specially prepared spool of filament with ±0.015 mm tolerance is extruded from 0 to 19 rpm. The TM uses the same components of the previous test except the nozzle which is swapped with a different one which has 0.6mm hole. No changes can be noticed in EI, thus excluding that EI is related to the nozzle hole size.
- **TEST 14:** PLA is extruded from 0 to 19 rpm, above its maximum extruding speed. TM is equipped with various Stepper Motor. Different high frequency, small amplitude oscillations can be observed.
- **TEST 15:** The MK3's extruder is mounted on high-end 3D printer with sturdier mechanics, and print tests are made. No visible enhancement can be seen on print test, thus excluding that EI is due to vibration on the printer frame.

MOST RELEVANT RESULTS

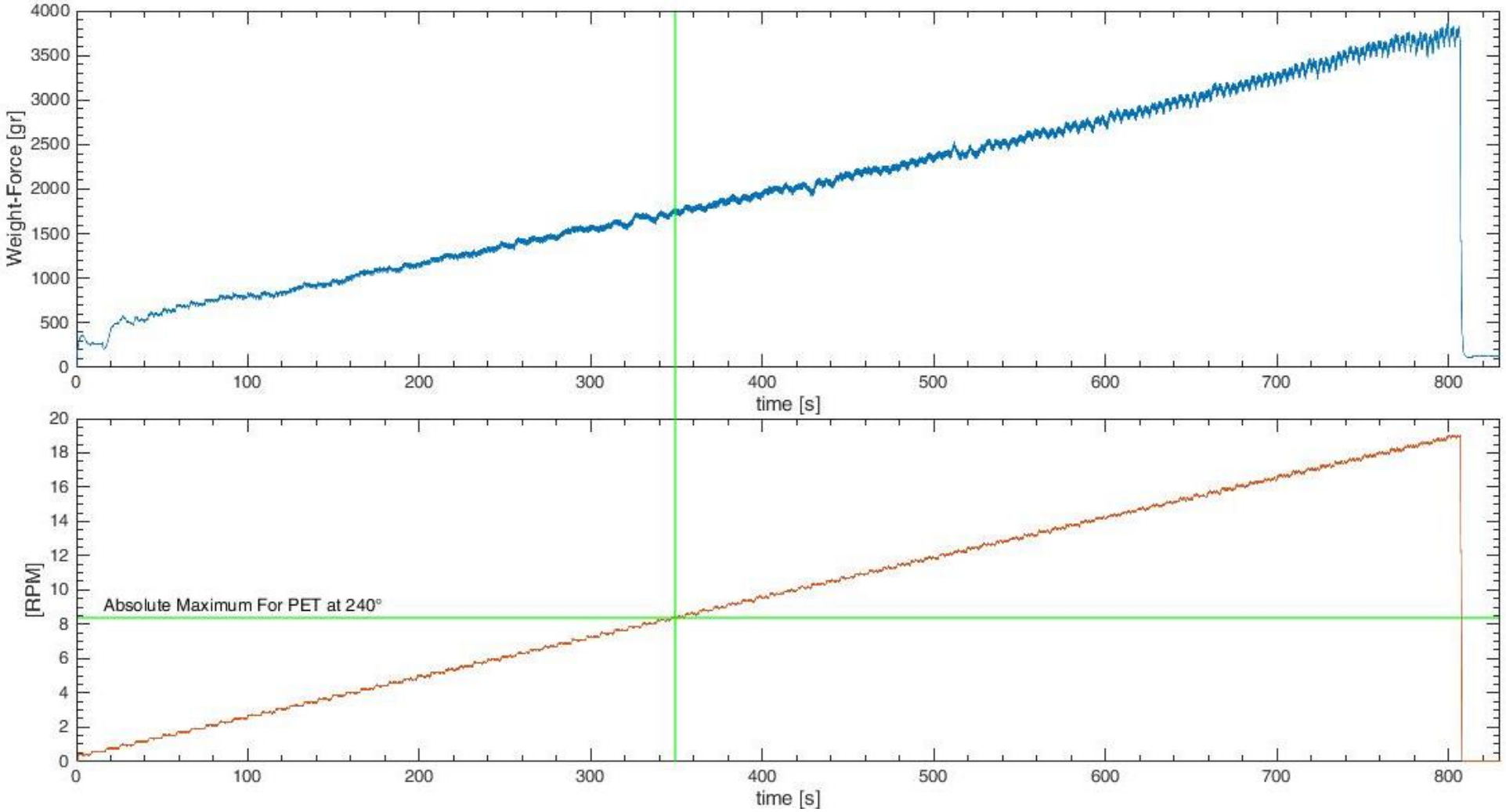
As first thing, the testing machine, with the same hardware and software explained in chapter 6, is employed to acquire data while extruding both PLA, PET and ABS. The stepper motor of the extruder is driven by an external driver receiving a pulse train generated by and a timer in the ATmega2560. The frequency of the timer is slowly increased so that the angular velocity of the stepper motor goes from zero to over 19 RPM, which over standard working condition of the E axis for the 3 polymers. The collected data are plotted in Graph 4 to Graph 6 which show the force required to extrude the filament in gram-force and the angular velocity of the stepper motor in RPM over time in seconds. The green horizontal lines in the graphs represent the maximum velocity which is imposed to the printer by the Gcode, while, with the green vertical lines, the point in time in which the limit is reached is reported in the graph regarding the extruding force.

The first thing to notice is that 3 different situations can occur when RPM are higher than the absolute maximum extrusion speed. In the PET case, when the rpm increases beyond the limit no grip loss or steps loss occur. To have a better understanding, it must be remembered that the deployed stepper motor, according to the provided datasheet, has a holding torque 5 Kg*cm. In the case of ABS, it is possible to notice that, while no step losses occur, after the time instant the angular speed reaches 15 rpm, the extruding force doesn't increase at the same rate of the RMP and at the end it starts to oscillate towards lower values very rapidly which is a sign of grip loss. Finally, for PLA since the extruder force reaches values close to 5 Kg-force, not only it is possible to notice a grip loss but also step loss which is represented by the rapid decrease in RPM towards the end of the time axis.

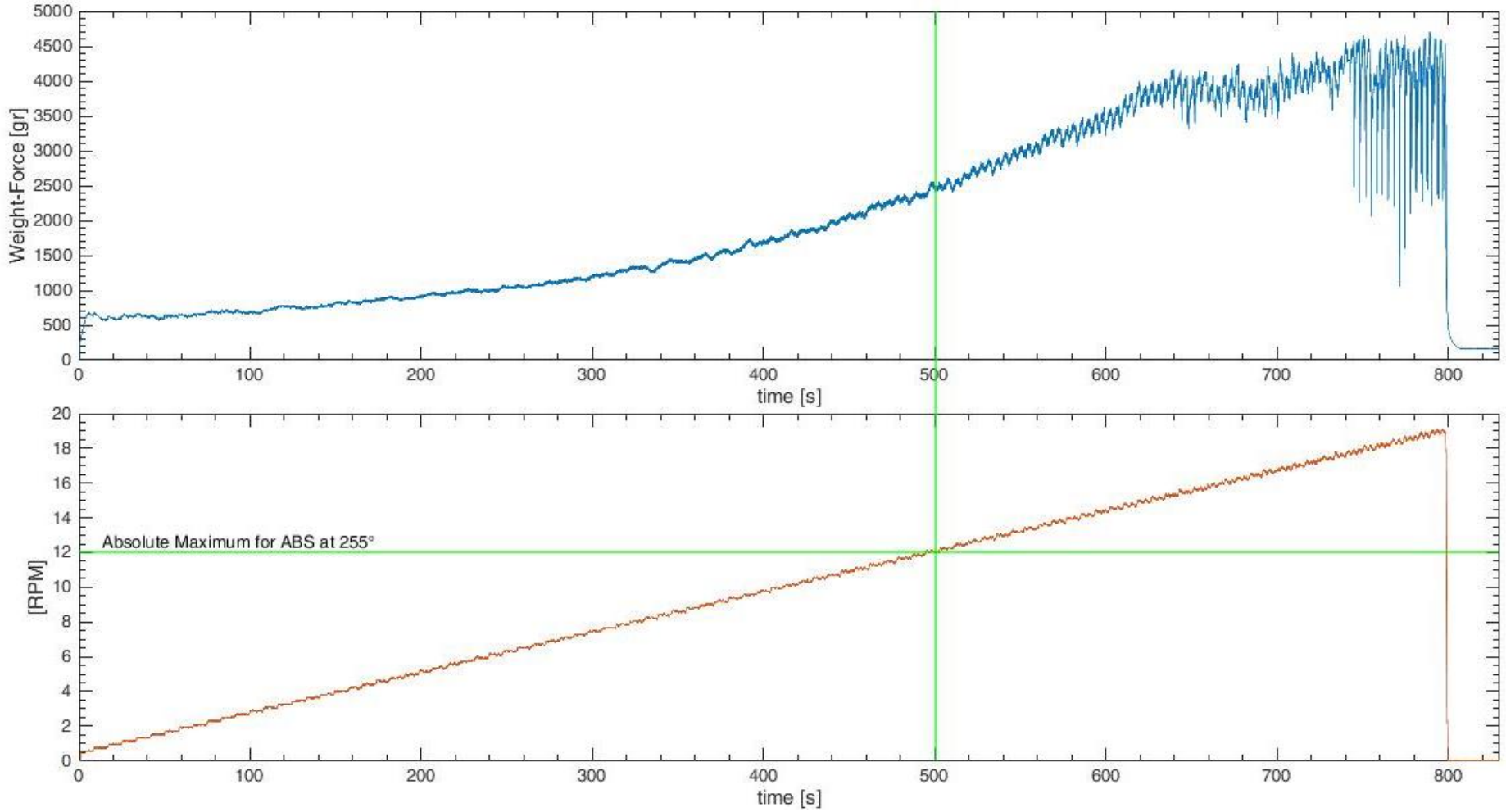
Furthermore, it interesting to notice that within the working range, with PET the extruding force increases proportionally with the speed while for PLA and ABS it increases exponentially. Albeit this the working range can be considered almost linear.



Graph 4: Weigth-Force and Angular Speed over time for PLA at 215°



Graph 5: Weigth-Force and Angular Speed over time for PET at 240°



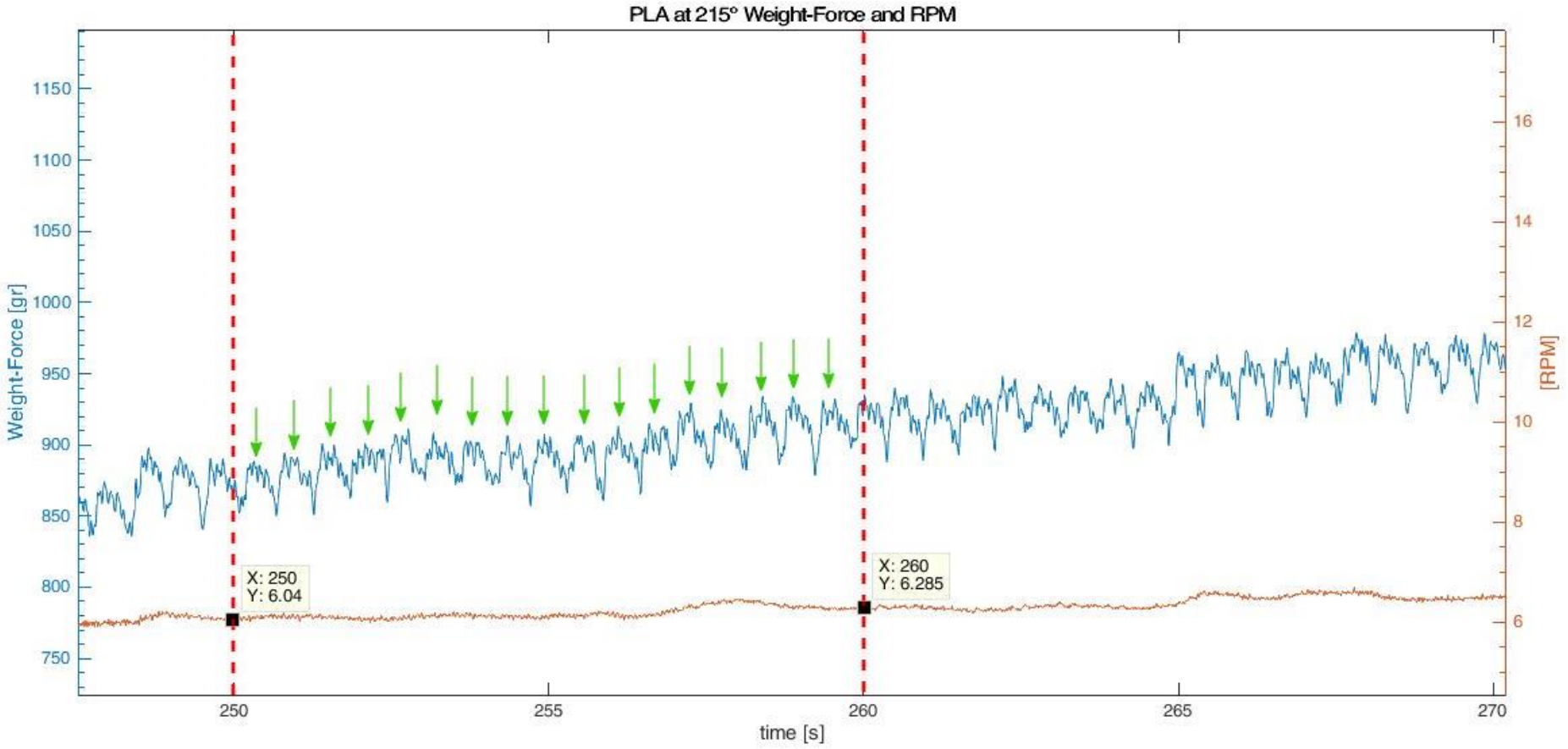
Graph 6: Weigth-Force and Angular Speed over time for ABS at 255°

Giving a closer look to the weight-force graphs, which can be assumed to be proportional to the pressure inside the nozzle, it is possible to notice that fluctuations and oscillations occur. In particular, it is possible to notice 2 different oscillations which differ in terms of frequency. Actually, their frequency in not constant in time, in fact it keeps increasing during the entire test. However, observing the angular speed of the extruder, it is possible to notice that the increase of the frequency is proportional to the increase in velocity, and strictly related on the motion of the gears of the extruder.

The first oscillation is the fastest between the two and has a peek to peek variation between 50 and 70 grams. As it is possible to notice in Graph 7, the pressure oscillates 17 times in a time span which corresponds to a full rotation of the extruder gears. This phenomenon is caused by the coupling between the spur gear on the motor shaft and the driven spur gear which rotates on bearings on the side of the motor. The coupling of the 2 gears is provided by 17 teeth for each gear. While the two gears are rotating, every time a new pair of teeth is engaged, the two gears separate from each other causing a grip loss and consequently a decrease in the pressure in the nozzle. This mechanism is designed in this way in order to provide grip on the filament, from both sides and not only by the gear installed on the motor shaft, such as other extruders, which instead of a driven gear, use an idler bearing.

A proof of what is exposed above can be obtained with the data in Graph 7: the two red lines highlight a time span of 10 sec; the two datatips show that the angular speed at the beginning and at the end of the time span is respectively 6.04 and 6.285 rpm; it is considered the mean speed of 6.1625 rpm and from it the time span for a complete revolution which is 9.736 sec is computed, from that it is possible to see that more that one rotation occurs in the span between the red lines, and it is possible to count 17 complete oscillations and a fraction of the weight-force signal.

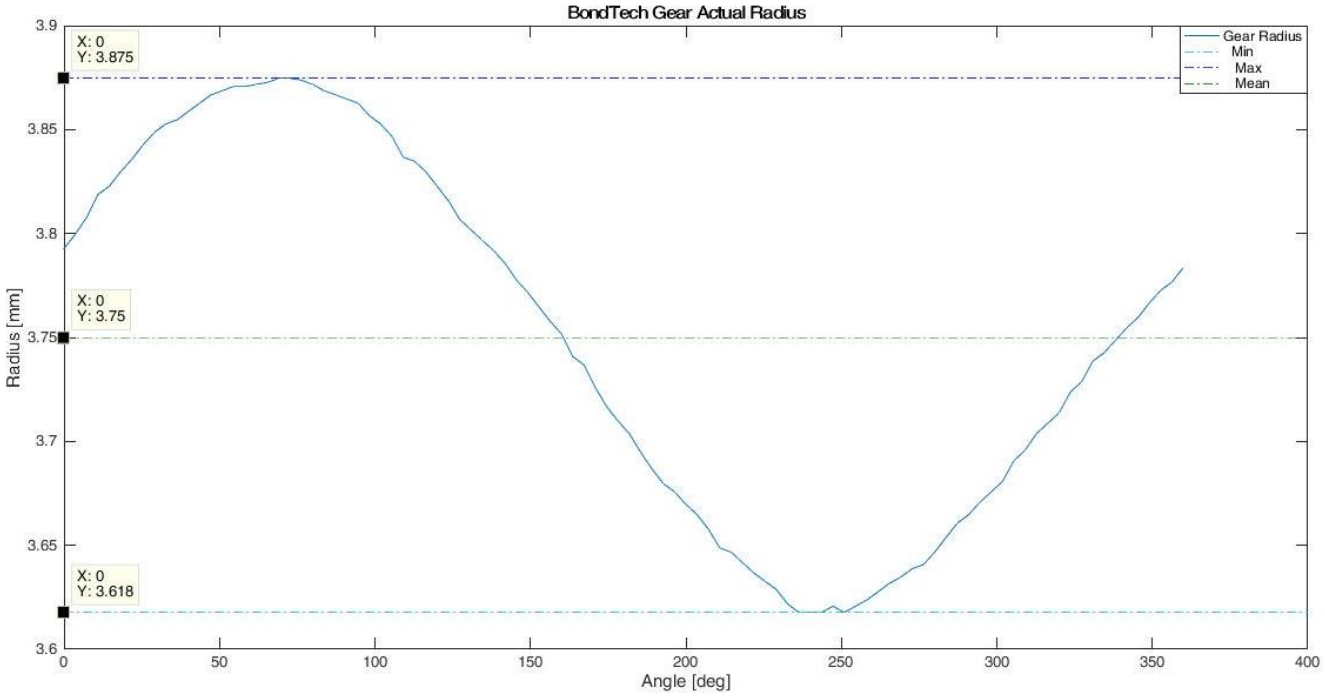
Figure 24: Extruder's Gears: the gear on the right can be screwed on the motor shaft and transmits motion with the spur gear teeth at its end to the gear on the left, the driven gear. Filament is gripped thanks to the rounded grooves on both gears



Graph 7: weight-force oscillation related to extruder spur gear coupling

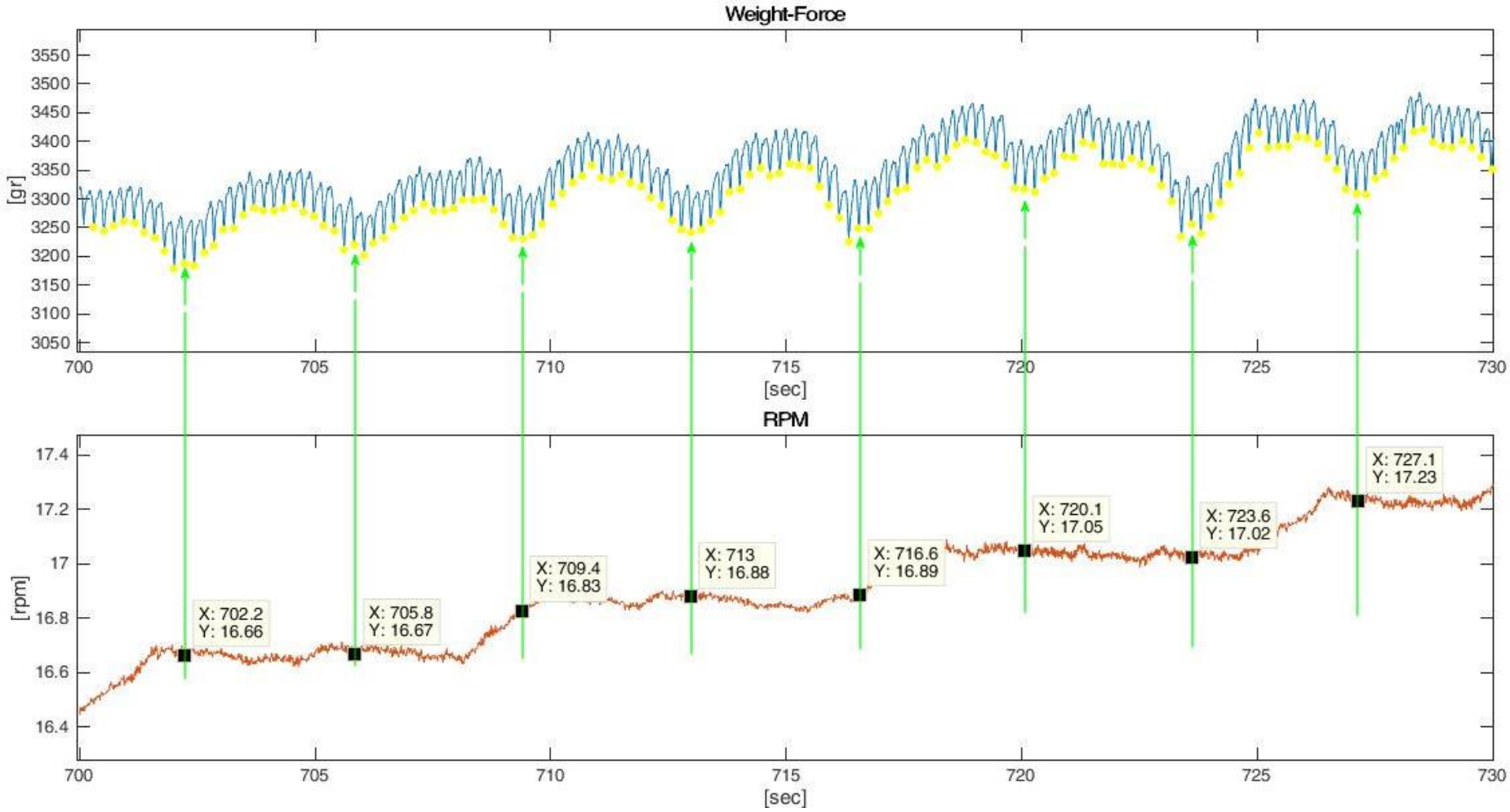
The second oscillation which can be distinguished, is again related with the mechanics of the extruder and has an oscillation with a frequency still variable in time but dependent to the angular velocity. As shown in Graph 9, the pressure oscillates once, in a time span corresponding to a full rotation of the motor shaft. This is not only recognizable by considering the realized rotation, but it is also possible to see that, super-imposed on this slower oscillation, 17 faster oscillations.

The main cause of this phenomenon which has a greater effect at high extruding speed, is that the gear on the extruder shaft is not perfectly concentric. This is caused by the method through which it is fixed to the shaft: a small screw fixes the gear to the 5mm diameter D shaped shaft. The gear hole inner diameter is slightly bigger (~ +0.1 mm) and thus resulting in an inconstant gear radius which not only decreases the grip but also insists on the transformation of the angular velocity in linear velocity. In Graph 8 the result of the measurements acquired with a digital indicator (accuracy ± 0.0005 mm) of the gear radius seen by the filament are reported. Data are acquired on a smooth part of the gear and not directly on the grooves which in any case are equally affected. Data shows a sinusoidal variation of the radius over the full rotation with a peek to peek variation of 0.2mm which is perfectly consistent with geometrical consideration.



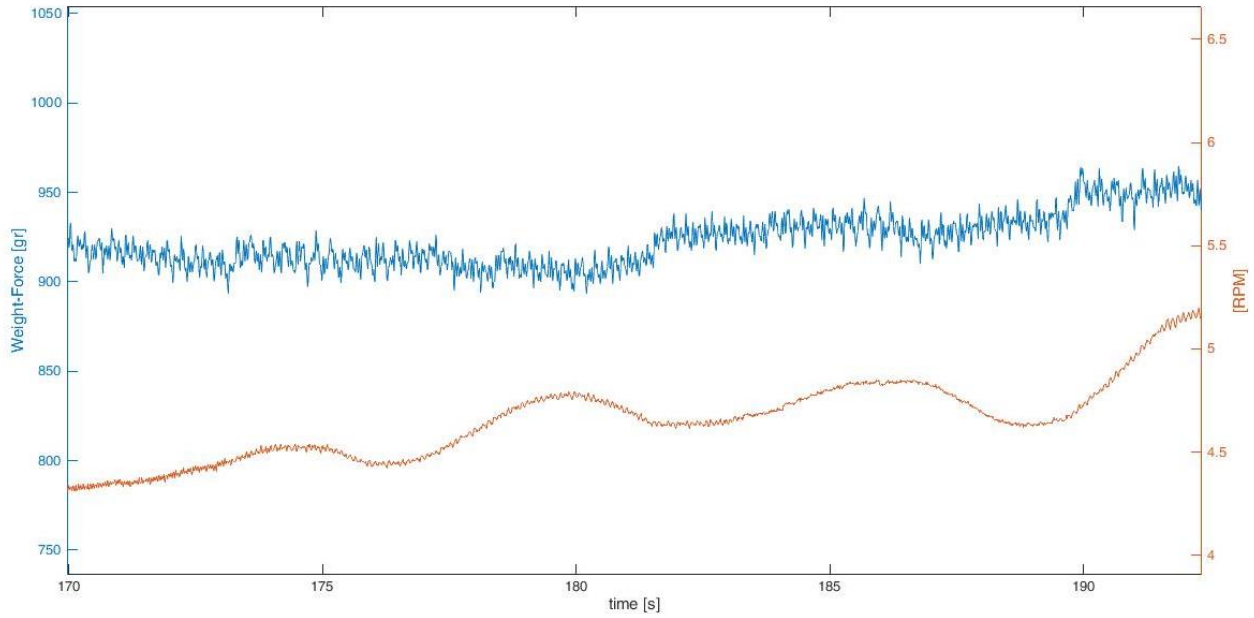
Graph 8: Oscillation of the gear radius seen by the filament

To proof what is explained above, through the data of Graph 9, it is possible to verify that the mean angular speed in the first interval is 16.665 rpm, which means that a full rotation is completed in 3.6 seconds. The two minima of the oscillation which delimits the interval, occurs at 702.2 and at 705.8 seconds respectively, and thus the elapsed time is exactly 3.6 sec.



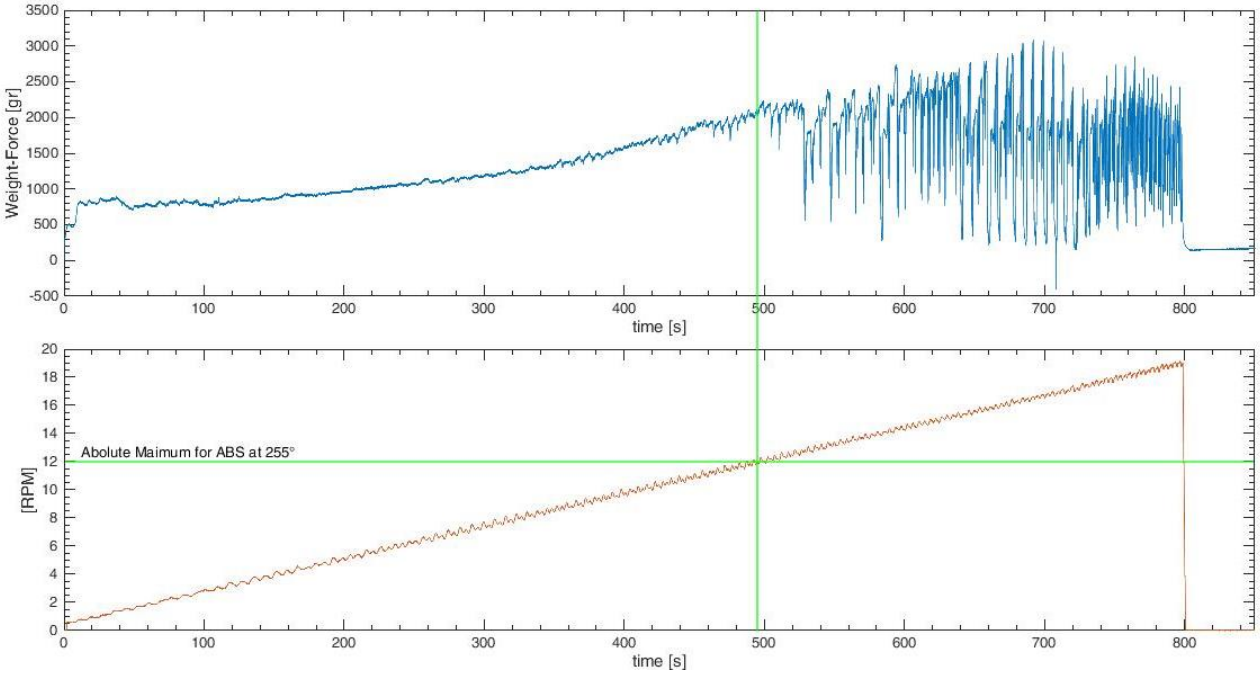
Graph 9: Oscillation due to not-concentric extruder gear, PET at 240°

Since the oscillation due to badly coupling spur gears, is easily solvable, a test is carried out by substituting the standard driven gear with a gear of the same type in which the teeth were removed on the lathe, from this moment on it will be referred as idler gear. The oscillation then disappears, as plotted in Graph 10, which shows the weight-force and the rpm related to ABS extruded at 255°.



Graph 10: Filament is not grabbed by two sides, since teeth on driven gear are removed; ABS extruded at 255°

The downside of removing the teeth is of course a grip loss, since the filament is not grabbed and pushed by both sides. This can be seen in Graph 11 in which ABS at 255° is extruded. In fact, by comparing it with Graph 6 it is possible to notice that the filament starts skipping sooner at around 12 rpm or 2.2 Kg-force, while in Graph 6, grip loss occurs at around 16 rpm or 4 Kg-force.

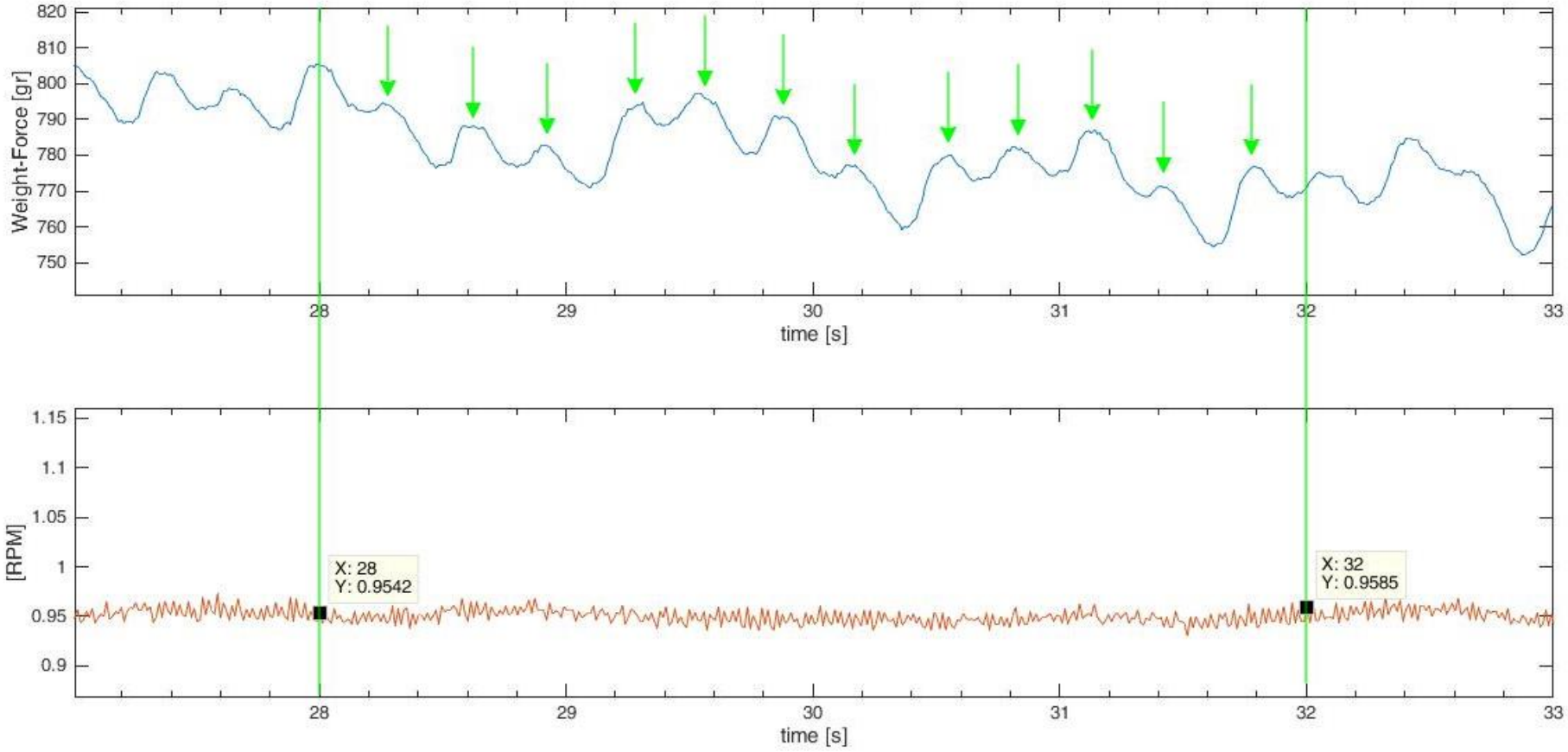


Graph 11: Weight-Force and Angular Speed over time for ABS at 255°. Teeth on driven gear are removed

The behavior in the previous test was obtained by fastening the idler gear against the motor gear with standard force, or in other words, in the same manner it is fastened in the standard setup. By over fastening the idler gear it is possible to reach nearly the same level of grip, to the detriment of an increase of a radial load on the motor shaft.

By carefully observing the data of the previous test, where the coupling between the gears is removed a further phenomenon, related to the nature of the stepper motor, can be discovered. By looking at Graph 12, it is possible to observe an oscillation with a peek to peek magnitude of around 15~20 grams. The important feature of this oscillation is its frequency, in fact, it matches the same stepping frequency of the motor. More precisely the oscillation in weight-force matches the frequency with which the motor, which has 200 step per revolution, rotates from one full step to the other.

A proof of what is said above, can be obtained with the data from Graph 12, the green lines mark a time span of 4 seconds in which the angular velocity of the stepper motor has a mean value of 0.95635 rpm. At that velocity, in that time span the rotor covers an angle of 22.9524° or in other terms the 6.37567 % of a full revolution. In a full revolution the rotor covers are 200 step and thus in 22.9524° 12.7513 steps are executed. As pointed by the green arrows, it is possible to see 12 full oscillations and a fraction of it at the edges of the interval.



Graph 12: ABS printed at 255°, oscillation with same frequency of the crossing of step position

CHAPTER 10

Design of the extruding system model

An extruder model is design in order to have a better insight of the system under investigation and to ease the process of tuning the PID controller which will be introduced in the following chapter. A model-based approach is used with the help of Matlab and Simulink, so the extruder model is, from the beginning, part of controller-plant-harness hierarchy. The first step is to write down the equation that describes the flow in $\left[{mm^3}/{s}\right]$ of melted polymer:

$$Flow = \frac{dVolume}{dt} = \frac{d}{dt}\left(\pi * {r^2}_{filament} * h\right) = \frac{d}{dt}\left(\pi * {r^2}_{filament} * \int V_{linear}\right) =$$

$$= \left(\pi * {r^2}_{filament} * r_{gear} * \dot{\theta}\right)$$

Where: $r_{filament}$ is the radius of the plastic filament in [mm]; h is the length in [mm] of the extruded filament in the time unit ; V_{linear} is the linear velocity with which the filament is extruded in [mm/s]; r_{gear} is the radius in [mm] of the gear which is gripping on the filament; $\dot{\theta}$ is the angular velocity of the motor in [rad/s]

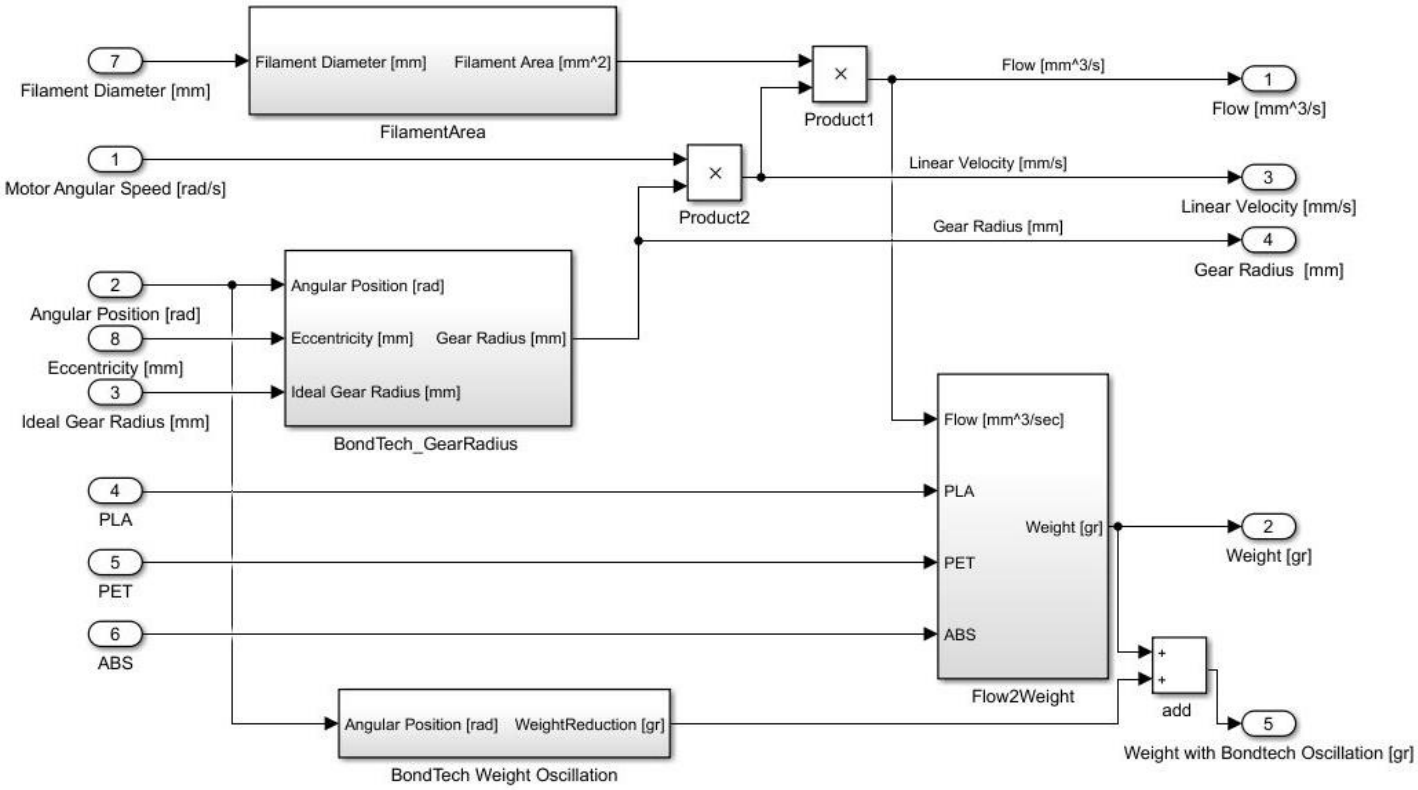


Figure 25: Extruder Model

The second step is to link the flow of material to the gram-force which is needed to extrude the filament.

$$F_{extr} = \beta * Flow$$

The coefficient β is hardly a constant value, as it possible to see from Graph 4 to 6, however from that data, it is possible to estimate a function which approximate the Speed/Force Characteristic shown in Graph 4 to 6. It is like having a continuous look-up table which relates the flow of the filament to the gram-force which it generates when extruded. This is done by approximating the curves in Graph 4 to 6 with a n-degree polynomial for PLA, ABS and PET. The estimate of the coefficients is done by using the Matlab function *polyfit.* Various attempts are made to choose the order of the polynomial for each polymer; so that, for each polymer the best fitting polynomial up to the 10^{th} degree is computed and then a comparison takes place. Finally, PLA and ABS can be approximated with a third-degree polynomial and PET with a first-degree polynomial.

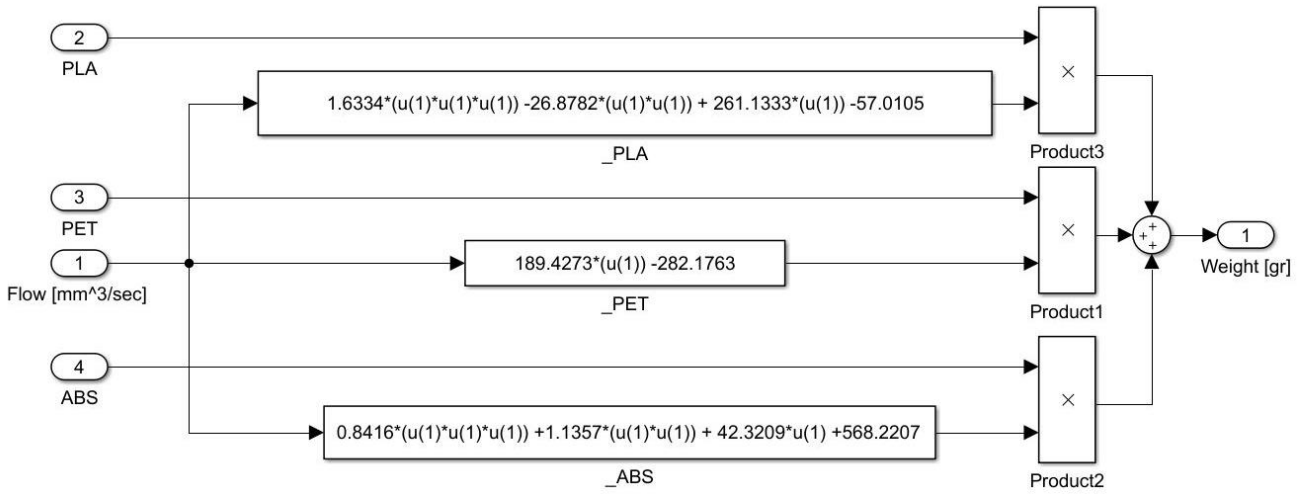


Figure 26: Content of subsystem Flow2Weight, use to convert the polymer flow into the weight-force

Finally, the mechanical non-idealities of the gears can be modelled. The oscillation due to the non-concentric gear on the motor shaft can be modelled easily exploiting geometrical considerations according to which the radius seen by the filament changes sinusoidally over a complete rotation of the gear. The mean value of the radius is its ideal value while the superimposed sinusoidal oscillation has, as amplitude, the eccentricity between the gear and the motor shaft.

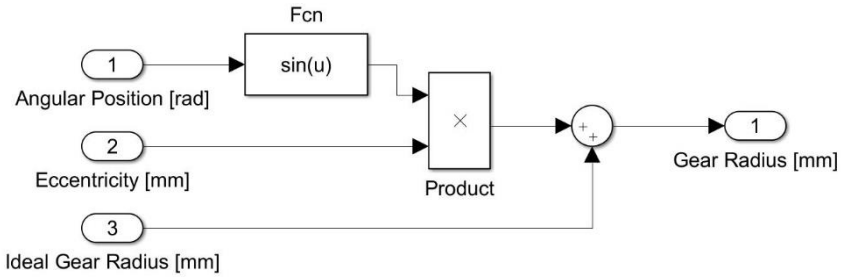


Figure 27: subsystem Content of BondTechGears_Radius subsystem, the non-ideal radius is obtained

A bit more difficult is to achieve a proper model for the oscillation of the badly coupling teeth of the spur gears. No direct formula is given, but, since the effect is that the weight-force signal oscillates 17 times per rotation with amplitude of 60-70 grams for any speed it is possible to recreate and impose, to the weight signal, an oscillation of the same kind. The signal manipulation is obtained as in Figure 28.

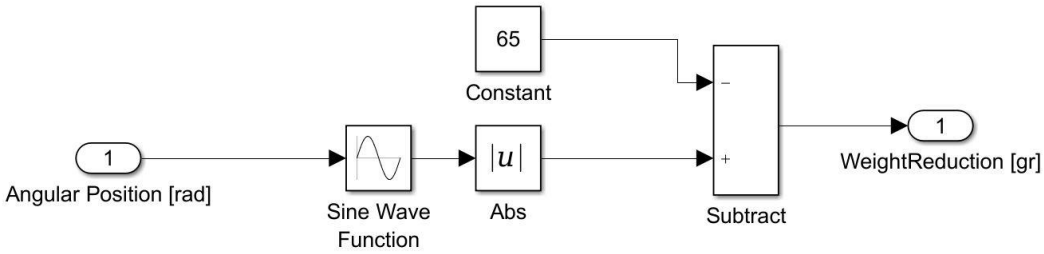


Figure 28: Content of BondTech Weight Oscillation

Next step is modelling the stepper motor and its driver. Various physical models are available on Simulink, which can introduce the torque and speed ripple which were visible in the data of chapter 9. However, the real issue relies on the stepper driver, since virtually none is available as an already made block in Simulink. More precisely some stepper drivers can be used in Simulink, however, their interface and the way they work don't make them compatible with this application. For the previous reasons, the stepper motor and its driver are chosen to be modelled in the most ideal way. So, it simply receives the stepping frequency and converts it into angular velocity, without introducing any "real-world" effects.

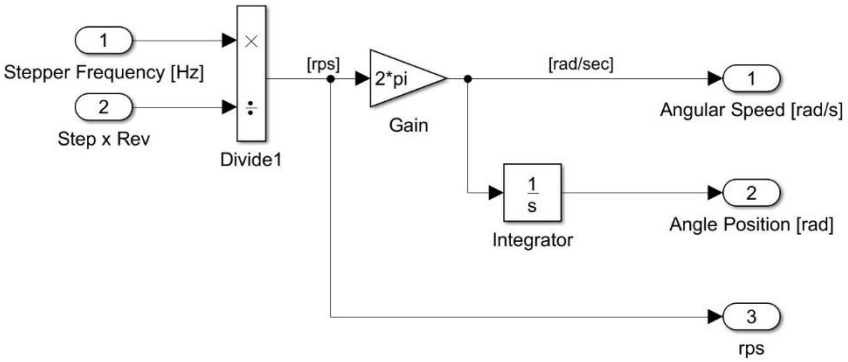
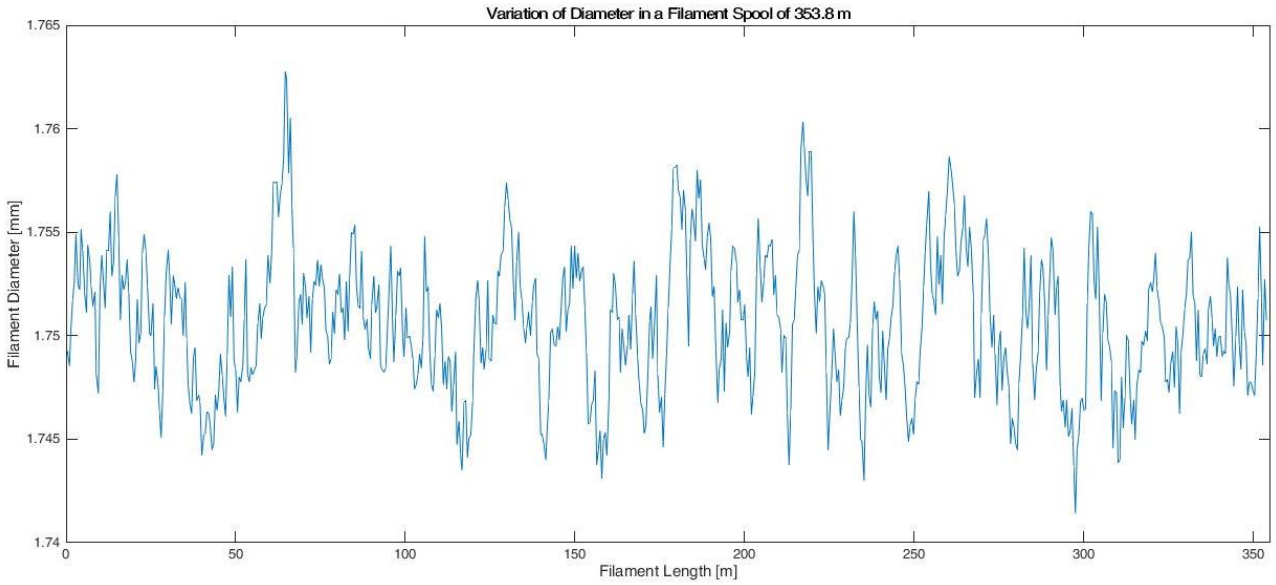


Figure 29: Ideal Stepper Motor Model contained in Stepper Motor subsystem

Finally, the filament is modelled. Actually, what is modelled is its diameter which as previously said has a tolerance of ±0.05 mm. Since the filament radius contributes to the flow with the power of two it is possible to prove that ±0.05 mm tolerance in the diameter can leads to a ~11% oscillation in the flow of material. However, the most important feature is the frequency with which it varies. If a good quality filament is used, which means that no lumps or necks occur, changes in the diameter, happens extremely slowly through out the entire length of the filament spool. Furthermore, the amount of filament extruded per segment is really low. Just to give an idea the longest segment or line which can be printer in MK3 corresponds to the diagonal of the heated bed which is 326 mm long. For printing this segment with standard setup only 13 mm of filament are required. Hence the diameter can be considered to be constant. As an example, in Graph 13 the diameter of an entire spool of filament is plotted. Data come directly from PrusaResearch extrusion lines.



Graph 13: Variation of filament diameter plotted along its length. Data from an entire 1Kg spool is plotted

Although the filament can be considered constant, for testing purpose to get closer to a worst-case condition, the filament diameter was modeled as changing along its length. This is achieved in the subsystem called “Real Filament Diameter”. The extruder model, the stepper motor model and the filament model are contained in the plant subsystem shown in figure 30. It is then possible to observe all the variable which are used as input and output.

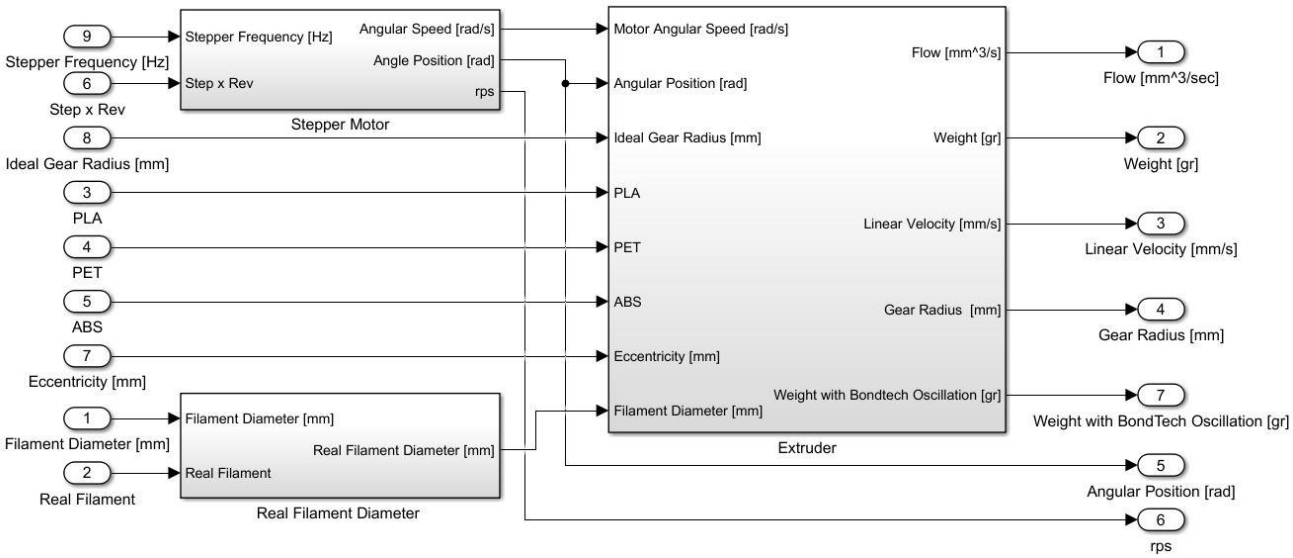


Figure 30: Plant Model with both Extruder and Stepper Motor Model

CHAPTER 11: Attempt to control plastic flow with PID controller

An attempt to control the filament flow is made since many aspects cannot be change or controlled. Any mechanical changes in the standard setup leads to a worsening of the performance in terms of grip, the oscillation in filament diameter cannot be controlled and better tolerances comes with a higher cost. Finally, some oscillations in the weight-force, which can be seen in the previous graphs, have an unknown origin or can be caused by turbulent behavior in the fluid. On top of that, since thousands of pieces are produced every month, developing a control which tune itself, no matter what the tolerances in the mechanics are, and control with a certain precision the flow, can be an interesting solution to explore.

DISCRETE PID CONTROLLER TUNED WITH MODEL-BASED APPROACH

Thus, the previous model is used as a Plant to tune a discrete PID controller with a Model-Based approach. The harness or the top layer of the Simulink project in which both the controller and the plant are, can be seen in the following figure.

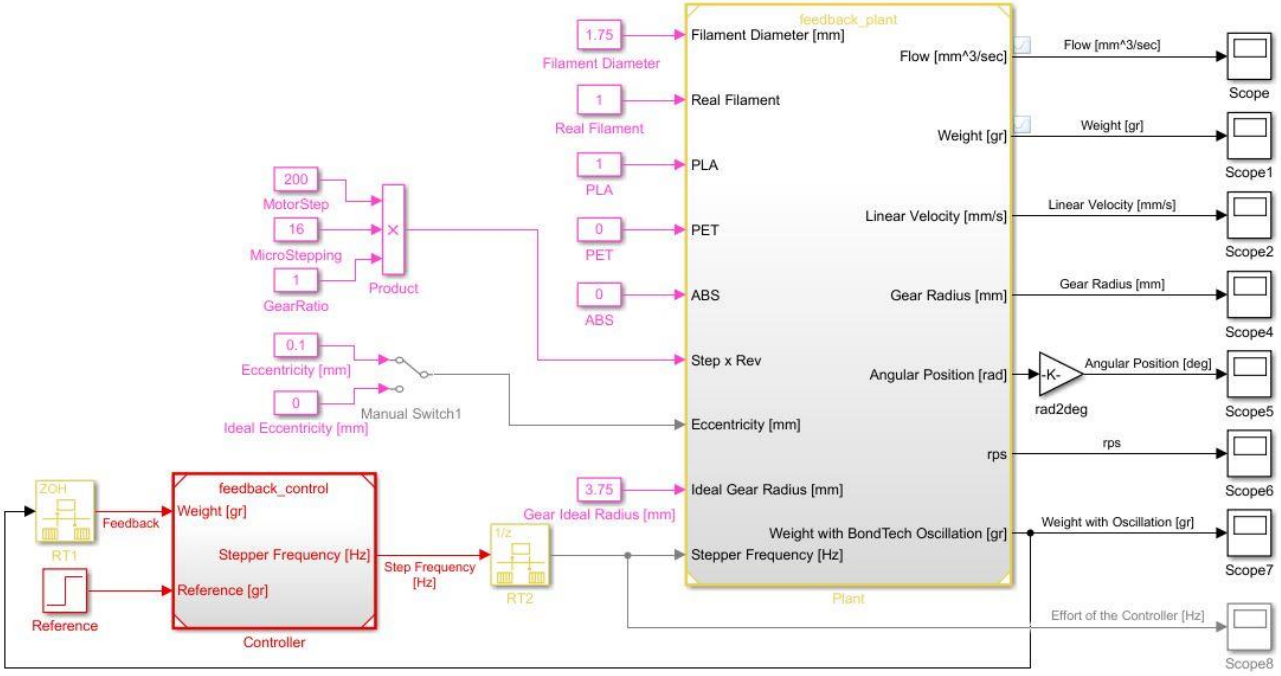


Figure 31: Harness layer, both the Controller and the Plant are here

In the harness layer the plant receives as input not only the desired speed computed by the controller, but also all the data which characterize the system. Furthermore, it is possible to choose which polymer is in use, by writing 1 or 0 in the constant value with the corresponding name. It is possible to select if gears are real and thus not concentric or ideal, as well as the filament diameter which can

be considered ideal or real. Lastly it is possible which signal the controller reads: the weight force with or without the oscillation due to the badly coupled spur gears.

While the plant runs in a continuous time with a variable step solver the controller runs in discrete time with a discrete solver, 80 times per second or every 12.5 msec. The reason of this choice can be found in the hardware in which it is deployed, that is the testing machine with which all the previous tests were performed. At the time this task was completed a better solution for sampling with a faster rate the output of the load cells, which could fit time, cost and complexity constraints was not found.

In the controller subsystem the error computation is performed, and it is given as input to the discrete PID controller which outputs the frequency with which the stepper motor is driven.

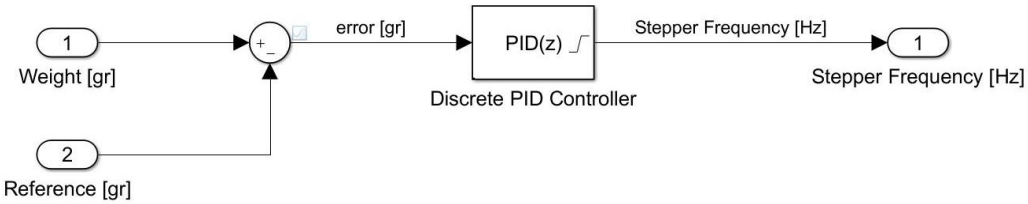


Figure 32: Content of the Controller subsystem

The PID is then tuned with the auto-tune function provided in Simulink, and the weight-force signal is plotted. The PID parameters found in Simulink are:

$$K_p = -0.0872688501785808$$

$$K_i = -5.4956983006667$$

$$K_d = -0.000346446065396624$$

The step response obtained with those parameters can be seen in the figure 33. It is possible to see an oscillation with max peek to peek of 80 grams. It must be noticed that the rising time is really quick because in the model no delays are introduced since no estimate can be done.

TESTING THE PID ON THE REAL SYSTEM: SOFTWARE SOLUTIONS AND RESULTS

To test the quality of the model in Simulink, a new piece of firmware is developed and deployed on an ATmega2560 micro-controller on the board which usually is used to run the printer. The firmware implements a PID controller which collects samples from the load cells every 12.5 msec, filters them to remove possible spike noise, computes the error between the reference and the weight-force signal, computes the frequency with which the stepper motor must run according the PID coefficients, sends via serial port the collected sample and eventually generates the pulse train at the desired frequency to drive to stepper.

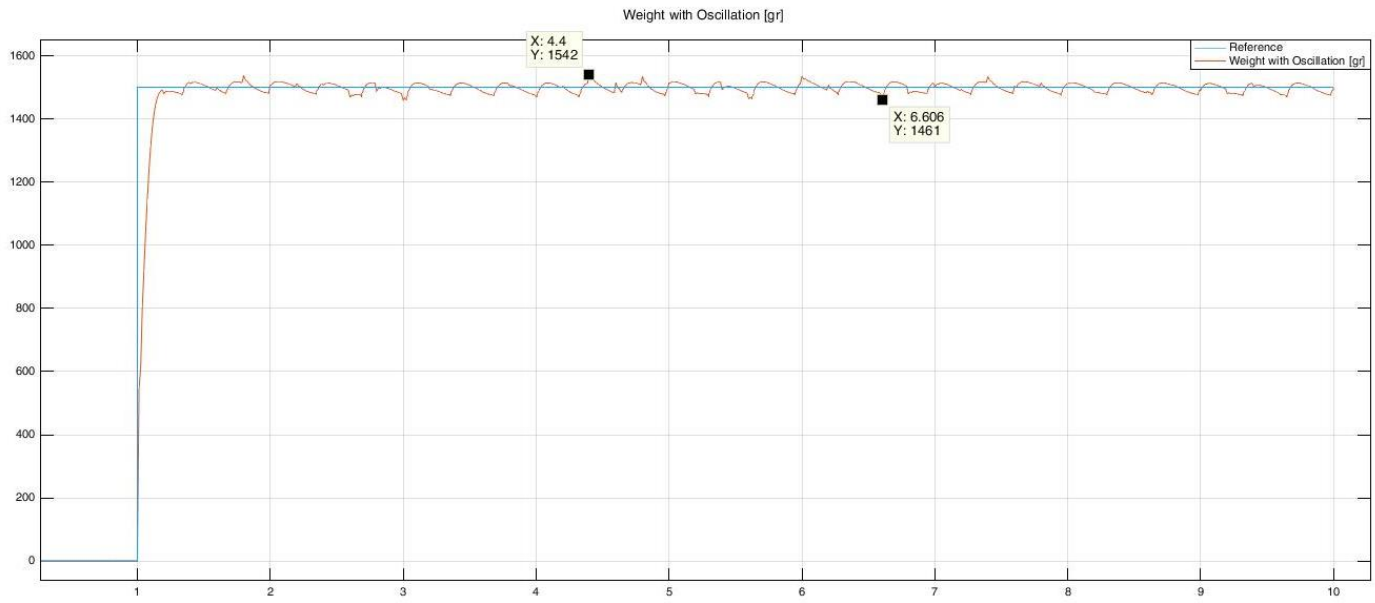


Figure 33: 1.5Kg step response obtained with the auto-tuned parameter from Simulink

The PID runs for 20 seconds and undergoes a step response. After that it stops and waits from the serial port to receive new PID coefficients. In this way it is possible to manually tune the coefficients, by receiving in real time the collected samples with which it possible to plot the step response. In fact, in tandem with the firmware on the uC, a script in Matlab is developed. It writes on the serial port the desired parameters for the PID and collects the samples with which the error, given to the PID, is computed. When all the samples of a step response are received in Matlab, a plot is produced allowing to have the necessary information to analyse and modify the parameters accordingly.

The parameters obtained in Simulink are then tested, and the step response is reported in figure 34. Pointless to say that an overshoot of 200 grams (40% of the reference), and an oscillation in steady state with a peek to peek of about 100 grams (20% of the reference) occur. Moreover, the timing of the dynamic of the signal is 10 times slower, than the one modelled in Simulink.

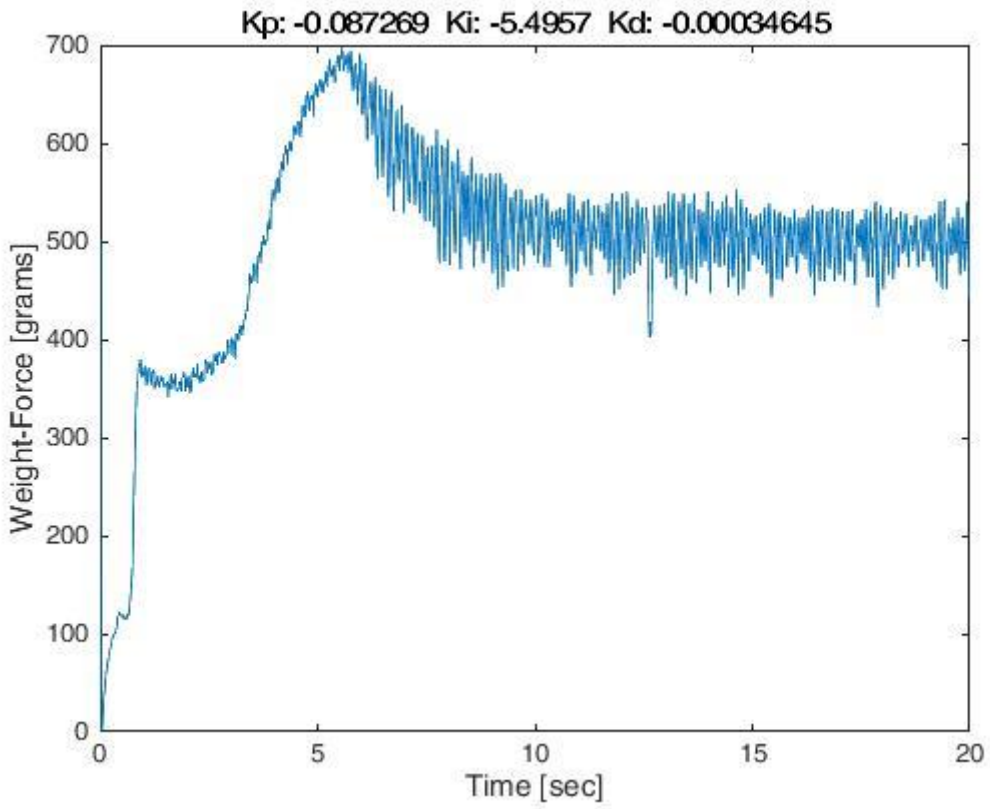


Figure 34: Step response of the real Extrusion System, obtained with PID parameters from Model Based Approach. Reference Signal is 500 grams

MANUAL TUNING OF THE PID

Considering the unsatisfactory results, from the parameters obtained with the model-based approach, it is necessary to tune the controller manually, starting from the already known parameters. After several trials the following coefficients are produced, and the related step response is plotted in figure 35.

$$K_p = -2.05$$

$$K_i = -0.9$$

$$K_d = -0.05$$

The step response obtained with the manually tuned PID, has a much smaller overshoot: the maximum peek occurs after 5.7 sec from application of the reference signal and reaches 520 grams (4% of the reference). At that instant, it is already possible to consider the response in steady state. The weight-force signal in steady state, oscillate around the reference of ±20 grams which can be considered a success, taking into account all the accepted compromises.

With the tuned PID, an attempt to prints some objects is made. The X and Y axes are driver by Marlin while the E axis is controlled by the previous firmware running no stop at constant speed. However,

no better results can be observed since the accelerations needed to move the extruder corrupts the control action provided by the PID. The information contained in the samples collected from the load cells, not only contains the information about the deformation of load cells, proportional to the pressure in the hotend, but also a disturb proportional to the acceleration on the X axis and to the mass hanged to the load cell. A different kind of sensor is needed to provide the feedback.

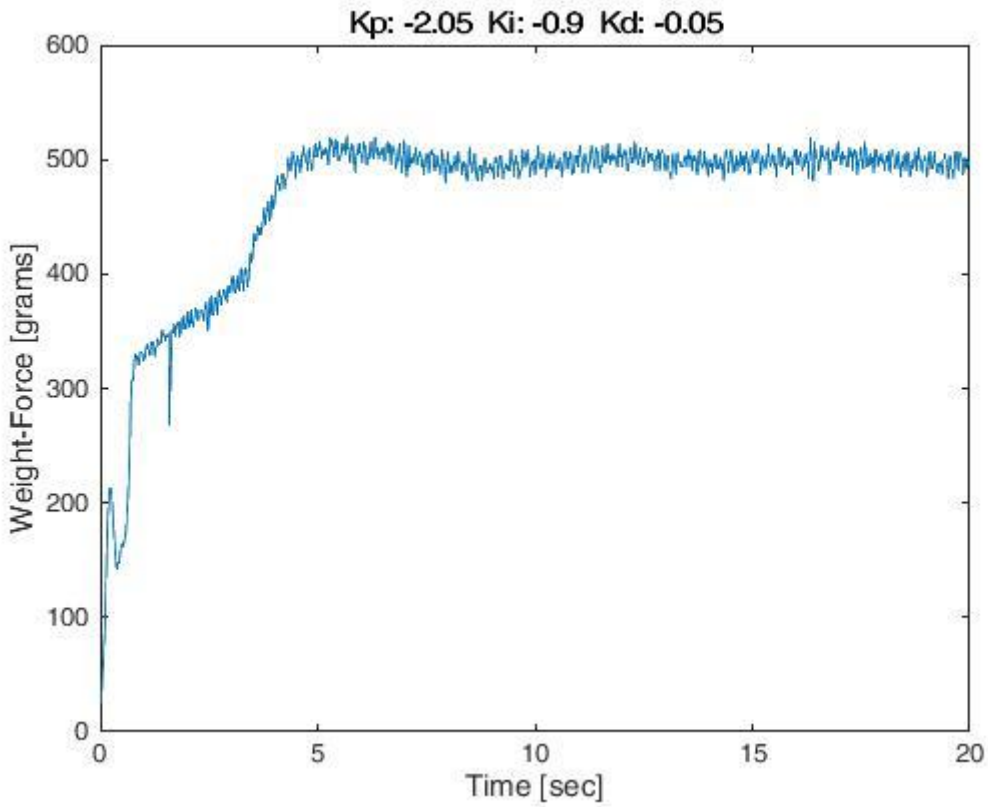


Figure 35: Step response of the Extrusion system, obtained with manually tuned PID parameters. Reference signal is 500 grams

CHAPTER 12: Proposal of better sensors to provide feedback

The previous printing tests point to the need for a different sensor to collect data useful to provide the feedback. Not only a faster sampling rate is required but also a strategy to minimize the disturbance introduced by the acceleration of the print head. Furthermore, load cells are too bulky for this application, so the investigation on possible sensors has to direct towards small and integrable solutions. Moreover, cheap solution should be preferred so that this control can be adopted not only in professional printers.

RESISTIVE STRAIN GAUGE ON PCB

The first sensor proposal is a resistive strain gauge on a PCB. The extruder can hang from a multiple-layers PCB on which a Wheatstone bridge can be realized. The resistors of the bridge can be made of copper traces conveniently placed so that, the deformation of the PCB induced by the filament extrusion, can results in changes of the resistivity of the copper traces and thus a signal proportional to the force used to deform the PCB.

This solution can be really cheap, and easy to scale but present some criticalities. The first issue is to find a good balance between the need to deform the PCB in order to generate the signal and the need of reducing the deformation to keep the tip of the nozzle at constant and controlled Z height. A possible solution can consist in lifting the extruder, along the Z axis, according to the deformation sensed by the sensor, compensating in real-time the vertical motion of the hotend. In any case an accelerometer should be employed to cancel the acceleration along the Z and X axes. Finally, the frequency response of the traces could be to slow making it incompatible with this application.

FORCE SENSORS ON PCB

An alternative solution can rely on of-the-shelf integrated circuit such as Honeywell FSS020WN. This force sensor can provide consistent readings up to 20N with a minimal deflection of 30 µm. A similar concept to the previous solution can be implemented. The PCB which houses the force sensors which must be 2 at least, for balance and symmetrical reason, has a hole in its middle. Around the hole the force sensors are placed. The hotend is inserted in the hole and can lean on the force sensors tip thanks to an extension ring which blocks the hotend in place, ensuring the contact. Whenever the filament is extruded the force sensors can collect sample, allowing to close the loop. In this solution the PCB is not expected to deform since measurement is accomplished by the sensors, and thus it can be supported from below making the set up sturdier. Furthermore, there is no need to compensate for the vertical motion of the hotend caused by the deformation of the sensor since at full scale the maximum vertical displacement of the sensors would be more that compatible with the application.

With this solution an accelerometer would be necessary to filter the disturbance due to the accelerations on the X and Z axis, unless a smart software filtering is implemented.

With this solution it is possible to realize a small PCB, industrial graded and slightly affected by EMI, with reliable and long-life components. The only and major drawback is the cost of the sensors: the retail price is around 80$ per unit and decrease up to 21.25$ per sensor for big volumes. This price leads to a cost of 42.5$ only for the sensors only which forces this solution out of MK3 market. Moreover, after a consult with PrusaResearch Chinese suppliers, it seems that no cheaper Chinese clone comparable to FSS020WN is available on the market.

PIEZOELECTRIC RING ON PCB

A further way to sense the force required to extrude filament is like the previous one but with the employment of a Piezoelectric ring. A PCB with the necessary electronics can host on its surface a piezo ring. The hole in the ring, which can be easily customized in size and shape, can match the hole in the PCB, through which the hotend can be inserted and then blocked on the top of the ring. In this way, whenever the filament pushes downwards, the corresponding force can be detected by the piezo.

The main advantages in adopting this solution is that piezo rings are largely available on the market and can be easily and cheaply imported from China. The PCB with on its top the piezo ring can be easily scalable and can represent a long-lasting solution since piezo transducer are considered reliable. Piezo transducer can provide a fast frequency response since the deformation needed to generate the signal is very small. Furthermore, it is a compact solution which can be easily integrated in the current design of the extruder. The main disadvantage consists in the fact that a piezoelectric crystal is more suited for dynamic applications, in fact it is basically a capacitor which varies its change proportionally to the applied force, but as every capacitor it discharges and thus the electric charge decays with time due to the internal impedance of the sensor and the input impedance of the signal conditioning circuits. Thus, a strategy to take that effect into account can be necessary in order to avoid drifts in the measurements and thus in the plastic flow. Finally, as all the previous proposal an accelerometer is required to cancel out the disturbance due to the accelerations.

PIEZOELECTRIC RING IN THE HOTEND

The last solution which could be worthy to investigate is placing a piezo ceramic ring, capable of standing high temperature, directly attached to the nozzle like in figure 34. In order to solve any issue regarding sealings at high temperature and pressure, induction heating is used in tandem with a new component which embeds the functionality of both the nozzle and the heatbreak.

The most interesting feature of this concept is that the disturbance due to accelerations are mostly avoided due to its shape. In fact, the disturbance due to the accelerations read by the piezo or the load cell can be written as follow: $F_{disturb} = a * M$, where M is the mass hanged to the sensor and a is the acceleration on the Z and Y axis which acts on the mass M. Hence, by reducing the masses hanged on the sensor it is possible to reduce proportionally the disturbance force.

The main downsize is that the temperature reached by the hotend would be limited by the capability of the piezo ring to withstand it. It is possible to find on the market ceramic piezo which can go up to 350°, albeit a decrease in performance occurs. Furthermore, induction heating for this application is not an available technology yet.

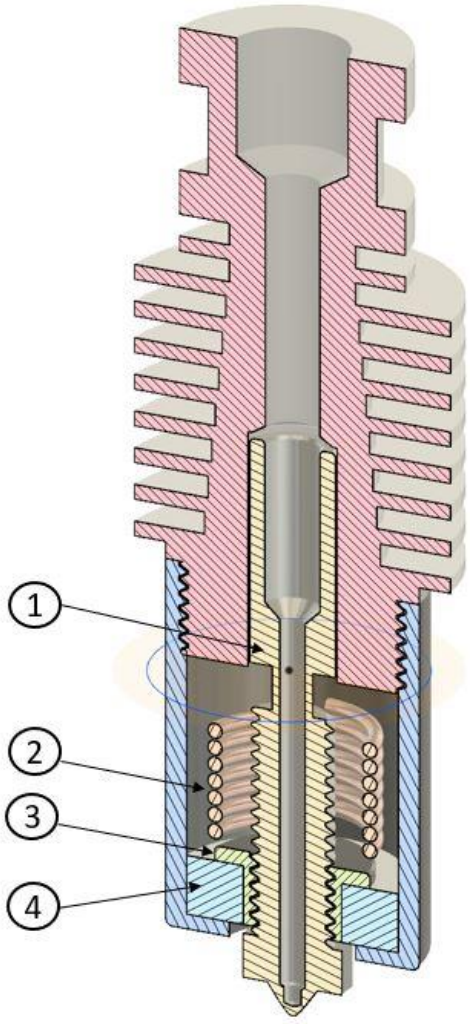


Figure 36: Hotend concept, courtesy of Mr. Ondrej Veverka:
1. Nozzle and heatbreak in one piece
2. Coil for Induction Heating
3. Metal threaded insert
4. Piezo Ceramic Ring

CHAPTER 13: Weight Reduction through Stepper Dimensioning

Reducing the moving masses in the printer is crucial to limit vibrations and to increase print quality and speed. In the X axis the entire extruder set up weighs around 550 gr and the major source of mass is the stepper motor (300 gr). Hence special care is taken to investigate a better solution for stepper motor.

A BIT OF THEORY ON STEPPER MOTOR

When a stepper is driven with full steps at low speed, a ripple in torque and thus in velocity occurs. Under low speed conditions which are verified for the extruder stepper motor, the dynamic torque can be deduced from the static torque and the torque ripple can be explained. The static torque of a stepper motor is shown in figure 37. When the motor is in step position, which means that the proper set of rotor and stator teeth face each other, no torque is produced. As soon as the rotor undergoes a displacement a torque is generated which forces it to move in the opposite direction, back to the step position. It basically behaves such as a pendulum. The maximum torque which can be produced to counteract the displacement is the peak static torque also known as the holding torque [1].

In steady state condition (no load torque), the rotor is moving from one step position to the following whenever a different phase is excited. The following conditions must be assumed: at t = 0, the rotor is at step position with velocity V and the corresponding phase is switched on. After few instants the rotor travels away from the step position and thus a resistive torque is produced, which decelerates the rotor until the next phase is switched on and thus positive torque is produced and accelerates the rotor. When the rotor reaches the next step position it has non-zero velocity and thus it rotates beyond it. So, a resistive torque is produced and hence the rotor decelerates until the following phase is switched on. Therefore, this process generates an oscillation with the same frequency with which the rotor traverses each step position. Furthermore, the maximum dynamic torque or pull-out torque, in this condition, can be deduced from the holding torque and can be from 60% to 100% of the holding torque depending on the excitation scheme [1].

The fact that motor is driven with micro-steps in no load condition leads to a frequency increase and a decrease in amplitude of the torque ripple, with respect to previous situation, being equal the velocity. In fact, new intermediate equilibrium step positions are created by increasing the micro-stepping, however, the holding torque for each phase decreases exponentially. This leads to the fact that, when a load torque is applied, a magnetic backlash or a slips results in the rotor position by displacing it from the desired position until sufficient torque is generated [2].

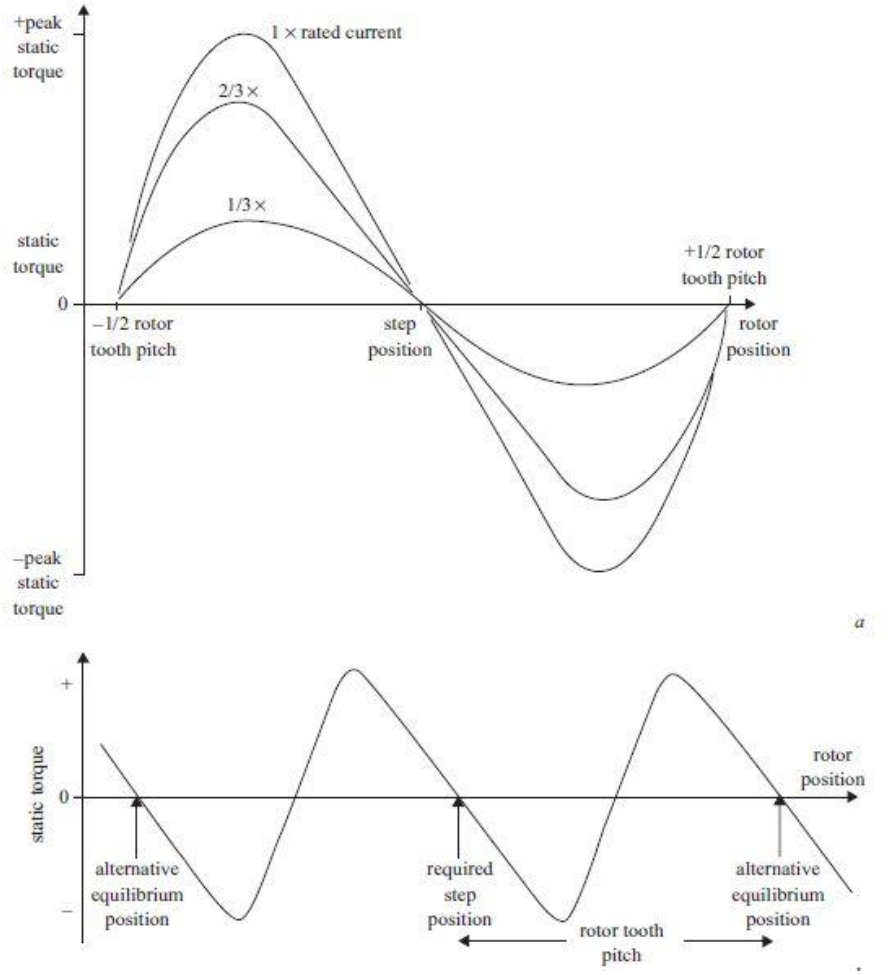


Figure 37: a. Static Torque vs Rotor Position at various Phase Currents;
b. Static Torque vs Rotor Position at Rated Phase Current;

It is then possible to define the incremental torque T_N produced every time N micro-steps are taken: $T_N = T_{HFS} * \sin((90 * N/\mu_{PFS}))$, where μ_{PFS} is the number of micro-steps per full step, N is the number of micro-step taken or executed and T_{HFS} is the holding torque at full step. Consequently, if the load torque is greater than the incremental torque of a micro-step, successive micro-steps must be realized until the accumulated torque exceeds the load torque and thus the rotor accelerates again [2].

ATTEMPT TO DIMENSIONING A LIGHTER STEPPER MOTOR

The stepper motor in the current setup has 200 step per revolution and is capable, at rated current, of a holding torque of 5 Kg*cm or ~500 mN*m. The stepper driver is set to 32 micro-steps per full steps which means a resolution of 6400 step/rev which results in a filament displacement of 0.00368 mm

per step. However, 32 micro-steps per full step, lead to a reduction in the holding torque of 95.09% with respect to the holding torque at full step and thus ~25 mN*m [2].

Taking into consideration the maximum force needed to extrude PLA, PET and ABS from Graph 4 to 6 it is possible to compute the maximum torque required for each filament with this setup:

Polymer	Max Extruding Weight-Force [kg]	Max Torque [mN*m]
PLA	3.1	110.6
PET	1.8	64.24
ABS	2.5	89.23

From the chart above it is understandable how the stepper has trouble to move accurately and thus the slip is introduced. It is also possible to estimate the slip in the worst-case scenario while extruding PLA at maximum speed. In fact, by using the equation for the incremental torque, with 500 mN*m of holding torque per full step, 32 micro-steps per full step and 110 mN*m as desired torque a 4.5 micro-step slip can be obtained.

Considering the issues above an attempt to dimensioning a different setup is made: a single stack or pancake stepper motor (162 gr) in tandem with a gear reduction is investigated if compatible with MK3. In order to keep the resolution as small as possible, a 400 steps motor is chosen and if a gear reduction of 3:1 is used (which can be found already on the market, called Titan Extruder by E3D) it is possible to reduce the micro-stepping from 32 up to 8. By doing so the resolution seen by the filament is 9600 step/rev or considering the different gear radius (3.875mm), every step moves the filament of 0.00254 mm. However, choosing a smaller motor of course means to have a reduced torque, in fact the maximum holding torque available is 1.295 Kg*cm or 126.9 mN*m, but since the micro-stepping is lower than in the current setup on MK3, the holding torque for each micro step is only reduced of 80.49% with respect to the holding torque at full step and thus 24.76 mN*m. But thanks to the gear ratio, the load torque seen by the motor is divided by 3, which means that in the worst-case scenario the required torque for PLA is 37.87 mN*m and thus the slip is reduced. Furthermore, the effects on angular speed of non-ideal torque generation, which can be seen with the previous setup (Graph 12), will result diminished by the inverse of the gear-ratio.

This of course, has a major drawback on the angular velocities which are increased by 3 times but, since the extruding velocity are small (max 16.37 RPM), the reduction on torque is negligible. A further negative effect which is introduced in this setup, is represented by the backlash in the gears. Even though the backlash in the gears can get worse with gear wearing, it is possible to estimate it and take it into consideration.

In conclusion by adopting the described setup, the resolution is increased by 50% while the holding torque on the motor shaft stays almost the same. Thanks to the gear ratio, however, the load seen by the motor is a third as well as the effect on angular speed of non-ideal torque generation. On top of that, the new set up, using the Titan Extruder, weights 450 grams instead of 550 gr which means a reduction of 19.2% of the initial weight.

CHAPTER 14: Conclusions and further advancement

CONCLUSIONS

In conclusion, in this work, it is assessed that extrusion inconsistency is caused by different causes. Extrusion inconsistency not only occurs in MK3 but affects the vast majority of 3D printers in MK3's market segment. The common denominator between every printer is the usage of stepper motors as actuators for all the axis. It is then reasonable to say, considering results from Graph 12, that stepper motor on the extruder has an impact on extrusion inconsistency. It is also possible to conclude that non-equal torque is produced by the 2 phases of the motor, caused by non-idealities such as geometrical imperfections. This consideration can explain the better results obtained by members of the MK3 Users community which swapped the standard stepper motor with a high-end model manufactured by Moons' Industries. This motor thanks to its proprietary technology, is sold for its smoothness and linearity. Furthermore, considering that oscillations caused by stepper motor are at high frequency, as shown in Graph 12, it is possible to explain why, on printers which adopts Bowden tube, the effects of extrusion inconsistency is so reduced. The filament in the bowden tube, acts as a spring and thus behaves like a low pass filter, absorbing oscillation introduced by the motor.

Moreover, Extrusion Inconsistency is also caused by Gear's non-idealities. Non-eccentric gear on the shaft of the motor and poor coupling of the spur gear teeth are proved to worse surface quality, albeit allowing higher grip on the filament.

On the other side it is tested that the frame of the printer, the tightening of the driven gear, the length of the melting zone, the step discretization and in general the firmware as well as the slicer have no role or have negligible effect in extrusion inconsistency.

Extrusion Inconsistency and, more in general, the entire control of the extrusion can benefit from a closed loop control. Standard mechanical components which provide great grip can be used since their drawback is compensated by the PID. Implementing sensors capable of providing the feedback can not only benefit the extrusion control, but also the introduction of new functionality such as clogging detection and stepper motor driver calibration. Proper calibration of stepper motor driver can help to reduce the mismatch in torque generation, due to stepper motor non-idealities. A sensor implementation similar to the proposals in chapter 12, can also be used to sense the printing bed height for implementing the mesh bed leveling algorithm, so that, the induction sensor of the current setup can be avoided.

Thanks to the dimensioning of a smaller stepper motor in tandem with a 3:1 gear ratio, it is possible to reduce the mass of the printhead of nearly 20%. Further 10 grams can be shaved off if the rotor of the stepper is customized by hollowing out the inner part of the stack. The weight reduction allows also to decrease the length of the bearings from 24mm to 17mm, with are slightly cheaper. Furthermore, the introduction of a gear ratio helps to reduce the effect of the motor non-idealities.

FURTHER ADVANCEMENT

Further resources should be focused in the integration of the PID control loop in the printer setup, or as a routine running with Marlin in the main micro-controller, or as an external and independent piece of firmware, running on an external board which can be sold as an add-on. Furthermore, different sensors should be investigated as proposed in Chapter 12 and conceived to be integrated in the geared extruder set-up or vice-versa. Moreover, the routine to calibrate the stepper motor driver to reduce inconstant torque generation should be developed as well as the needed changes in the bed levelling routines to use the weight sensor instead of the induction sensor.

REFERENCES:

[1] C. W. Hull, Apparatus for production of three-dimensional objects by stereolithography, US Patents: US4575330A, 1986.

[2] S. S. Crump, Apparatus and method for creating three-dimensional objects, US Patents: US5121329A, 1989.

[3] P. Acarnley, Stepping Motors: A guide to theory and practice, 4th edition, Institution of Engineering and Technology, 2002.

[4] MICROMO, «Stepper Motor Technical Note: Microstepping Myths and realities» 2015.

ACKNOWLEDGEMENTS

I want to thank all the people who supported me in my academic journey. My family above everybody else, has cheered in happy moments and helped me to stand back up in hard times. Without you I could achieve none of this. I want to address the most heart-felt gratitude to all my friends in Torino and Pavia ho supported me and loved me despite my distance.

A sincere thanks to Alex, and PrusaResearch hardware development team, you have welcomed me and let me be part of your team unreservedly. A huge thanks goes to my new friends, Ondřej and Michal, I hope new adventures are waiting for us.

Last but not least to Giulia, who chose me, no matter what.