S Goetz, D Hardman, F Iida (04.10.2023)

ENGINEERING TRIPOS PART IIA

Extension Feedback Control of 3D Printers

1. INTRODUCTION

3D Printers are mechatronic systems that utilise feedback control to perform additive fabrication. One of the most popular and economic technologies employs temperature-controlled extrusion nozzles which morph thermoplastics into a thin filament, the so-called Fused Deposition Modelling (FDM) method. For a high-precision and high-speed printing, it is necessary to design and operate feedback controller of 3D Printers, based on mechanical characteristics of 3D Printers and thermoplastic materials. This Engineering Extension Activity provides an opportunity to apply engineering methods to characterise 3D Printers as mechatronic systems, and to perform reverse-engineering of the feedback controller in the systems. Investigations should lead to improvement of printing performance and/or fabrication of unconventional structures such as hanging or web-like structures.

2. TIMETABLE & PLANNING

Before starting the lab activities, it is recommended to read through the instructions, to read some from the booklist, and to make a plan of lab activities. The timetable for the lab is as follows (S = Supervised sessions with demonstrators, U = unsupervised):

	Friday		**Mon**	**Tues**	**Wed**	**Thurs**	**Fri**		**Mon**	**Tues**	**Wed**
Morning	Study				Lab (S)	Lab(U)	Lab (S)		Lab(U)	Lab(U)	
Afternoon	Study				Lab (U)		Lab (U)				Presentation

First Friday till first Wednesday.
Students should go through all the reading materials in the Moodle site:
https://www.vle.cam.ac.uk/mod/folder/view.php?id=4663591

In addition, students without experience using FDM printers should spend some time exploring the technology and its capabilities via YouTube and the examples displayed in the Dyson Centre. No knowledge of how to use the printers is necessary at this stage: this is explained by these instructions and the demonstrators in the supervised sessions. Impressive starting points for research include 3D printed buildings, instruments, body parts, and the SpeedBoatRace (https://www.youtube.com/results?search_query=3DBenchy+SpeedBoatRace). To fit a reasonable number of investigations into our 2 hour laboratory sessions, we will be printing small models with a wide nozzle, so it is important to understand what the printers are capable of when given more time!

It is highly recommended to read the books listed below before the start of first lab session. It is also recommended to communicate with your lab partner (if there is one) on this day to discuss how you work together. It is up to each pair to decide on the partition of work between the two members of the pair. Your partner should be found here:
http://to.eng.cam.ac.uk/teaching/apps/cuedle/index.php?context=ExAFS3D

There is no physical lab session on the first Friday. On the first Wednesday, you are supposed to show up at EIETL at 11am where the demonstrators will give you instructions and supports. There will normally be another group in the Inglis EIETL Lab working before the first Wednesday, and you are asked not to disturb them at this stage.

Lab Sessions. You will have access to the laboratory and equipment from the first Wednesday until the following Tuesday. In particular on the timetabled laboratory days (Wed and Fri) demonstrators will be available in the Inglis EIETL Lab in the mornings to help.

- In the first supervised lab session students should follow the provided instructions to familiarise themselves with the operation of the rig. It is particularly important to understand how position control and temperature control of the extrusion nozzle are achieved in the software.
- From the second lab session, students should conduct the planned lab activities. There are some sample questions below, but it is encouraged to explore your own problems. Make sure that these investigations are summarised into a 10-minute presentation which includes engineering explanations and validation of the performance of 3D Printing. It is highly recommended that you use a personal laptop for these sessions, rather than relying on the EIETL machines.

Please ensure that any prints left running should finish by 16:30, so that the printers are not left operating unsupervised in the EIETL. Desks should be left tidy at the end of each day, with waste filament left in the boxes or bins provided.

Final Wednesday Presentation Session. You should put together a ten minute presentation on your findings. A member of staff present will act as session chairman. The objective is a clear well-organised ten minute presentation, to be followed by a five minute period for questions and answers. The specific time and location (or meeting link) will be emailed to you during the week.

3. HEALTH & SAFTEY

The 3D printer extruder heats up to well in excess of 100°C, and the heated bed up to around 100°C. Therefore, extreme care should be taken around these parts when heated as they could easily burn. The solvent used to clean the print bed can cause irritation to skin, so take care to avoid contact with the skin and wash hands if this occurs.

4. BOOK LIST

3D Printing

- Dyson Centre 3D printing overview: https://www.dysoncentre.eng.cam.ac.uk/3d-printing/3d-printing-default-page
- Ian Gibson, David Rosen, Brent Stucker: Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing, Springer (JW.1.2 & DM.50 in CUED library).
- G-code Wikipedia Page: https://reprap.org/wiki/G-code
- Chris Anderson, *Makers: The new industrial revolution*, Random House Business (BA.355).
- Hod Lipson and Melba Kurman, *Fabricated: The new world of 3D printing*, John Wiley & Sons (YH.2)
- Manufacturer's documentation: https://downloads.monoprice.com/files/manuals/33012_Manual_UK_180724.pdf

PID Control

- Good Overview of Ziegler-Nichols tuning (and its pitfalls): http://www.mstarlabs.com/control/znrule.html
- A good introduction to PID: https://en.wikipedia.org/wiki/PID_controller or http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID. See also CUED's IB P6: Information Engineering
- Myke King, *Process control: A practical approach*, Wiley-Blackwell (good introduction about some alternatives to the Ziegler Nichols).
- Antonio Visioli, *Practical PID Control*, Springer (a more in-depth look at PID control).
- History of automatic control: http://ieeecss.org/CSM/library/1996/june1996/02-HistoryofAutoCtrl.pdf

Introduction to Python Information

Learn python (or get the gist in 10 minutes):
https://www.stavros.io/tutorials/python/

Some longer, more detailed tutorials for python

https://www.tutorialspoint.com/python/index.htm
http://jwork.org/learn/doc/doku.php?id=python:start
http://hetland.org/writing/instant-python.html

FIRST LAB SESSION: Familiarisation Exercises (≈1.5 hours)

The 3D printer is controlled using 'G-code' which is a numerical control programming language which is used for many CAD applications. When printing a 3D model, the CAD file is converted into G-code using a 'slicer'. The G-code then controls the printer, setting the position, extrusion and heating. In this lab G-code is used to control the printer directly. This can be done through a graphical interface 'Pronterface' or, as will be covered later, directly sending G-code commands over the USB connection using Python scripts. *It is recommended to read the documentation in the book list on PID control and Ziegler Nichols tuning before beginning the instructions below.*

Calibrating and Initialising the 3D Printer

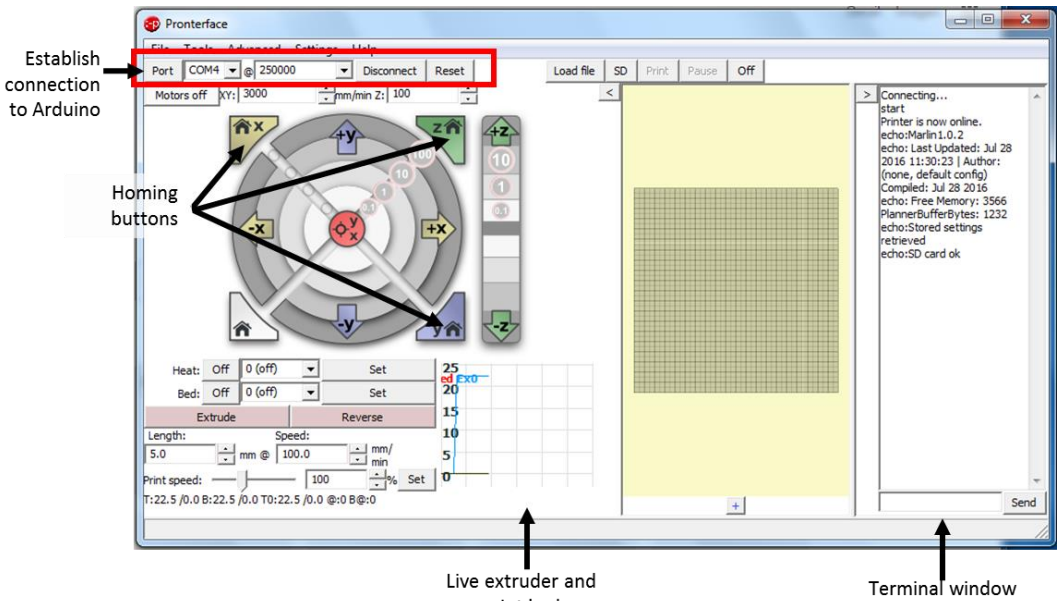


Figure 1: Pronterface interface showing the key parts

1. Load Pronterface (Printer Interface).
2. Connect the printer over USB and turn on the power to the printer. Do not remove the USB lead from the printer: we have found these microUSB ports to be weak, and a common failure point if continuously plugged and unplugged.
3. In the top left of Pronterface make sure the Baud rate[1] for the printer communication is set to 115200. The baud rate at which the printer works is actually 38400 (you will need this later for python scripting), but Pronterface is capable of detecting that as long as the selected baud rate is not too far from the actual printer's rate. Then select the correct USB port for the printer. This can be found under 'Devices and Printers' (Control Panel), or under the 'PORTS (COM)' in 'Device manager'. The port is in brackets next to this, e.g. (COM11). Now press 'connect'.
4. If the printer is connected properly the terminal on the right of Pronterface should display a connection message from the printer: 'Printer is now online'.
5. We can now calibrate the printer - check there is nothing obstructing the 3D printer's motion, especially the z-axis. To home the three axes of movement, press the individual homing buttons separately (pressing the x & y axis before the z) or by pressing the master home button on the bottom left of the motor interface.
6. Test the motion on the x, y & z axis using the motor control in Pronterface – **you should raise the z axis a little before moving x and y to prevent damage to the nozzle**. Note that the '+' directions are the directions away from the printers home, i.e. increasing distance from home. Be careful not to exceed the motors limits - if this does happen power down the printer using the red switch at the back. Once powered back on, you will need to reconnect to the printer through the Pronterface software and re-home the axes.

Using G-code

G-code commands can be sent directly to the 3D printer using the Pronterface terminal interface. A full list of G-code can be found at http://reprap.org/wiki/G-code, the firmware used is Marlin. Try entering the following commands into the terminal to practice using G-code.

Test commands:

'**M106 S255'** - this should turn the hot-end fan on. The 'M###' part specifies a command, whilst the 'S###' part here is a specific variable, other M-commands can have many more variables. In this case 'M106' tells the printer to set the fan, the 'S255' tells it to set the fans power to 255. The fans power can be set between 0 - 255, 255 being the maximum. Try changing the fan speed. (https://marlinfw.org/docs/gcode/M106.html)

'M301' - allows you to obtain the PID parameters of the extruder nozzle heater for the printer, this should give an output similar to 'ok p:24.31 i:2.18 d:67.71 c:1.00' these are the default PID parameters for the nozzle heater (the c:1.00 just indicates that it is the first nozzle heater). The variables for this command are as follows: P: Proportional Value I: Integral value D: Derivative value. To set the PID parameters you can send a command in the following format 'M301 P0 I0 D0'. (https://marlinfw.org/docs/gcode/M301.html)

[1] Baud Rate specifies how fast data is sent over a serial line, it is the symbol rate of transmission. Both the receiving and transmitting end must be using the same baud rate.

'M304' – this gives you the same PID information for the heater and has exactly the same variables. Send this command to the printer and compare the default parameters for the bed heater and nozzle heater; why are they so different?
(https://marlinfw.org/docs/gcode/M304.html)

Auto-Tune PID, and setting the PID Parameters

G-code 'M303 SX CX'', runs an autotune function for the PID controllers of the nozzle or bed heaters. It takes parameters 'S' the target temperature to tune around and 'C' the number of cycles to run for (https://marlinfw.org/docs/gcode/M303.html).
The autotuner raises the nozzle to the desired temperature the then cycles overshooting and then undershooting this target[2]. The autotune runs 3 cycles to establish 'bias' and 'd' variables, which set the maximum power to the heater. Further cycles are made, recoding the maximum/minimum temperatures reached during the overshoot/undershoot, and the time difference between these two points. The current to the heater is then adjusted iteratively allowing values for K_u (the 'ultimate' proportional gain) and T_u (the time period between max/min) to be estimated. Values of the proportional gain (K_p), Integral gain (K_i) and derivative gain (K_d) are then calculated from Ziegler-Nichols tuning rules. The function gives three different values for K_p, K_i & K_d based on three of the different Ziegler-Nichols Rules: the 'classic PID', 'Some Overshoot', and 'No Overshoot' rules.

1. Make sure the print nozzle is raised off the bed and run the PID autotune on the nozzle heater by sending 'M303 S100 C5' in the Pronterface console. After about 3 minutes the PID autotune should be done. Once the auto-tune is finished make a note of the K_p, K_i & K_d values for the different Ziegler-Nichols rules of thumb. *There should be three sets of values given, which are given in the order of: classic, some overshoot, no overshoot. In some cases only two sets will be returned, which is not an issue: continue with these.*

2. Download the PIDTemperatures.py script from Moodle site and open it locally using a Python IDE of your choice. Before running this first script ensure you have all of the correct Python libraries installed on your machine: time, pyserial, numpy, scipy, & matplotlib. If you are using the EIETL machines these should already be installed: see appendix for troubleshooting with Python in the EIETL.

   Run this Python file. This script produces temperature response data for the nozzle heater to observe the step response of the PID response. The script used for this opens the serial port, sets the temperatures of the nozzle to 100C, then records its temperature every 0.5 seconds. During this time, the three PID parameters (Kp, Ki, & Kd) can be changed using sliders, as can the target temperature. The actual and target temperatures are plotted in real time, and the data and graph are saved at the end of the script. Please look through the script to give yourself an idea of how it works for future reference.

[2] The C++ based firmware for the 3D printer can be found on GitHub: https://github.com/MarlinFirmware/Marlin this includes the code used to perform the autotuner: . This is for interest only.

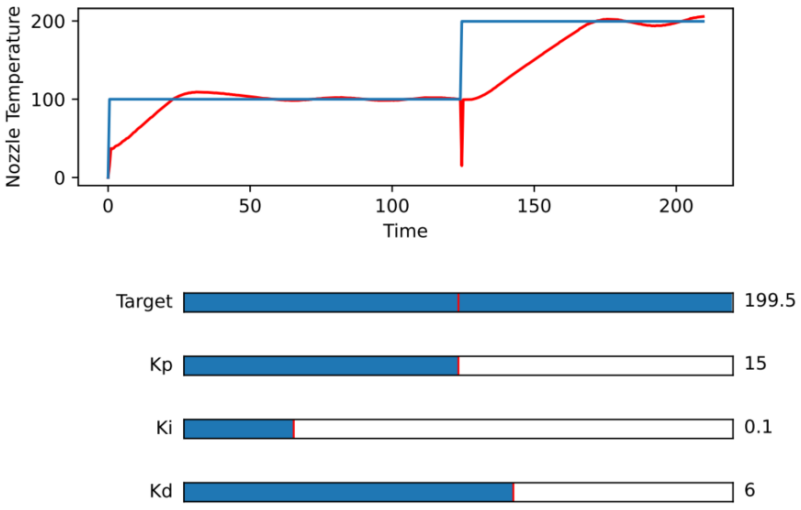


Figure 2: The graphical interface shown during the running of the Python script

3. Ensure that the nozzle is approximately at room temperature (at least below 40) before starting (check via printer's LCD screen). Disconnect the printer in Pronterface as only one program at a time can communicate over USB to the printer. At the beginning of the Python script, change the *com = "COM4"* line to reflect the port which was found above. Run the script (Run → Run Module, or F5). Set Kp, Ki, & Kd to suitable values obtained from previous experiments, and observe how the printer responds to changes in the target temperature.

Note that some of the printers display target temperatures below 100C without a decimal point e.g. 95.0C displays as 950C on the LCD screen. The actual temperature is not affected by this.

4. After 3 minutes you will be prompted to name the pdf and data files which are saved to the working directory. Data is stored using Numpy files, which can be reloaded into Python for further processing using the library's load function (https://numpy.org/doc/stable/reference/generated/numpy.load.html).

Investigation of PID Control

Investigate how to tune the PID parameters by observing the PID response, for example, consider the step response with only proportional gain only, no integral/derivative gain, etc.

Compare all of these responses with one another, noting their effects on target changes of varying sign and magnitude. Did you manage to improve on the auto-tuned parameters, and if so how did you do this? Did the step response change as expected? Which approach do you think gives the most desirable response for a 3D printer nozzle? Why do you think this rule's values are best, are they what you expect?

5. EXTRUDING FILAMENT (≈1 hour)

This section focuses on filament extrusion, and investigates the effects that changing the printer G-code parameters has on printing a single thread 'bridge' structure between two platforms. These initial investigations use PLA plastic.

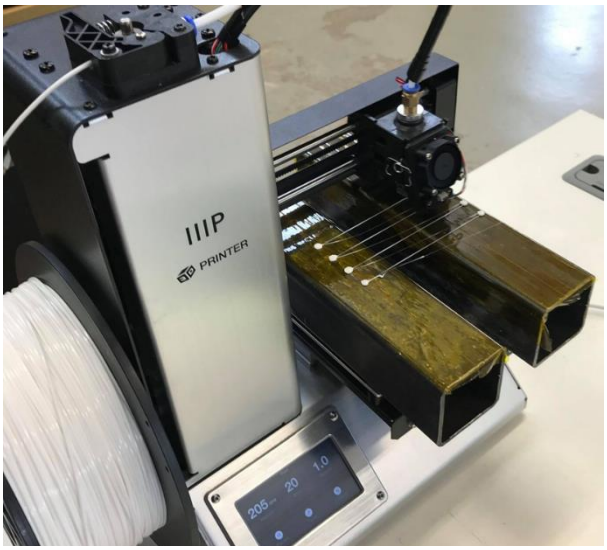


Figure 3: Experimental setup for bridge printing and filament extrusion

Moving the print head and extruding filament

The G-code command to move to a given location and control the extruder is 'G0' (or 'G1'; virtually identical) which can be followed by the following parameters:

- X, Y, & Z which specify a point for the printer to move to relative to the home position.
- E sets an amount of filament to extrude in mm and can be *absolute* (i.e. G1 E5 followed by G1 E15 will extrude 5 then 10mm of filament) or *relative* (this would extrude 5 then 15mm of filament). These modes are set using M82 and M83, respectively.
- F sets the 'feed rate' (in mm/min) for all 4 axes (X, Y, Z, E). If F is specified above the maximum for a given axis then the print head will move at the maximum for that axis, normally the Z-axis operates at its limit (120mm/min).
- The S parameter determines whether a flag is raised for an end-stop being hit, it is better to take care not to exceed the print area rather than use this parameter.

1. Open Pronterface, plug the printer in to the computer via the USB and connect as before. Clear anything off the print bed and home the axes
3. Send the G-code command 'G0 F4000 Z70' on the console in Pronterface, this will request the printer to move to (0, 0, 70) at a feed-rate of 4000mm/min, which is above the z-axis maximum, so it will actually move at its maximum.
4. Next send 'G0 X100' so that the printer head moves to (100, 0, 70), this command will use the earlier set feed-rate. Quickly experiment with different rates and positions. Now press disconnect in Pronterface.

Now download a second Python script from Moodle, 'bridge_print.py'. The first 30 or so lines are similar to the previous program, they open a connection to the printer using Python Serial, turn on the fans, and heat the nozzle. Using a mix of G92 commands which can be used to set a single axis and G0 commands, single threads are printed between the beams. This is repeated 6 times using a for loop, after each print user input is required to go on to the second line.

Clear everything from the bed and run this script, making sure the nozzle is heated sufficiently before proceeding to the printing. After calibrations, the script prompts you to place the bars on the bed and confirm when you are ready to proceed. Make sure that when you run this code that you do not press enter too early before a thread has finished. If you face adhesion difficulties, the provided Kapton tape & glue sticks can be used to help the filament stick to the beams. The last few lines send the printer back to (100, 0, 70) and turn off the nozzle and fans. Note each thread is increasingly thicker (why?).

Investigation

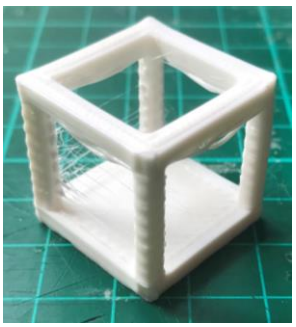
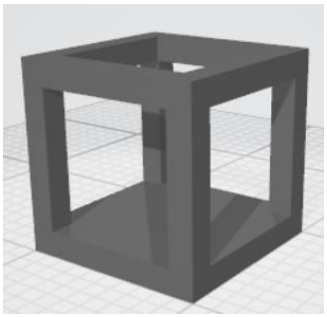
Now make a copy of this script (make sure to save with the file extension .py as otherwise will not run) and alter it to briefly investigate printing the threads with different settings. Examples of things you could investigate include:

- What effect does changing the feed-rate have for a given extrusion amount?
- Can you find a relationship between the feed-rate, extrusion amount and filament thickness, and what thread characteristics provide the greatest tensile strength?
- What is the maximum/minimum thread thickness you can achieve?
- What are the limiting factors?
- What about the deflection of the thread across the gap and how can this be varied?

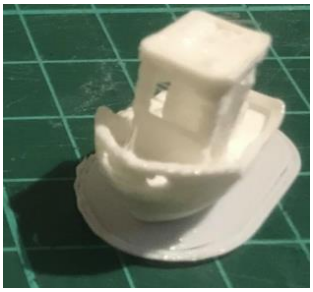
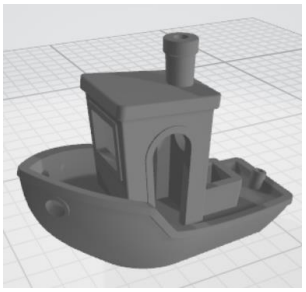
6. 3D PRINTING FROM GCODE FILES

3D objects are often printed from G-code files, which contain long lists of commands (such as G0 X70) to be executed by the machine. The slicedgcode folder provided contains pre-made gcode files for 5 3D objects, all of which can be printed in approximately 5 minutes with the printer's 0.8mm nozzle, and which test different aspects of the printing quality. The printed objects are shown below:

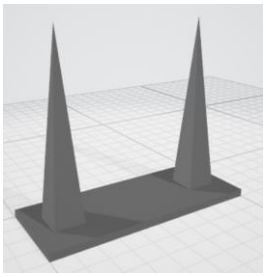
Hollow Cube

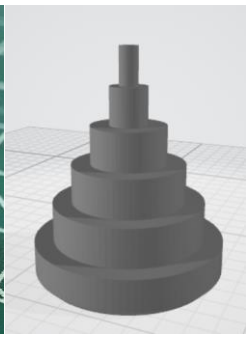
'Benchy' Boat

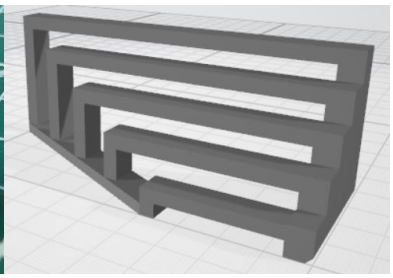
Stringing Pyramids

Dimensional Accuracy Pyramid

Bridging Ramp

To begin, we must ensure that the first layer of the print is deposited at the correct height, such that it adheres to the printer bed without the nozzle colliding with the surface. With the printer powered on, select 'Print' on the LCD screen, and select 'singlelayer.gcode' to begin printing a short 'purge line' along the edge of the bed, followed by a centred 30mmx30mm square for calibration. The printer's LCD screen will display the progress of the preheat. Once the nozzle and bed have reached sufficient temperatures, the z axis will calibrate its starting position relative to the bed using either a proximity sensor or a microswitch. This positioning is very important in ensuring that the first layer of the print sticks to the heated bed. Small adjustments to this position can be made – the method for doing so depends on the printer which you are using:

***WITH** proximity sensor (green, red & white wires)*

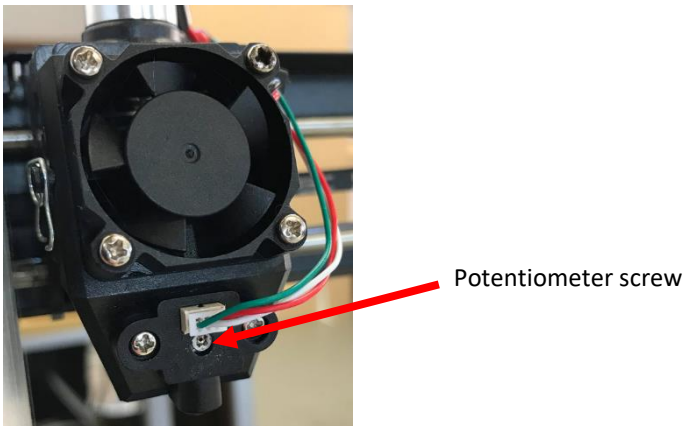


Figure 4: These 3 wires indicate the presence of a proximity sensor.

By selecting 'Tune' on the LCD screen during the print, small adjustments of the position can be made using the touchscreen. If the ideal height of the first layer falls out of this tuneable range, the variable potentiometer of the proximity sensor can be manually adjusted with a screwdriver: ask your demonstrator for help if this is the case.

***WITHOUT** proximity sensor (internal microswitch used for homing)*
The homed nozzle height cannot be changed, so we instead adjust the height of the bed until the first layer adheres. Firstly, check whether the bed is already at the correct height by trying to print the square. If adjustments are necessary, use the supplied hex key to manually raise/lower the screws in each of the 4 corners.

Adjust the printer until the first layer sticks to the bed: be very careful not to bring the 2 too close together, as this will scratch the bed. For more information, see: https://www.matterhackers.com/articles/3d-printing-essentials-first-layer.

The methods described above should be the first port of call for printing a successful first layer. If you continue to have difficulties, the supplied Kapton tape and glue sticks can be used to aid adhesion.

Finally, choose one of the 5 objects above. Ensure that you are connected to the printer via USB/Pronterface, and select Pronterface's 'Load file' icon. Navigate to the slicedgcode folder, and load the chosen G-code file. A small preview is displayed in the Pronterface window – click 'Print' to begin the preheat/homing steps. Once the print begins, it is possible to adjust the first layer height again to obtain good adherence. Check that the whole file can now be printed over approximately 5 minutes. Successful but low quality prints are not a problem here: the rest of this activity focusses on how the quality can be improved!

7. USING THE USB MICROSCOPES

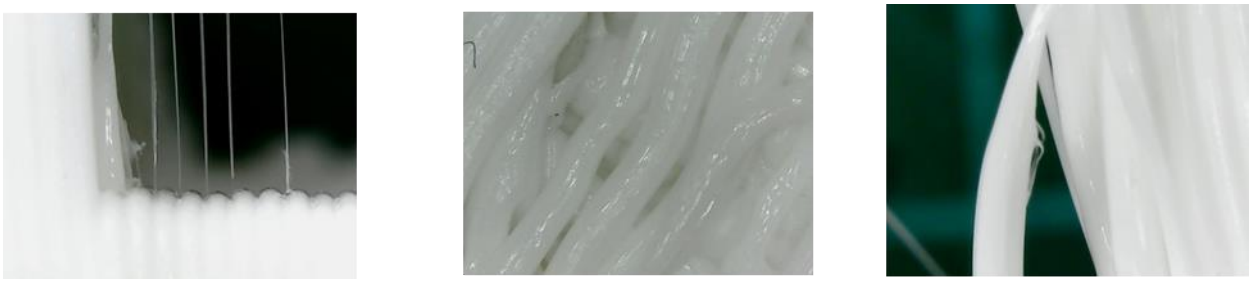


Figure 5: Sample images captured using a USB microscope.

Each kit comes with a USB microscope, which can be used to investigate the quality of your printed objects in more detail. The large dial on the body of the microscope is used to adjust the focus, whilst the small dial on the cable varies the brightness of the LEDs which it contains.

When plugged into a USB port, the microscopes are automatically recognised by most Windows computers, and are accessible through the Camera application. If you are having issues using the microscope in this way with a personal laptop, see: https://ww.jiusion.com/art/microscope. The EIETL machines use 'amcap' software to access the microscope.

View your print under the microscope, changing the focal length, amount of light, and area of the object being looked at. Ensure you have removed the protecting cap at the end of the microscope, otherwise you will face poor-quality images containing scratches. Notice particularly the visibility of each individual layer, the effect of any stringing in the print, and the consistency of the line width. Save any images of areas of interest which might provide the basis for later investigations.

8. CHANGING PRINT PARAMETERS

A huge number of parameters control the output quality of a printed object, including nozzle temperature, layer height, retraction length, and speed of printing. Often, these are tuned and set for each print using 'slicing' software, such as Cura.

Given the short amount of time available in the laboratory sessions, we focus on the effect of just 9 parameters on the print: the 3 extruder PID values, the nozzle temperature, the bed temperature, the speed of printing, the amount of extrusion, the retraction distance, and the fan speed. Using the provided python script, these can all be varied *during* the print, allowing us to examine how they affect the quality and output of the 5 test objects.

Open your Python IDE (there is no need to disconnect the printer from Pronterface: we will simply generate new G-code files which can be printed), and load ExampleGcodeGenerate.py. The comments which it contains provide an overview of the process: the central function is processgcode, which can be used to define all 9 of the control variables.

```
def processgcode(filestub, commands, kp=15.5, ki=0.13, kd=6.0, nozzletemp=210, bedtemp=55, speedfactor=1,
                 extrusionfactor=1, retraction=2.5, fanspeed=255):
```

This function takes up to 11 arguments. The first two must be defined for the function to run successfully:

- *filestub* is a string used to name the output gcode file, which is saved automatically to the outputgcode folder.
- *commands* is a list containing the data of the object to be printed. See ExampleGcodeGenerate.py for details of loading these from the objectdata folder.

The remaining 9 arguments are optional, and can be used to vary a number of parameters throughout the print. The values are interpolated over the print, based on the length of the list passed. For example, *nozzletemp = [190, 220]* will linearly increase the nozzle temperature from 190°C at the start of the print to 220°C at its completion. *nozzletemp = [190, 220, 190]* causes an increase to 220°C in the middle of the print, which returns to 190°C by the end. More complicated functions, such as sine waves, can be defined by passing longer lists.

- *kp*, *ki*, and *kd* represent the PID control parameters investigated earlier. Before beginning, it is wise to adjust the default values in ProcessCommands.py to those which you determined experimentally.

- *nozzletemp* and *bedtemp* define the *target* temperatures in degrees Celsius for the nozzle and bed respectively.

- *speedfactor* scales every speed of the prints introduced in section 7. For example, setting this as a constant of 0.5 will cause the entire print to run 2x slower.

- *extrusionfactor* scales the default extrusion lengths of the prints introduced in section 7. For example, setting this as a constant of 1.1 will cause 10% more material to be extruded.

- *retraction* is the value in mm of the hotend's retraction: the amount by which the filament is pulled back into the nozzle as the printhead moves between lines. It is often used to control the 'stringing' of prints – for more, see https://3dprinterly.com/best-retraction-length-speed-settings/.

- *fanspeed* is a value between 0 & 255 which controls the speed of the hotend's DC fan. By default, this is set to 255 throughout the print, which corresponds to the fan being fully on.

When processgcode is called, a new G-code file is saved into outputgcode, which can be printed directly from Pronterface using the steps in section 7. Open ExampleGcodeGenerate.py, select one of the 5 test models, and try varying the parameters within sensible values (limits for which are defined in ProcessCommands.py to prevent any mistakes from damaging the printer). Check that you can print the output file, and that you understand how to vary each of the 9 input variables.

9. OPEN ENDED INVESTIGATION

By now you should be able to start investigations on the topics of your interest. By choosing one or more of the parameters and test prints, explore how the physical outputs are affected by the parameters in a series of experiments of your choosing. Some examples might include:

- Is there a single nozzle temperature which universally produces the best prints, or should this be varied depending on the stage of the print/feature being printed?
- Define a measure of energy efficiency (using temperatures, speed, extrusion, and print time), and find a set of parameters to optimise this.
- How consistent do the dimensions of the printed objects remain whilst each of the 9 parameters is varied?
- Which factors limit the speed at which a print can be successfully produced?
- Can you identify a set of parameters which reliably control the quantity and thickness of 'stringing' in the print? Use the USB microscope provided to gain an insight into where and how this effect is occurring.

Alternatively, conduct further investigations into the tests of Sections 5 and 6, and produce a thorough report on your findings. Ideas might be:

- Derive a transfer function for the extruder heater. Conduct experiments to obtain impulse response, and explore how feedback controller can be designed or improved.
- Investigate the effect of altering the PID parameters on the quality of printing.
- Investigate PID tuning rules for this printer. Are the Ziegler-Nichols rules the most applicable for the printer? What are some alternative tuning rules (see 'Process Control : A Practical Approach by Myke King') and how do they compare? Can you make your own tuning rules?
- These instructions focused on PLA for printing, investigate the printing of threads/arches/etc from ABS instead, how do they compare?

Experienced users might wish to use CAD and a known slicing software to design prints for their own tests. Though this is not discouraged, you should ensure that these are quick prints to avoid wasting filament and waiting around during allocated sessions – the 5 suggestions above use a large nozzle diameter and layer height to achieve this. The provided G-code examples are generated with Cura, using:

- *Monoprice Select Mini v2*
- *0.8 mm nozzle*
- *0.35 mm layer height*
- *10% infill*

10. IMPORTANT NOTES, HINTS AND TIPS

- During the bridging tests, do not kill the printer mid-script *unless absolutely necessary* as this means you will have to remove the platforms from the bed to re-home the axis as opposed to simply setting the value of an axis with the G92 command. If some plastic gets all tangled up it is much easier to just remove it with pliers and continue the script to get the extruder back to position and start again.

- If the blobs of PLA for the 'bridges' aren't adhering to the platforms, properly cleaning them with some tissue and acetone should help. If this does not work check there are no rips in the Kapton tape (the yellow tape over the metal surface) that the plastic could be getting caught on, if there are then you should either re-tape (or get a demonstrator to re-tape) the block or use another if there are spares.
- Sometimes some plastic 'leaks' through the extruder as it is heating and this can snag on the blobs at the beginning, it helps to remove any pieces of plastic stuck to the nozzle with pliers before the printing begins. However, do not do this during the Z-axis calibration, since this might prematurely trigger the proximity sensor.
- Remember that the E parameter for the G0/G1 commands is absolute by default, i.e. if you send 'G0 E5' then 'G0 E10' (with no G92 command between them) the printer will extrude 5mm of filament, and then 5mm of filament. If after this you sent 'G0 E6' the printer would reverse 4mm of filament.

11. Appendix: USING THE EIETL MACHINES

If using the EIETL PCs, log into Windows using your departmental login. Pronterface is accessible from the desktop, and VS Code can be used as your IDE. Python 3.10 should already be installed – if not, this can be done directly through VS Code. If you have not used VS Code to run a Python Script before, you might need to point it to the directory in which Python is installed:

- Press Ctrl+Shift+P to bring up the command palette, and search for "Python Select Interpreter".
- Select "Enter Interpreter Path"
- Select "Find"
- Browse to the installed Python directory e.g. C:\Program Files\Python310\Python.exe