3D Printing with a Cobot Arm

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Bachelor's thesis

Mechanical Engineering and Production Technology

Spring 2022

Ales Trtnik

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Mechanical Engineering and Production Technology Abstract
Author Ales Trtnik Year 2022
Subject 3D Printing with a Cobot Arm
Supervisor Jaakko Vasko

The aim of this thesis was to create a proof of concept system for 3D printing with a robot. Robotic 3D printing is a new branch of additive manufacturing that is just becoming commercially viable. The benefits of robotic 3D printing are larger print sizes and extra degrees of freedom.

The software used for creating the 3D printing program was RoboDK and the robot used was Universal Robots UR5. A printing head was built out of off-the-shelf components and a control system was devised for it. Several test objects were successfully printed, which demonstrated the viability of the technology. Further research could focus on developing a dedicated control system for the printing extruder and sending the movement commands to the robot directly from the computer.

The main part of the thesis is a step-by-step walkthrough of the performed experiment. This is so that results can be reproduced and ultimately upgraded.

Keywords 3D printing, robotics, additive manufacturing
Pages 44 pages and appendices 4 pages

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Degree Program in Mechanical Engineering and Production Technology		Tiivistelmä
Tekijä	Ales Trtnik	Vuosi 2022
Työn nimi	3D-tulostus kobottilla	
Ohjaaja	Jaakko Vasko	

Tämän opinnäytetyön tavoite on tarkastella robotin käyttöä 3D-tulostamisessa käytännön kokeilun kautta ja arvioida sen toteuttamiskelpoisuutta. 3D-tulostus robotin avulla on uusi suuntaus materiaalia lisäävän valmistuksen saralla ja myös sen kaupallinen käyttö yleistyy. Sen etuihin lukeutuvat muun muassa mahdollisuus tulostaa suurempia esineitä sekä useammat vapausasteet tulostimen liikkeille.

Tulostuksessä käytettiin Universal Robotsin UR5-robottia ja 3D-tulostamiseen käytetty ohjelma luotiin RodoDK-ohjelmiston avulla. Tulostuspää koottiin hyllytavarasta ja sille kehitettiin ohjausjärjestelmä. Tällä kokoonpanolla onnistuttiin tulostamaan monta koekappaletta, mikä todistaa tutkitun konseptin toimivuuden. Erityisesti tulostuspään ohjausjärjestelmän sekä robottia ohjaavan ohjelmiston olisi jatkokehitys toisivat parannuksia nyt kehitettyyn kokoonpanoon.

Tämä opinnäytetyö keskittyy kuvaamaan tulostuskokoonpanon rakentamisen ja käytön vaiheittain, jotta se olisi toistettavissa sekä edelleen kehitettävissä.

Avainsanat 3D-tulostus, robotiikka, esinetulostus
Sivut 44 sivua ja liitteitä 4 sivua

Contents

1 Introduction 1

1.1 Experiment plan 1

2 Theoretical background 2

2.1 Standard 3D printing vs. Robotic 3D printing 2

2.2 Use cases 4

2.2.1 Printing oversized plastic products 4

2.2.2 3D printing casting moulds 4

2.2.3 Wire arc additive manufacturing 5

2.2.4 Concrete printing 5

2.3 Components and tools used in the experiment 6

2.3.1 Printing head 6

2.3.2 Printing bed 6

2.3.3 Filament 6

2.3.4 Robot simulation software 7

2.3.5 Slicing software 7

3 Preparing the experiment 8

3.1 Pen 8

3.2 Printing head 8

3.2.1 Mounting bracket 8

3.2.2 Components 9

3.2.3 Tool centre point 9

3.2.4 Printing head control system 11

3.2.5 Controlling the motor 11

3.2.6 Controlling the hotend 13

3.2.7 Controlling the fan 14

3.3 Printing Bed 14

3.3.1 Description 14

3.3.2 Assembly 15

3.3.3 Levelling 17

3.4 Filament Handling 19

3.4.1 Spool holder 19

3.4.2 Filament guiding.. 20

3.4.3 Loading the filament into the extruder... 20

3.5 Preparing the simulation in RoboDK... 21

3.5.1 Preparing the simulation setup... 21

3.5.2 Configuring Slic3r inside RoboDK... 24

3.6 URSim 26

4 Conducting the experiment...27

4.1 Creating the robot program with the simulation software 27

4.2 Creating the robot program with URSim 31

4.3 Uploading the program to the robot 34

4.4 Testing the program on the robot 34

4.5 3D printing 34

5 Findings...37

5.1 Pen experiment... 37

5.2 3D printing experiment.. 38

6 Conclusion 41

References...42

Appendices

Appendix 1 Experiment setup

Appendix 2 Arduino program for running the extruder motor

Appendix 3 Extrusion speed calculation

Appendix 4 GCode program for heating the hotend

1 INTRODUCTION

3D printing is a relatively young technology and it is in the stage of rapid diversification. Robotic 3D printing is one of the newest branches of the discipline.

The aim of this thesis is to test the feasibility of robotic 3D printing with the equipment and software already present in Häme University of Applied Sciences’ robotics and 3D printing labs. Most importantly, the Universal Robots UR5, because collaborative robotics is the labs’ main research focus. The goal is not to get perfect prints, but to manufacture minimum viable products that are going to justify further research.

The main part of this document is a step by step description of the experiment. It is written in a way that enables an operator with basic knowledge of robotics and 3D printing to replicate the results of the experiment.

1.1 Experiment plan

The experiment has two stages. First is the building of the experiment setup and programming of the robot. In RoboDK the 3D printing of an object is simulated, and that program is exported to the robot. The correct routing of the robot is tested by installing a pen in place of the extruder. The pen traces the path of the tool centre point (TCP) on a piece of paper attached to the printing bed.

In the second stage, the pen is replaced with a 3D printing extruder. Optimal printing parameters are determined by adapting the parameters for conventional 3D printing. The parameters are then optimised by printing test objects and fine tuning the settings. Several test objects are printed with optimised parameters to determine the limitations of the experiment setup.

2 THEORETICAL BACKGROUND

2.1 Standard 3D printing vs. Robotic 3D printing

Robotic 3D printing combines a standard multi axis industrial robot with a 3D printing head. The two main advantages of that are increased printing size capacity and additional motion flexibility. (Kauppila, 2021)

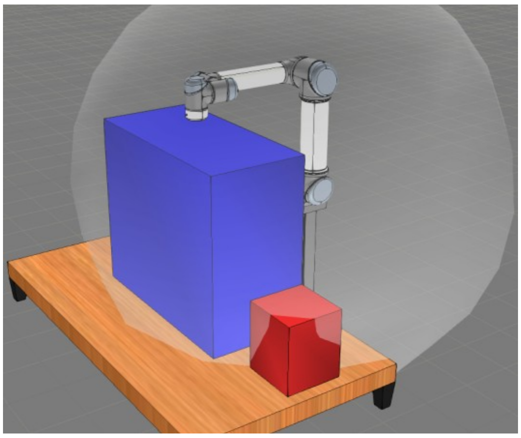


Figure 1. Printing volume comparison for Prusa i3 and Universal Robots UR5

The printing volume is determined by the size of the robotic arm. This can be additionally augmented by employing linear axes either for the robot or an augmented printing table.

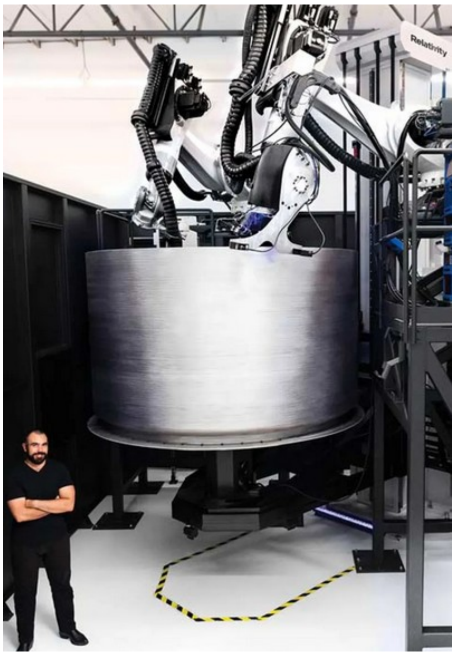


Figure 2. 3D printing a rocket (Relativity Space, n.d.)

Robotic 3D printing is not limited to the motions of a traditional XYZ printer, as it can print in six dimensions. It is also not bound to horizontal layer-by-layer printing, because it can manoeuvre around completed geometry. This has the potential to improve the strength of the final product, because the printing direction can be optimised to maximise the directional load capacity. The additional flexibility also means less printing supports are needed. Especially if articulated printing table is also used. The table can manipulate the angle of the printing, so that overhangs are avoided.

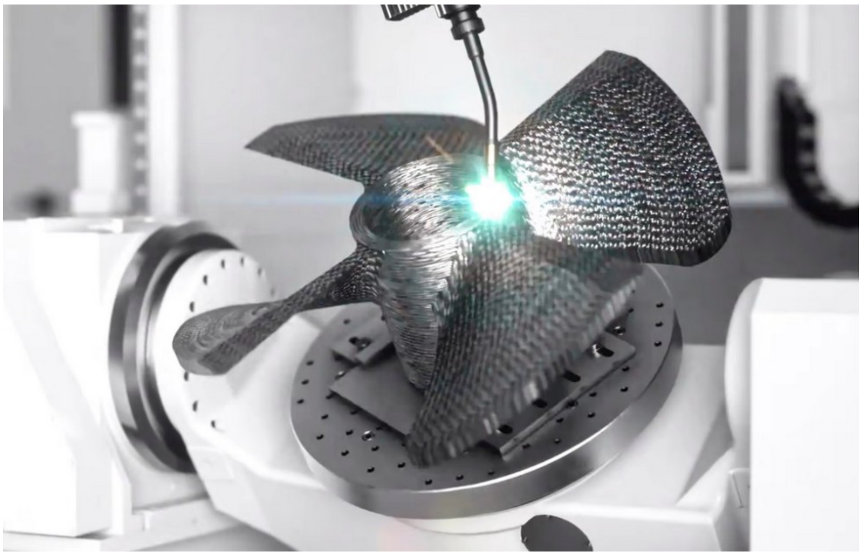


Figure 3. Robotic 3D printing with articulated welding table (Schwaar, 2022)

The main drawback of the technology is that there are currently few off-the-shelf solutions. And those that do exist require a high degree of knowledge, because the robot and the printing head often have to be controlled separately. Processes also have to be very well planned to avoid potential collisions. Print quality can be an issue, because of higher risk of vibrations and deviations, due to the size and configuration of the robotic cell. This can result in increased costs, since both equipment and labour are high value.

Taking the drawbacks into account, it is not worth printing small simple parts with a robot. But when size or complexity are a limiting factor for conventional 3D printing, robotic 3D printing is an asset.

2.2 **Use cases**

2.2.1 Printing oversized plastic products

Printing oversized plastic products is essentially same as traditional XYZ printing, only that the products can be much larger. Typical products at the moment are furniture and art projects, although more utilitarian uses are also being developed.

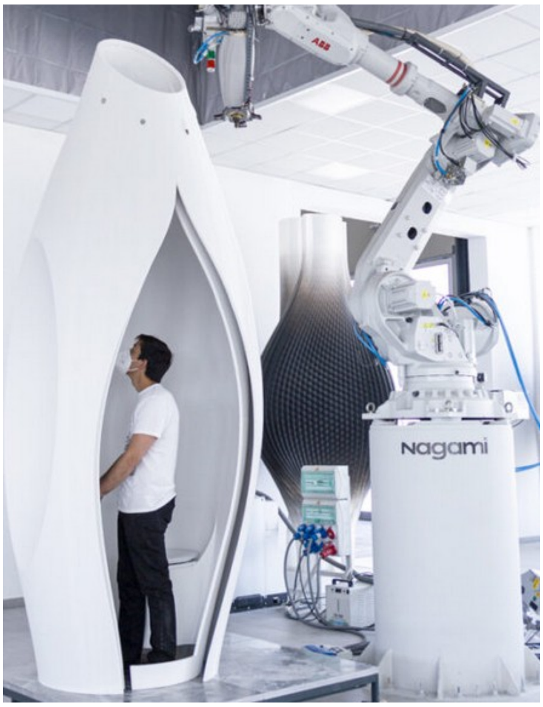


Figure 4. 3D printed portable toilet (Nagami, n.d.)

2.2.2 3D printing casting moulds

3D printing sand casting moulds is beneficial, because it is faster and uses less material than conventional methods.

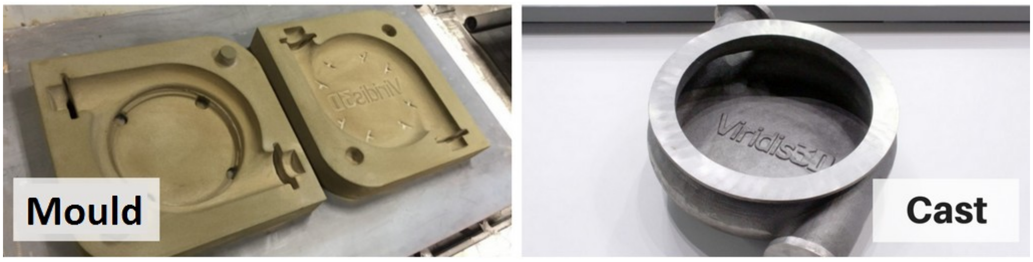


Figure 5. 3D printed sand casting mould (3HTi, n.d.)

2.2.3 Wire arc additive manufacturing

Wire arc additive manufacturing (WAAM) uses a welding robot to 3D print metal workpieces layer by layer (Schwaar, 2022). Because the printing accuracy is low, the contact surfaces have to be finished using CNC manufacturing.

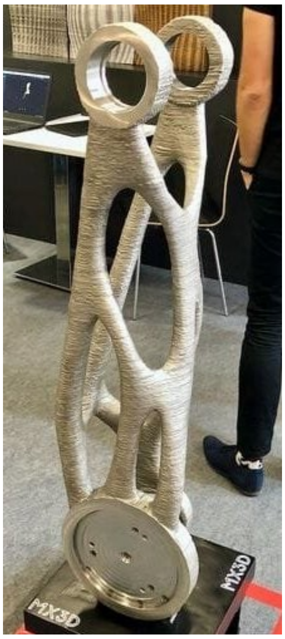


Figure 6. 3D printed replacement part for a robotic arm (Stevenson, 2019)

2.2.4 Concrete printing

Due to its large print capacity, robotic printing is optimal for architectural elements. The structures can be prefabricated in a factory (Branch, n.d.) or printed directly at the worksite.

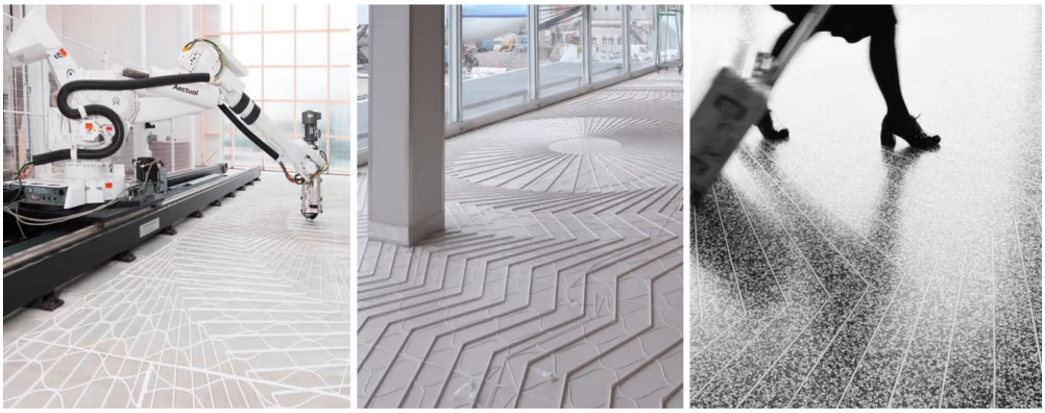


Figure 7. 3D printed terrazzo floor (Kauppila, 2021)

2.3 **Components and tools used in the experiment**

2.3.1 Printing head

The printing head is delivering the printing material to the printing surface. In fused deposition modelling (FDM) it consists of three parts: a motor, an extruder and a hotend. The motor drives the extruder to push plastic filament into the hotend. Once there, the plastic melts and is then deposited via a nozzle to a desired location. (Mensley, 2021)

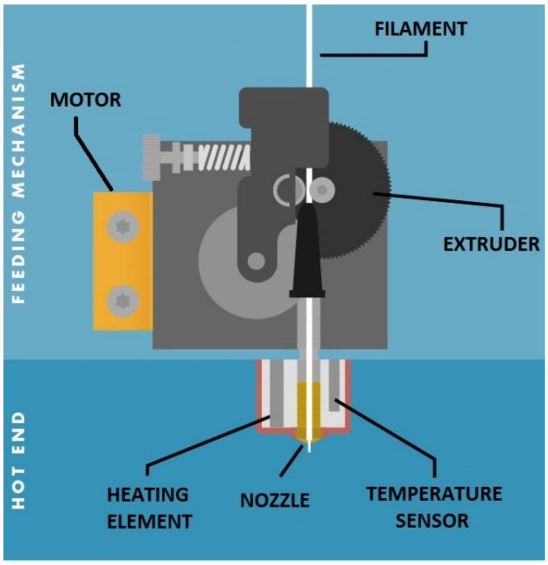


Figure 8. 3D printing head schematics (Budmen, n.d.)

2.3.2 Printing bed

The printing bed is the platform where the material is deposited. Depending on the material requirements it can be heated or unheated. It consists of an adjustable support structure, on which sits a glass plate. Glass is optimal, because it is rigid and thermally stable. The glass is covered with a plastic sticker, which ensures better first layer adhesion to the surface and also makes removing the print easier.

2.3.3 Filament

The printing material for FDM comes in the form of a plastic filament. The materials used are commonly polytetrafluorethylen (PTFE) or polylactic acid (PLA). PTFE has better engineering properties (MakeItFrom, n.d.), but one of the benefits of PLA is that it does not require a heated bed. To make the setup simpler, PLA is used for this experiment.

2.3.4 Robot simulation software

The purpose of the robot simulation software is creating and testing programs on a computer instead of directly on the robot. Conventionally this is done to maximise the robot's operational time and to optimise the processes. Complex operations, like CNC and 3D printing, can only be done using simulation software due to their complexity. For this experiment RoboDK and URSim are used. RoboDK is an industrial robots simulator and a tool for offline robot programming (RoboDK, n.d.). It was selected because it provides the possibility to create robotic 3D printing programs. URSim is a Universal Robots simulator. It is essentially the same software that is running on the robot. It enables the creation and verification the programs that can then be transferred to the robot.

2.3.5 Slicing software

Slicing software creates the gcode commands necessary for printing a 3D model. It does it by dividing a 3D model into horizontal layers and defines paths that the extruder has to move along to print those layers (Additive-X, 2021). Gcode is a programming language for computer numerical control (CNC) machines (RepRap, 2022). RoboDK uses Slic3r for this job. Slic3r is a free and open source slicing software (Slic3r, n.d.) It is highly modifiable, which makes it ideal for this purpose.

3 PREPARING THE EXPERIMENT

3.1 Pen

To make the pen holding jig, a standard ballpoint pen is 3D scanned and reverse engineered. Using the pen's 3D model, a closely fitting jig is designed and 3D printed. It consists of two parts. The base is static and is attached to the robot. The movable part can move in and out of the base. It is kept tensioned by a spring and held in place by a nut on the inside of the base. This mechanism makes it possible for the pen to be pressed against the paper without damage to any of the components.

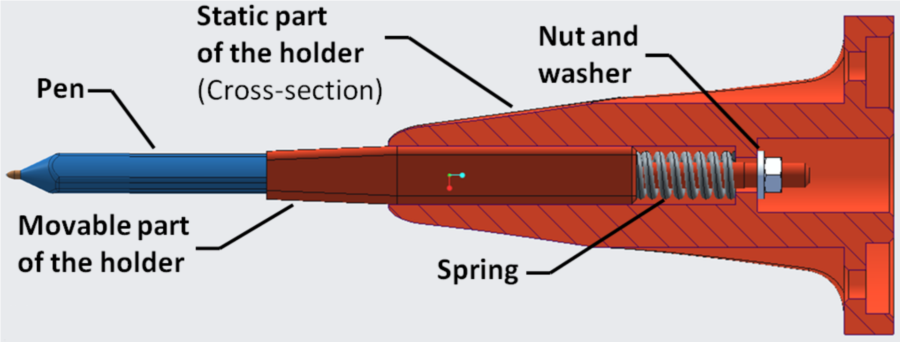


Figure 9. Pen holding jig schematics

3.2 Printing head

3.2.1 Mounting bracket

The printing head is mounted to the robot using a 3D printed bracket.

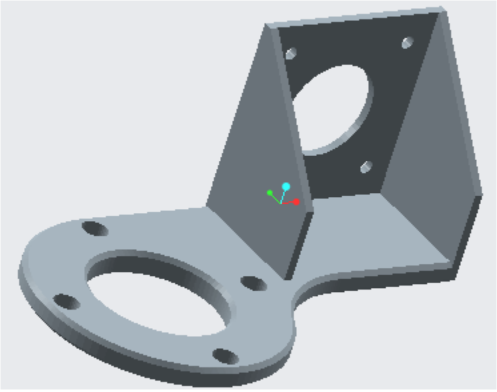


Figure 10. 3D model of the mounting bracket

3.2.2 Components

The printing head is assembled with components made by e3D. It has 24V Volcano HotEnd with 1.75 mm filament input. The filament is guided with a Titan extruder fitted with a 1.75 mm Bowden adapter. The extruder is driven with a high torque stepper motor.

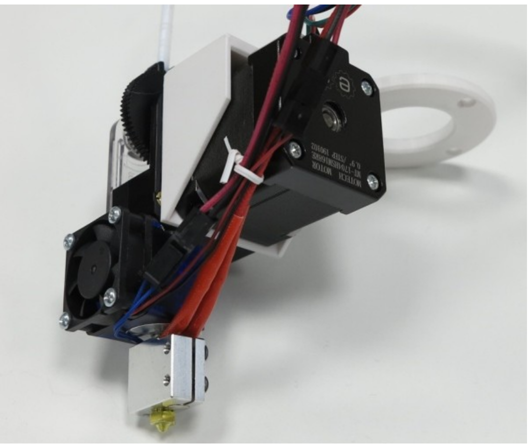
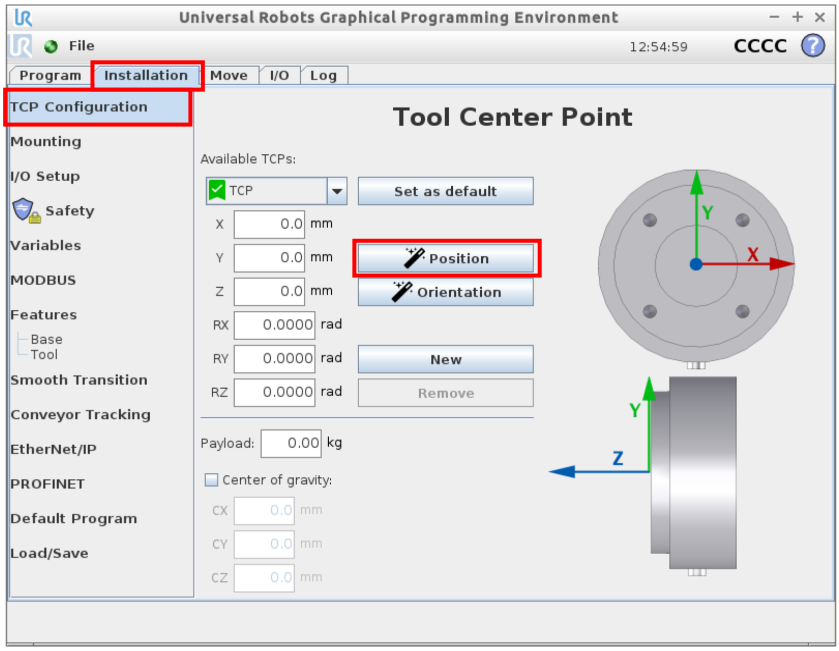


Figure 11. Assembled printing head

3.2.3 Tool centre point

The best way to find the tool centre point (TCP) is to use the tool provided by the robot. It can be found by going to "Program Robot", in the "Installation" tab in "TCP Configuration".



Tool Center Point

Available TCPs: ☑ TCP

X 0.0 mm
Y 0.0 mm
Z 0.0 mm
RX 0.0000 rad
RY 0.0000 rad
RZ 0.0000 rad

Payload: 0.00 kg

□ Center of gravity:
CX 0.0 mm
CY 0.0 mm
CZ 0.0 mm

Figure 12. TCP Configuration

To find the TCP, a static point needs to be used as the reference position. A screw held in place by a magnet is used in this case, but any pointy object is appropriate.

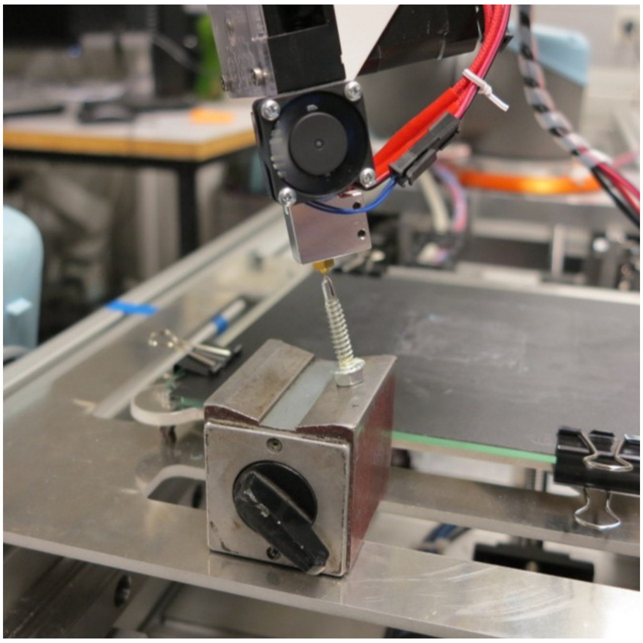
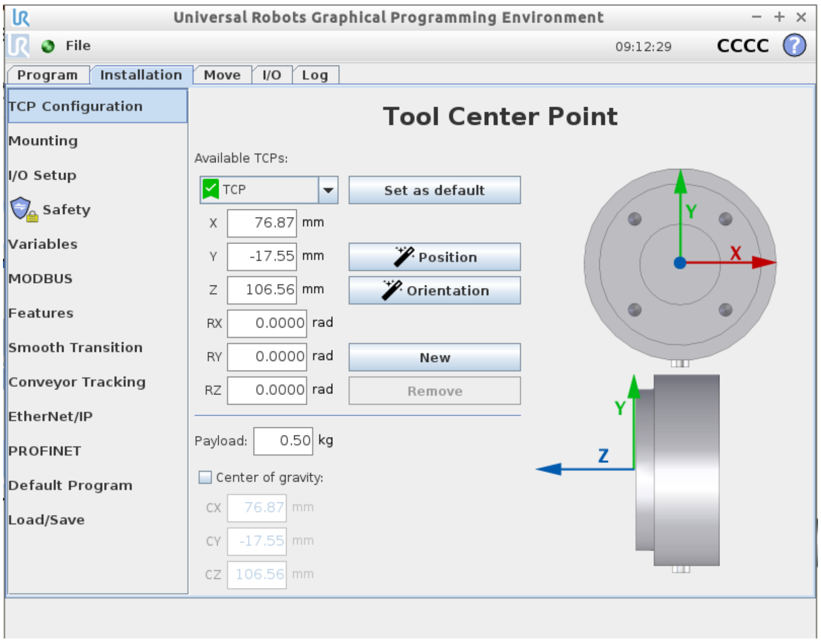


Figure 13. Saving a point for the TCP configuration

For detailed instructions about TCP configuration see Universal Robots Original Instructions page II-42.



Tool Center Point

Available TCPs: TCP

X 76.87 mm
Y -17.55 mm
Z 106.56 mm
RX 0.0000 rad
RY 0.0000 rad
RZ 0.0000 rad

Payload: 0.50 kg

□ Center of gravity:
CX 76.87 mm
CY -17.55 mm
CZ 106.56 mm

Figure 14. TCP after using the position wizard

3.2.4 Printing head control system

There are three components of the printing head that have to be controlled: the motor, hotend and cooling fan. For the experiment all three are controlled separately (see Figure 15). Photo of the setup is attached in Appendix 1.

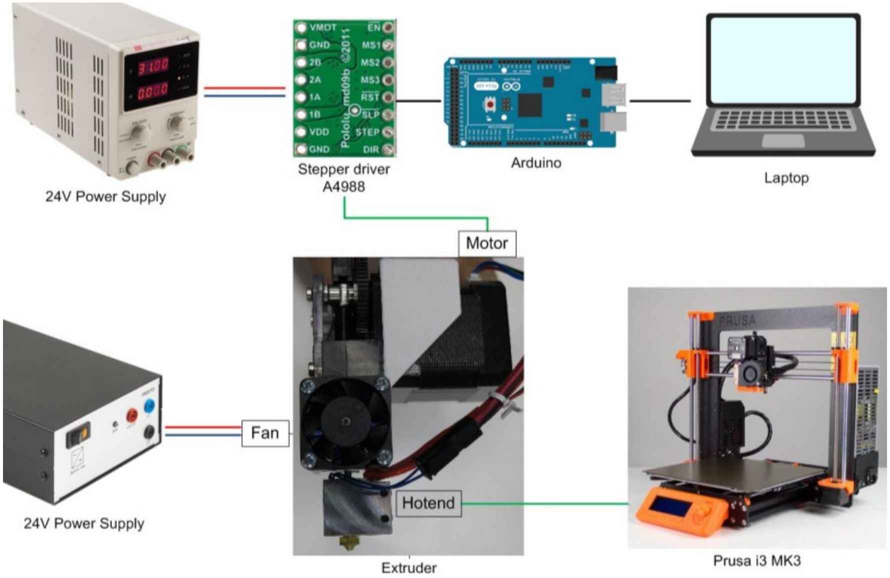


Figure 15. Printing head control schematic

3.2.5 Controlling the motor

The motor is controlled with an A4988 driver which receives its commands from an Arduino microcontroller. The wiring is done according to the wiring diagram (see Figure 16)

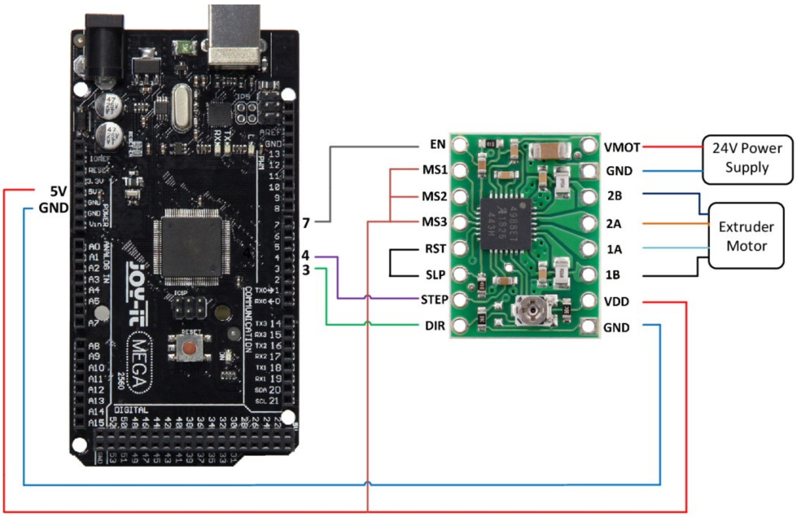


Figure 16. A4988 wiring diagram

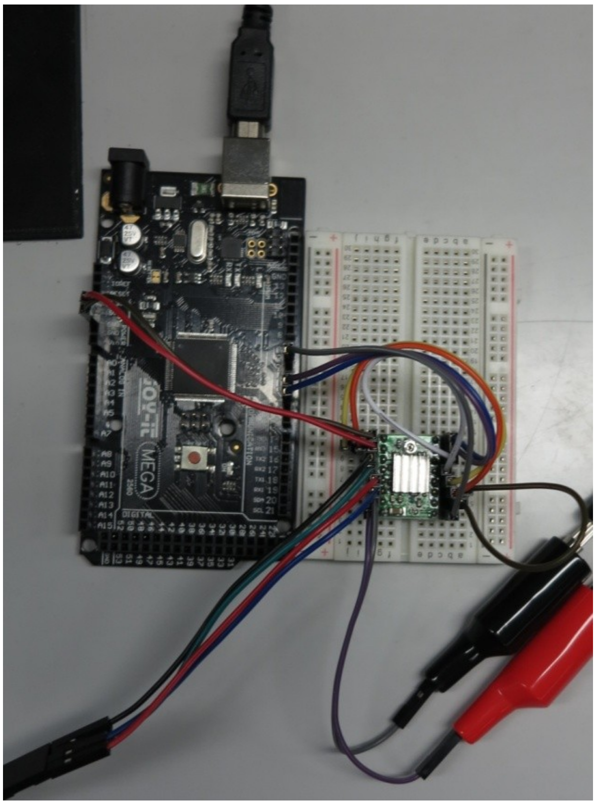
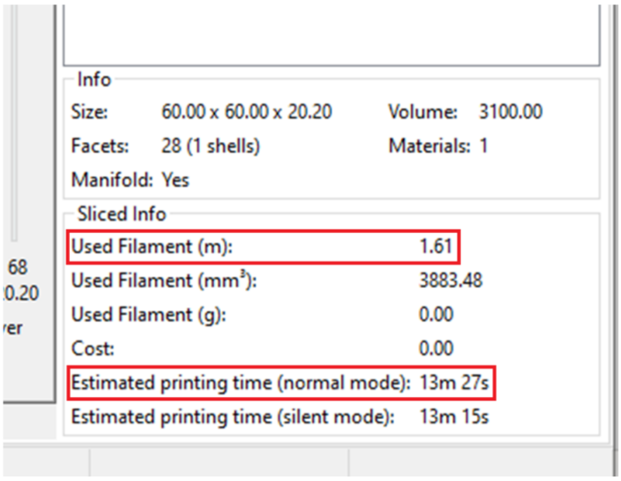


Figure 17. Motor driver wired to Arduino Mega

The Arduino is connected to a laptop via USB. The program is run directly from the computer. The program is attached in Appendix 2.

To drive the motor, its speed has to be determined. Using Slic3r an example 3D model is prepared for printing, using the same configuration as in paragraph 3.5.2. This produces the length of the filament and the printing time.



Info

Size: 60.00 x 60.00 x 20.20 Volume: 3100.00

Facets: 28 (1 shells) Materials: 1

Manifold: Yes

Sliced Info

Used Filament (m): 1.61

Used Filament (mm^3): 3883.48

Used Filament (g): 0.00

Cost: 0.00

Estimated printing time (normal mode): 13m 27s

Estimated printing time (silent mode): 13m 15s

Figure 18. Filament length and printing time in Slic3r

Using that data, the extrusion speed of 0.002 m/s is calculated. This can be used to calculate the rotational frequency of the motor. The filament is fed into the extruder using an 8 mm hob that is connected to the motor through a 1:3 transmission. The frequency of the motor is therefore:

$$f = 3 * \frac{v}{\pi * D} = 3 * \frac{0.002\left[m/s\right]}{\pi * 0.008\left[m\right]} = 0.239/s$$

Because a stepper motor is used to drive the extruder, the frequency has to be expressed as number of steps per second. One step equals 0.9 degrees of rotation. This means in order to complete the whole rotation, 400 steps are needed. To increase the smoothness of operation, the steps are reduced to 1/16 size. This increases the number of steps per rotation to 6400. The speed of the motor is therefore:

$$N = 6400 * f = 6400 * 0.239\left[/s\right] = 1530/s$$

Using that value as a starting point, it has been determined by trial and error that 1200/s is the optimal value. The speed calculation is also attached as Appendix 3.

3.2.6 Controlling the hotend

The hotend is controlled using a Prusa i3 MK3 printer. The heating and the thermistor cables are plugged to the control box of the printer.

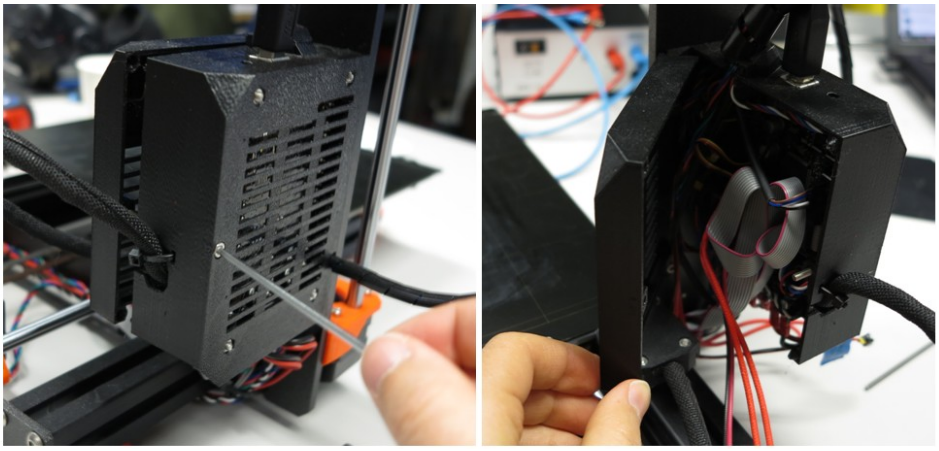
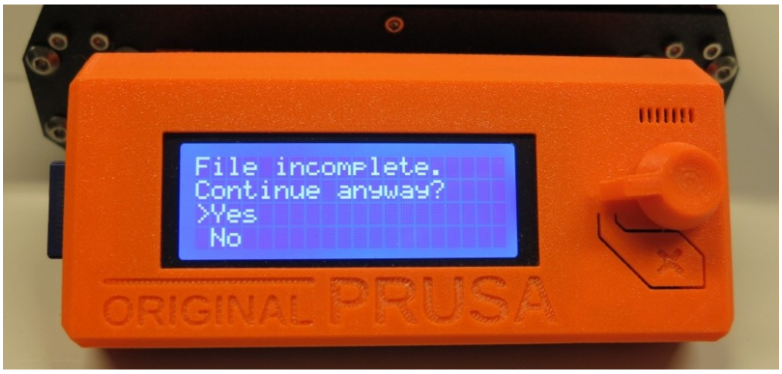


Figure 19. Accessing the printer control box

The temperature of the hotend is set by running a gcode program on the printer (See Appendix 4). It is transferred via an SD card same as other printing programs. Because it has only the basic temperature control commands, the printer asks for confirmation before initiation.



```
File incomplete.
Continue anyway?
>Yes
 No
```

Figure 20. Confirming initiation of an incomplete program

3.2.7 Controlling the fan

The fan is powered separately with its own 24 V power supply. It is turned on and off manually. The fan has to be operating whenever the hotend is heated. It can only be turned off after the hotend has cooled completely.

3.3 **Printing Bed**

3.3.1 Description

The printing bed is modelled on the Prenta Duo bed. Any other dimensions that fit inside the robot´s work envelope can also be used.

The bed is manufactured so that it fits on the linear rail where the UR5 robot is mounted. This way it can be assured that there aren´t any relative movements between the elements of the robot cell.

The glass from the original 3D printer is reused. Other non-standard parts are manufactured specifically for this purpose. The glass has a BuildTak 3D Print Surface glued to it, which makes first layer adhesion better and removing the print easier.

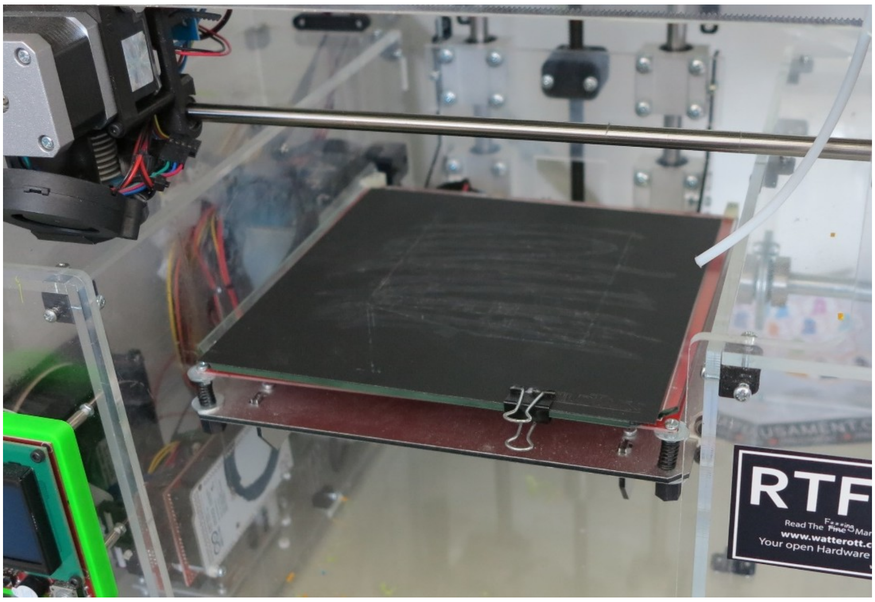


Figure 21. The printing bed that is the inspiration for the design

3.3.2 Assembly

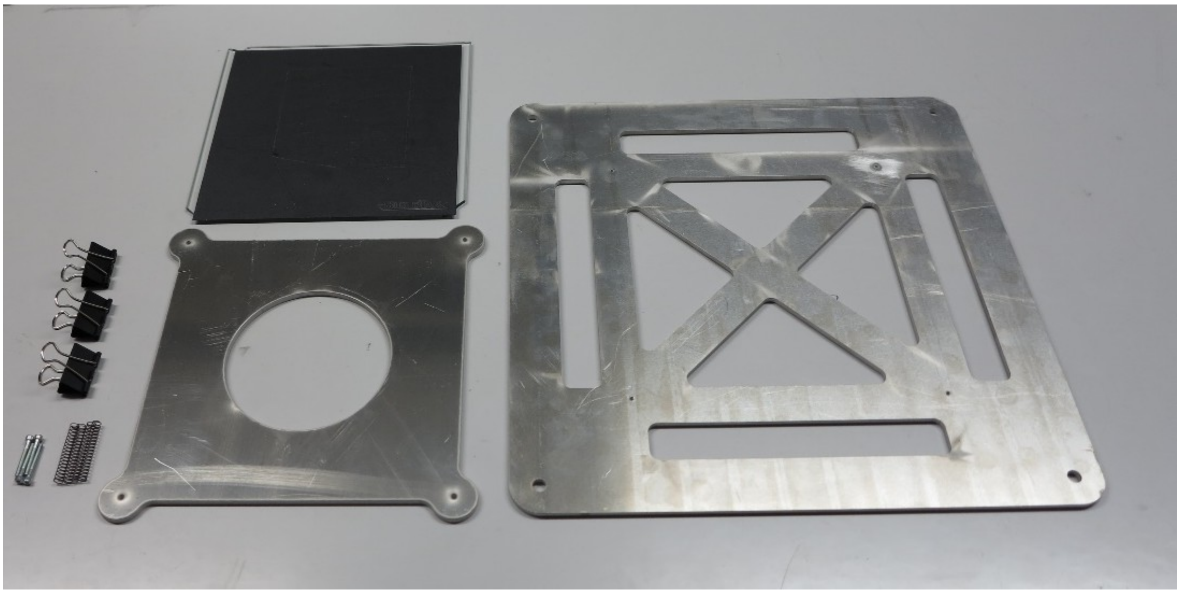


Figure 22. Parts required for the bed assembly

First step is putting the bolts through the two opposite holes on the small metal plate and placing springs on the bolts.

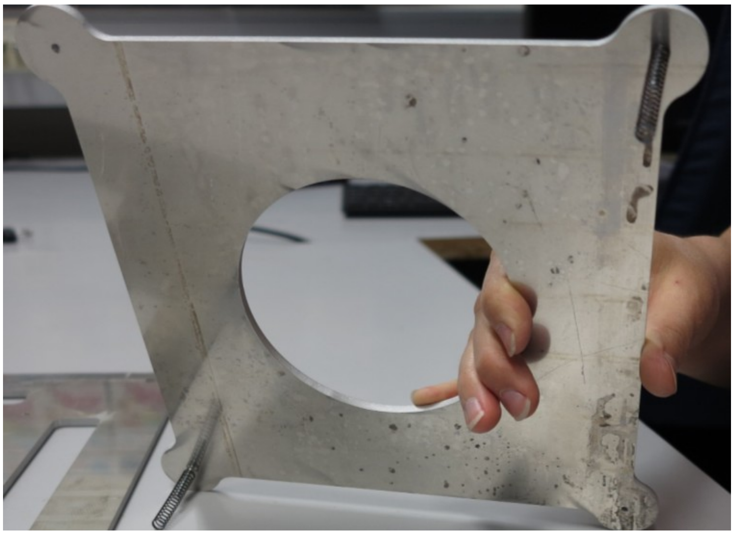


Figure 23. Springs inserted on the bolts

Next, the small metal plate is placed over the bigger one and the bolts are screwed into the corresponding threaded holes. The springs need to be compressed so that the bolts can reach the thread. Extra care needs to be taken on this step, since the springs are prone to coming loose.

The third and fourth springs are inserted by pressing the opposite corner of the top plate and inserting them under an angle.

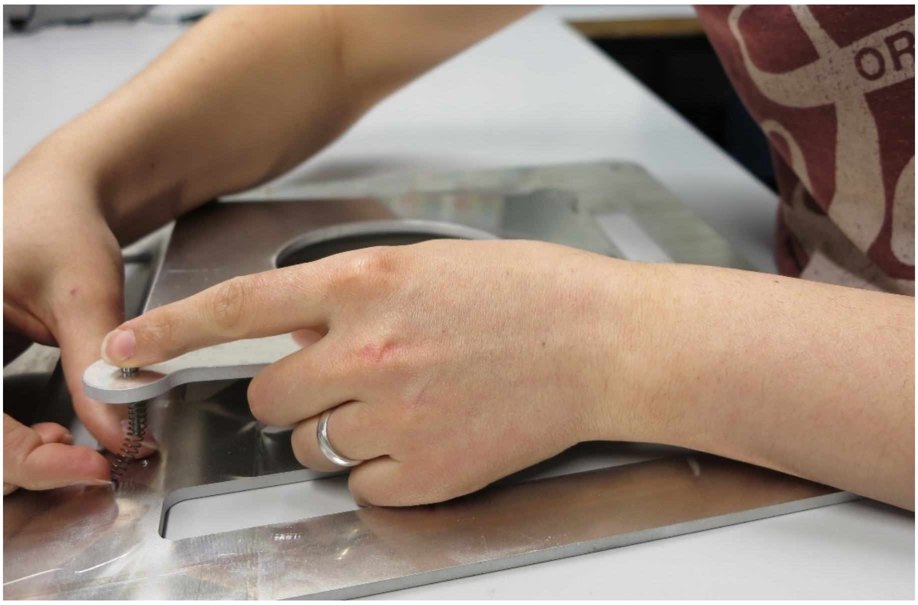


Figure 24. Inserting the third spring

Last step is tightening all the bolts to an equal degree, placing the glass over to top plate and securing it with binder clips.

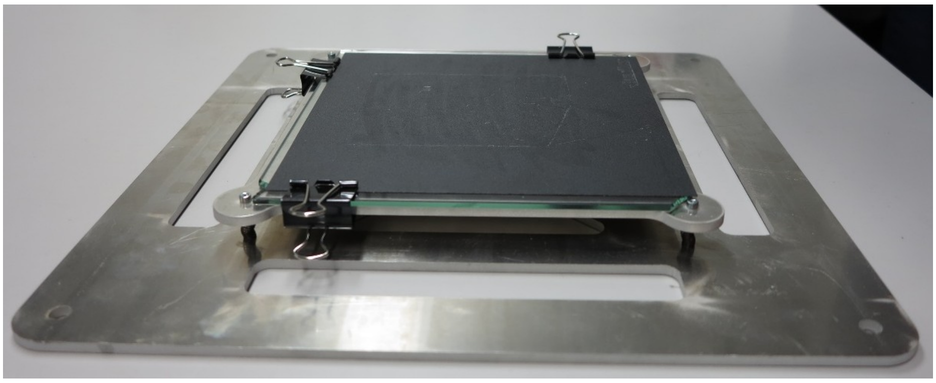
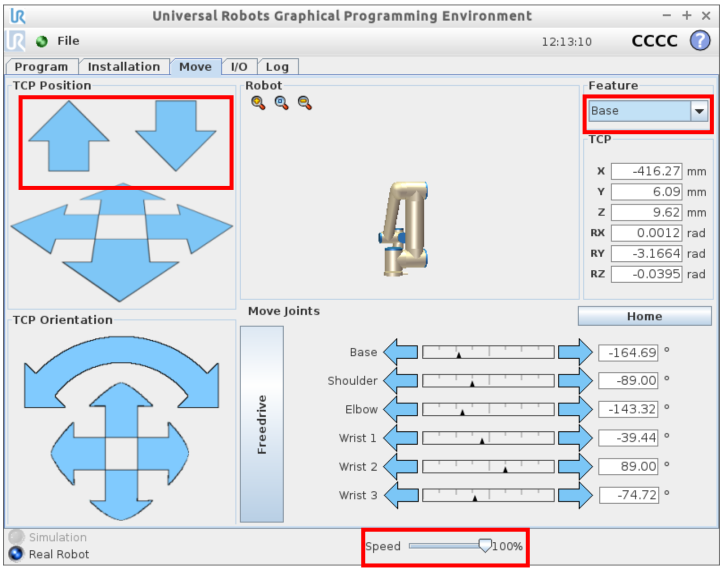


Figure 25. Completed bed assembly

3.3.3 Levelling

The bed has to be level with the robot´s coordinate system. The top surface is defined to be at z = 6 mm in the robot’s coordinate system. To level the bed the extruder has to be mounted on the robot and its TCP set.

When setting the robot in the correct position, it is first moved above the printing bed and it´s flange aligned using the freedrive mode. At this point the coordinates are approximate. It is important that the “Feature” is set to “Base”.



12:13:10 CCCC

Feature: Base

TCP

X	-416.27	mm
Y	6.09	mm
Z	9.62	mm
RX	0.0012	rad
RY	-3.1664	rad
RZ	-0.0395	rad

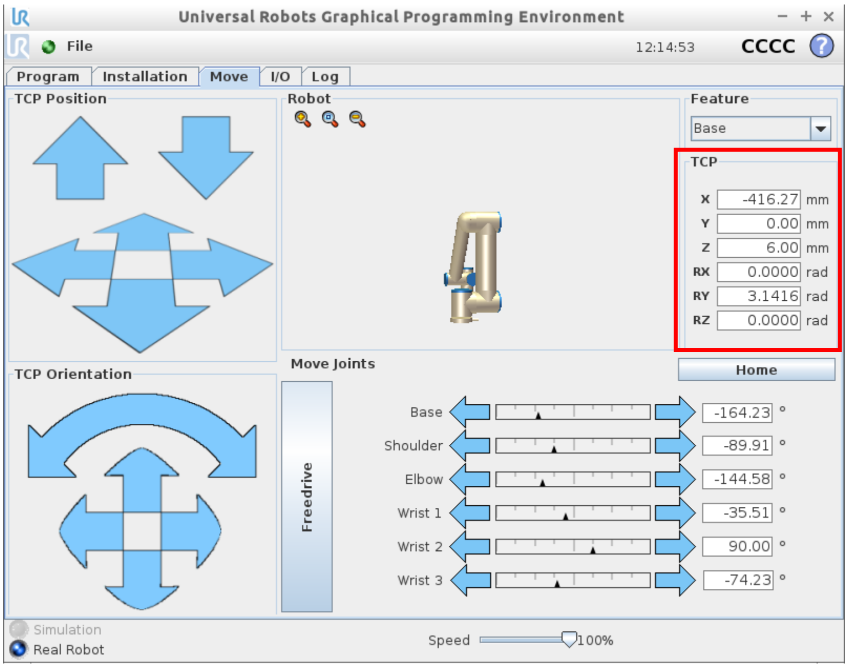
Base	-164.69 °
Shoulder	-89.00 °
Elbow	-143.32 °
Wrist 1	-39.44 °
Wrist 2	89.00 °
Wrist 3	-74.72 °

Speed 100%

Figure 26. Robot´s position after moving it above the printing bed

To get roughly the correct bed height, the robot is lowered using the arrows on the screen. To do it with higher precision, the speed can be reduced. The bolts holding the bed are adjusted so that when the z-coordinate of the TCP is roughly 6 mm, the nozzle is touching the bed. This is repeated for all four bolts.

Next the coordinates are set exactly. This is done by clicking on one of the TCP coordinates and inserting the values manually. The Z value and the rotational coordinates are set this way.



Feature: Base

TCP

X	-416.27	mm
Y	0.00	mm
Z	6.00	mm
RX	0.0000	rad
RY	3.1416	rad
RZ	0.0000	rad

Base	-164.23 °
Shoulder	-89.91 °
Elbow	-144.58 °
Wrist 1	-35.51 °
Wrist 2	90.00 °
Wrist 3	-74.23 °

Speed 100%

Figure 27. Robot´s position after setting exact coordinates

With the z coordinate being 6 mm, exact levelling is performed. This is done by sliding a piece of paper between the nozzle and the bed. The paper needs to be able to move, but at the same time there needs to be some resistance present. The height of the bed is again adjusted by turning the bolts. The levelling needs to be performed at least in each of the corners, but ideally in several locations on the bed. The robot is moved between the locations using the arrows in the “Move” tab.

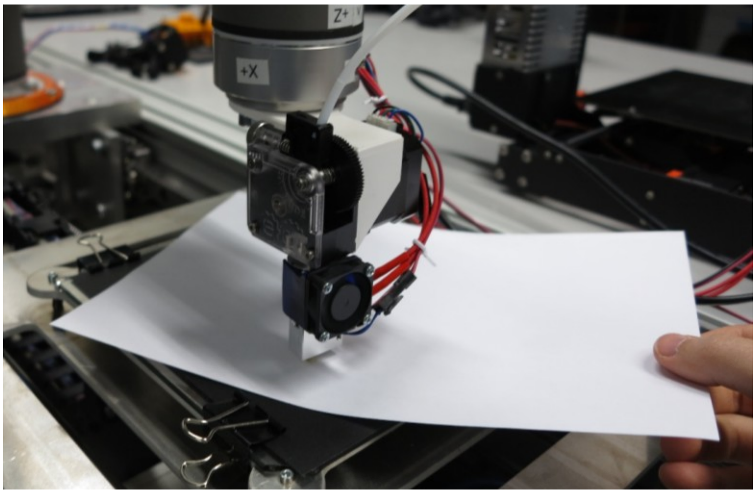


Figure 28. Bed levelling

3.4 **Filament Handling**

3.4.1 Spool holder

To minimize the possibility of the filament getting tangled, the spool is mounted on the lower arm of the robot. This adds load to the robot, but because all it needs to carry is the extruder, it is not a problem.

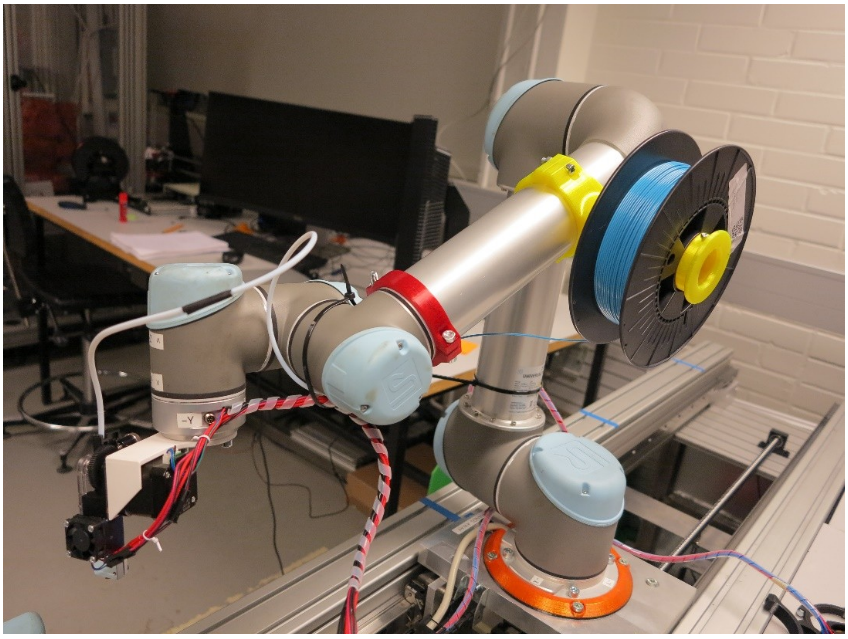


Figure 29. Spool holder and filament guide mounted on the robot

3.4.2 Filament guiding

Another measure to prevent filament tangling is guiding it inside a Bowden tube from the lower arm to the extruder. This way the filament is “free” only for a short distance and in static conditions. The Bowden tube is long enough that the whole range of wrist motions can be done.

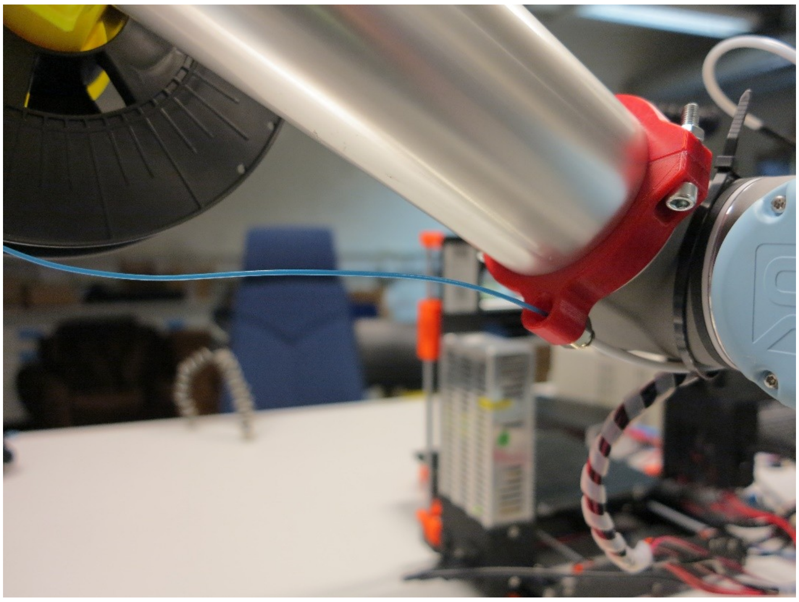


Figure 30. Guiding the filament in the Bowden tube

3.4.3 Loading the filament into the extruder

To load the filament the nozzle must be heated up. This is done by navigating to Menu>Settings>Temperature>Nozzle on the Prusa printer. The temperature is set to 220°C. When the correct temperature is reached, the filament is fed through the Bowden tube until it reaches the filament drive gear. Then the plastic gear is rotated to push the filament into the nozzle. The direction of the rotation is indicated on the body of the extruder. When the plastic starts extruding from the nozzle, the filament is properly loaded. The plastic gear should then be rotated in the opposite direction approximately 180°, to minimise oozing of plastic from the nozzle. The nozzle temperature is then set back to its initial value.

3.5 Preparing the simulation in RoboDK

3.5.1 Preparing the simulation setup

To create an accurate printing program, an exact model of the robot mounting setup is needed. This is done by acquiring the dimensions of the linear track and the printing bed. A model is created with 3D modelling software with the model´s coordinate system being set to coincide with the robot´s coordinate system. Lastly the model must be exported to a standard file format, i.e. STEP file.

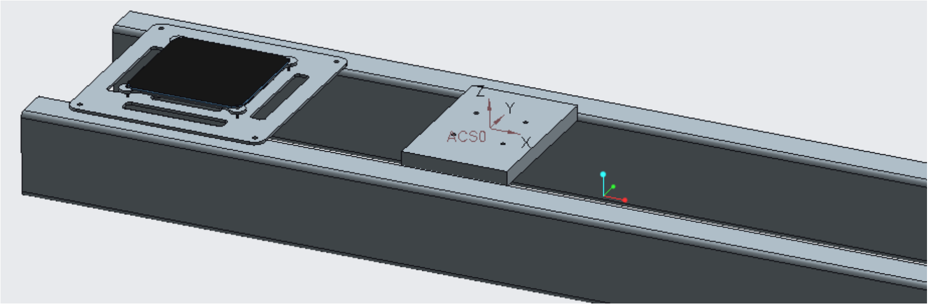
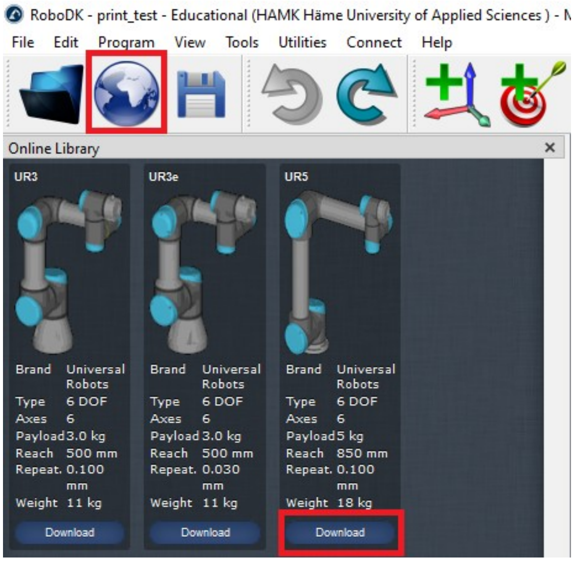


Figure 31. A 3D model of the mounting table (ACS0 coinciding with the robot coordinate system)

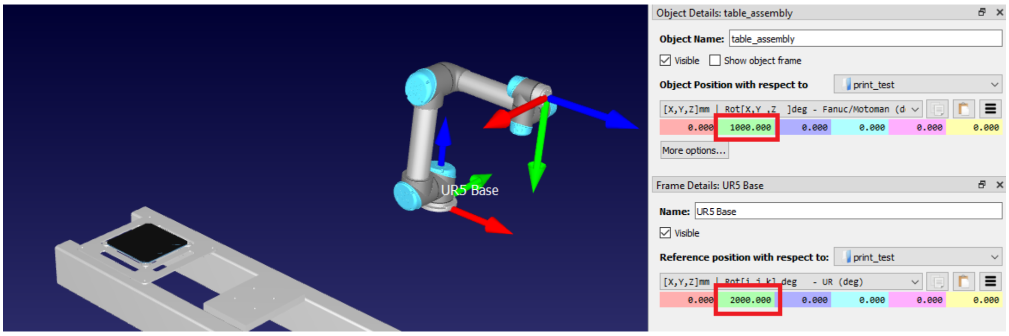
To prepare the simulation a new station is opened in RoboDK. The STEP file of the mounting table is dragged-and-dropped into the working area. Next the applicable robot is downloaded from the online library.



	UR3	UR3e	UR5
Brand	Universal Robots	Universal Robots	Universal Robots
Type	6 DOF	6 DOF	6 DOF
Axes	6	6	6
Payload	3.0 kg	3.0 kg	5 kg
Reach	500 mm	500 mm	850 mm
Repeat.	0.100 mm	0.030 mm	0.100 mm
Weight	11 kg	11 kg	18 kg

Figure 32. Importing a robot to the RoboDK simulation

When the table and the robot are imported, they might not be aligned. This is checked by opening their properties by double-clicking them and checking their position.



Object Details: table_assembly

Object Name: table_assembly

☑ Visible ☐ Show object frame

Object Position with respect to: print_test

[X,Y,Z]mm | Rot[X,Y ,Z]deg - Fanuc/Motoman (d

0.000	1000.000	0.000	0.000	0.000	0.000

More options...

Frame Details: UR5 Base

Name: UR5 Base

☑ Visible

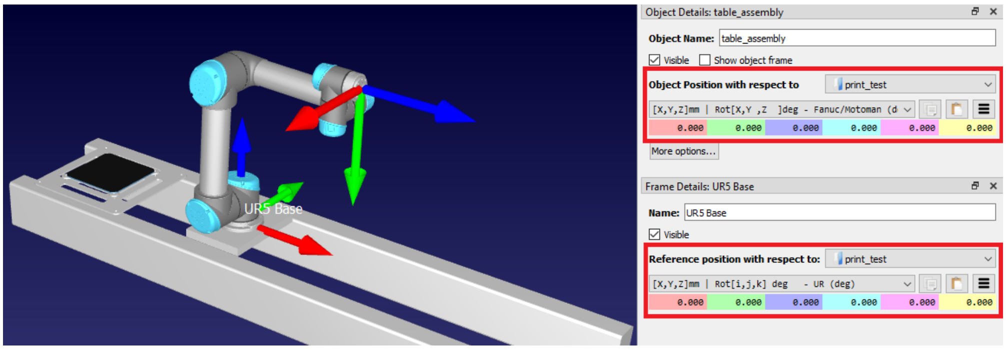
Reference position with respect to: print_test

[X,Y,Z]mm | Rot[i,j,k] deg - UR (deg)

0.000	2000.000	0.000	0.000	0.000	0.000

Figure 33. Table and robot misaligned

To achieve the alignment, their displacement in relation to the base coordinate system must be 0 in all directions.



Object Details: table_assembly

Object Name: table_assembly

☑ Visible ☐ Show object frame

Object Position with respect to: print_test

[X,Y,Z]mm | Rot[X,Y ,Z]deg - Fanuc/Motoman (d

0.000	0.000	0.000	0.000	0.000	0.000

More options...

Frame Details: UR5 Base

Name: UR5 Base

☑ Visible

Reference position with respect to: print_test

[X,Y,Z]mm | Rot[i,j,k] deg - UR (deg)

0.000	0.000	0.000	0.000	0.000	0.000

Figure 34. Table and robot aligned

The actual model of the extruder does not have to be used. All that is needed is its TCP, which is obtained in section 3.2.3.

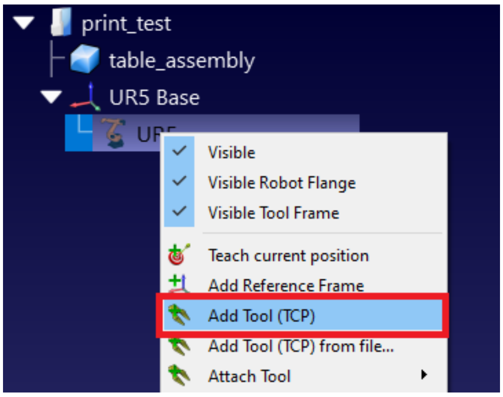
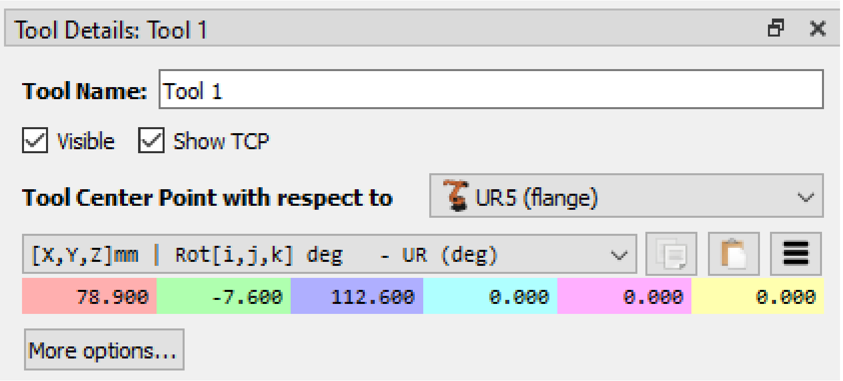


Figure 35. Adding a tool to the robot



Tool Details: Tool 1

Tool Name: Tool 1

☑ Visible ☑ Show TCP

Tool Center Point with respect to: UR5 (flange)

[X,Y,Z]mm | Rot[i,j,k] deg - UR (deg)

78.900	-7.600	112.600	0.000	0.000	0.000

Figure 36. Configured TCP

The last step in preparing the simulation is setting up the printing frame. This will be the coordinate system that will be the reference for printing movements. For ease of adding printing objects, it is best if the base of the coordinate system is in the corner of the printing table. The exact coordinates are obtained from the 3D modelling software.

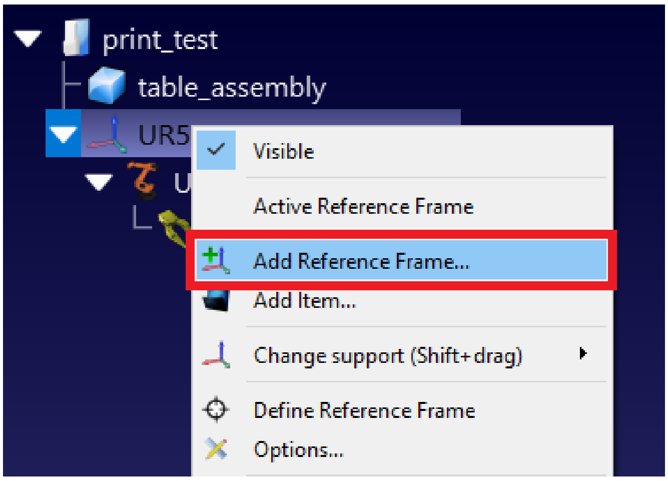
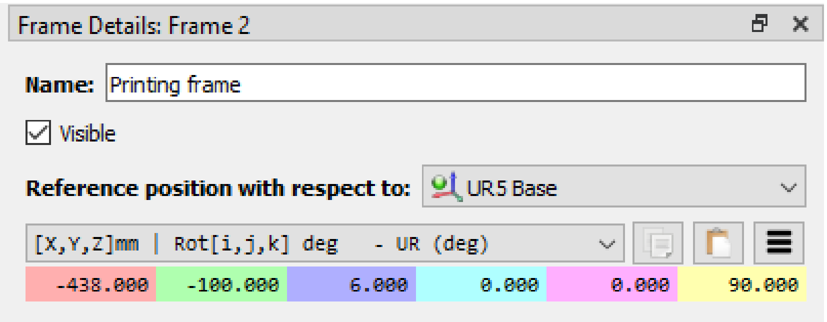


Figure 37. Adding a reference frame to be used as the printing frame



Frame Details: Frame 2

Name: Printing frame

☑ Visible

Reference position with respect to: UR5 Base

[X,Y,Z]mm | Rot[i,j,k] deg - UR (deg)

-438.000	-100.000	6.000	0.000	0.000	90.000

Figure 38. Position of the printing frame

This template is used for all further experiments, so it is recommended that it is saved and its copy is created for each printing experiment.

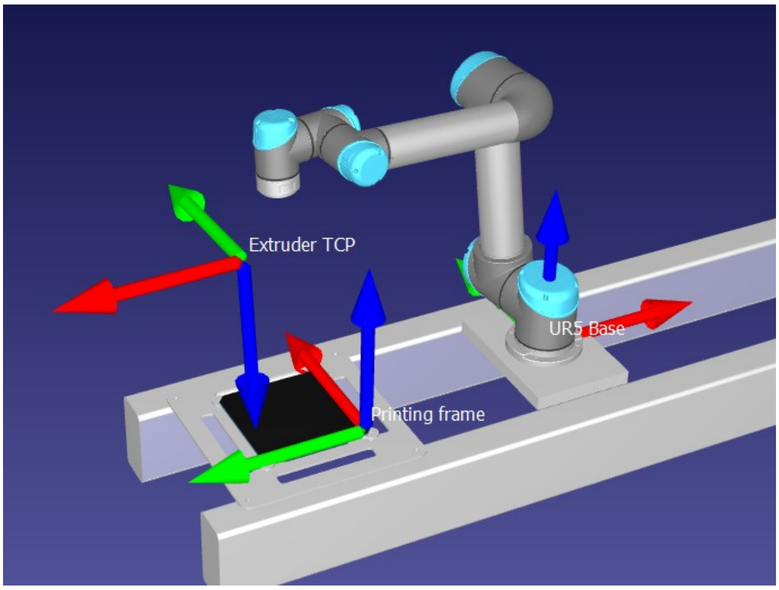


Figure 39. Completed setup in RoboDK

3.5.2 Configuring Slic3r inside RoboDK

RoboDK needs its own installation of Slic3r. Any existing installations cannot be used for this purpose.

The software can be downloaded from https://slic3r.org/download/, and the files should be extracted to C:\RoboDK\Other\Slic3r.

In order to create the user folders for Slic3r, the software has to be opened and then closed. The user files are then found at C:\Users\*[account name]*\AppData\Roaming\Slic3r. Printing settings can be imported to that location or the setting can be set directly in Slic3r.

Because of the parameters of the experiment, the printing speed should be set to be the same for all types of movements, except for non-printing moves. Those should be faster so as little of the filament as possible is extruded while non-printing moves are being performed. The printing speed is set to be low, so that all the printing types can be completed successfully.

To set the speed, RoboDK is opened after Slic3r has been installed, and a new printing project is started.

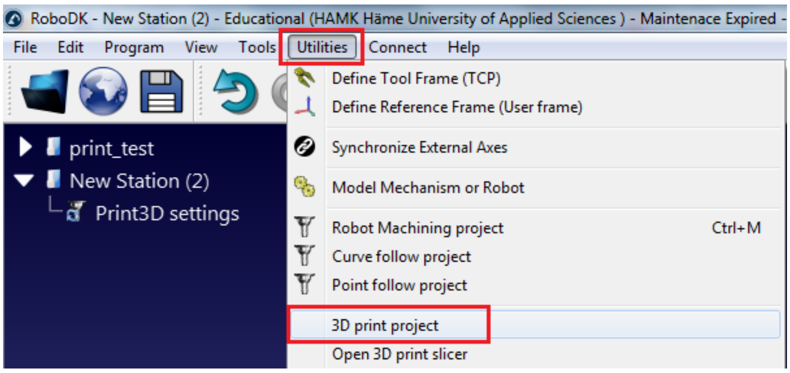
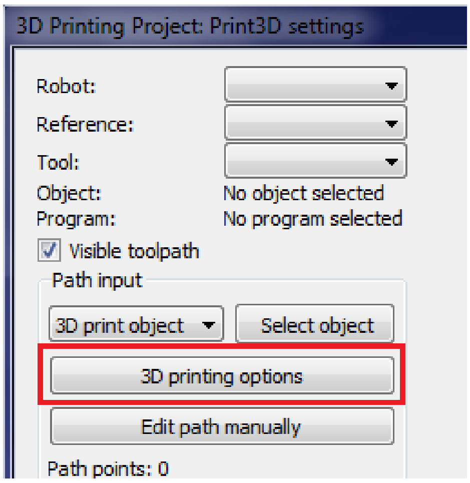

RoboDK - New Station (2) - Educational (HAMK Häme University of Applied Sciences) - Maintenace Expired -

Figure 40. Starting a printing project

To open Slic3r, the “3D printing options” button is selected.


3D Printing Project: Print3D settings

Robot:

Reference:

Tool:

Object: No object selected

Program: No program selected

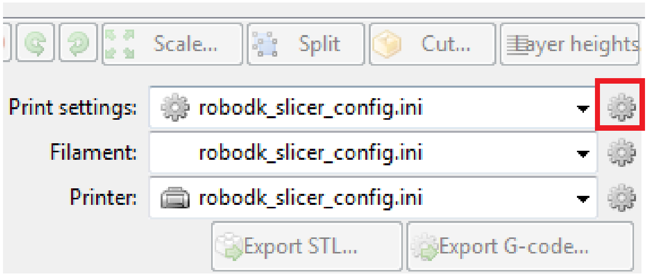
☑ Visible toolpath

Path input: 3D print object

Path points: 0

Figure 41. Opening Slic3r from RoboDK

For this experiment only the printing settings have to be modified. This is done by clicking on the printing icon next to the print profile.

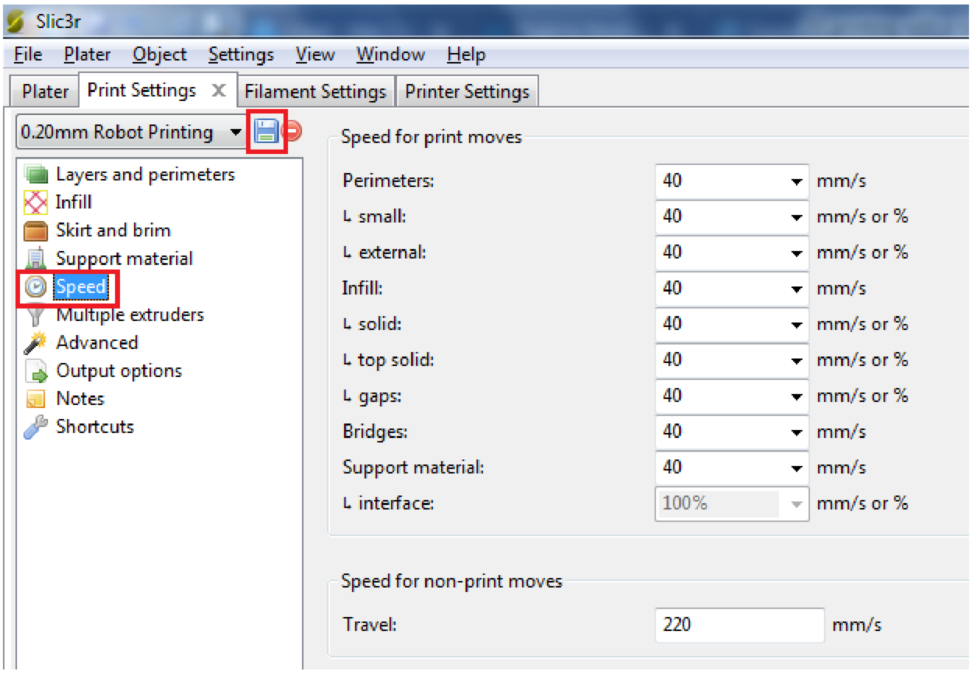

Print settings: robodk_slicer_config.ini

Filament: robodk_slicer_config.ini

Printer: robodk_slicer_config.ini

Figure 42. Opening the print settings

After that, the speed can be modified to the desired values and the profile is saved.



Speed for print moves

Perimeters:	40	mm/s
└ small:	40	mm/s or %
└ external:	40	mm/s or %
Infill:	40	mm/s
└ solid:	40	mm/s or %
└ top solid:	40	mm/s or %
└ gaps:	40	mm/s or %
Bridges:	40	mm/s
Support material:	40	mm/s
└ interface:	100%	mm/s or %

Speed for non-print moves

Travel:	220	mm/s

Figure 43. Setting the speed of the print

3.6 URSim

URSim can be downloaded from the Universal Robot (Universal Robots, n.d.-a). Attention needs to be paid to the selection of the same Polyscope version as is installed on the robot. The installation procedures for specific versions are available on the website.

4 CONDUCTING THE EXPERIMENT

4.1 Creating the robot program with the simulation software

The robot program is created in the template that is made in section 3.5.1.

First the 3D model (in .stl format) of the printing object is imported. When double-clicking the printing frame, the option to “Add Item” is available.

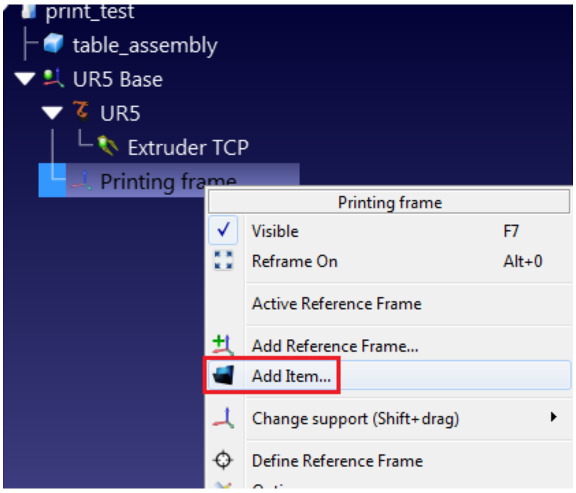


Figure 44. Adding an item to the printing frame

When successfully inserted, the printing object will appear on the printing bed.

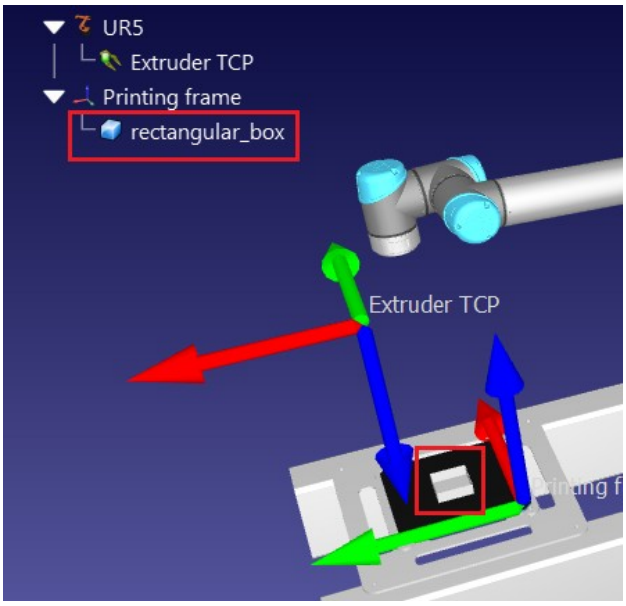


Figure 45. Object on the printing bed.

Next a new 3D printing project is started.

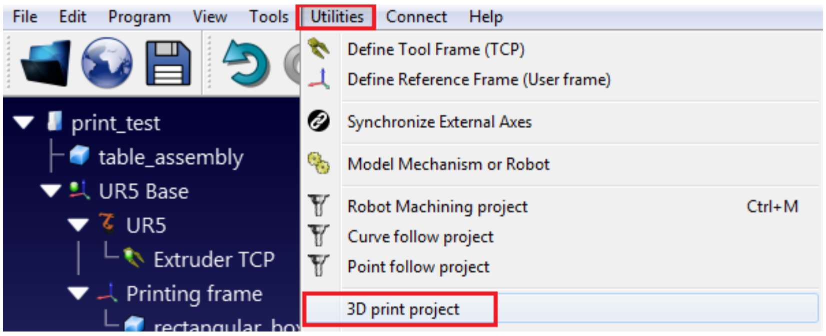
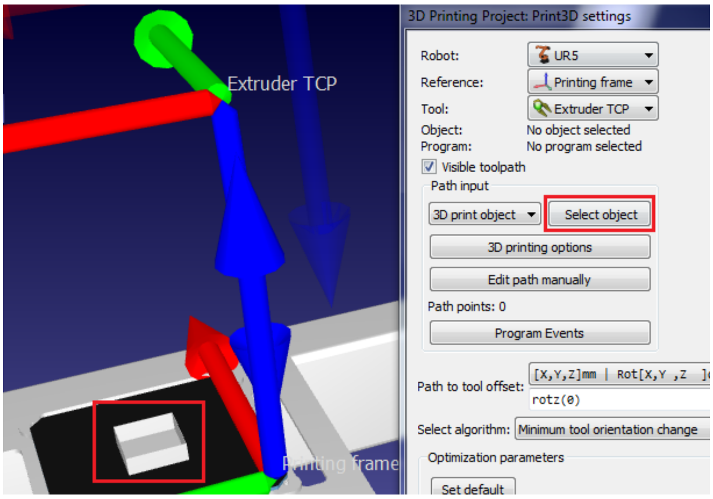


Figure 46. Starting a new 3D printing project

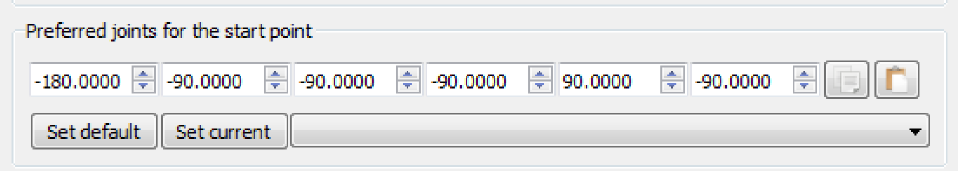
Most the settings for the printing project are already correctly selected by default. The printing object still has to be selected. The object is clicked on in the graphics window. The selection cannot be done in the station tree. After the selection is processed, the robots path becomes visible.


3D Printing Project: Print3D settings

Robot: UR5
Reference: Printing frame
Tool: Extruder TCP
Object: No object selected
Program: No program selected
☑ Visible toolpath
Path input: 3D print object
Path points: 0
Path to tool offset: [X,Y,Z]mm | Rot[X,Y ,Z]
rotz(0)
Select algorithm: Minimum tool orientation change
Optimization parameters

Figure 47. Selecting the printing object

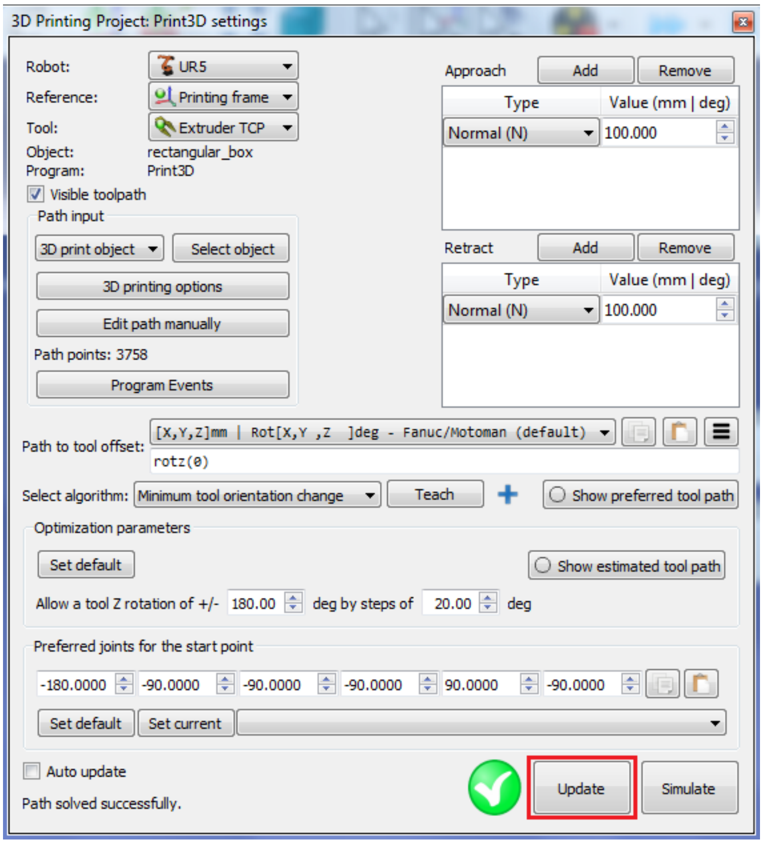
The starting position of the robot should be adjusted, so that it does not have to do much movement before the printing starts.


Preferred joints for the start point

-180.0000 | -90.0000 | -90.0000 | -90.0000 | 90.0000 | -90.0000

Figure 48. Start point values

Finally, the robot program is created by clicking the “Update” button. When the program is ready, the simulation can be run by clicking the “Simulate” button.



3D Printing Project: Print3D settings

Robot: UR5
Reference: Printing frame
Tool: Extruder TCP
Object: rectangular_box
Program: Print3D
☑ Visible toolpath

Path input

Path points: 3758

Approach Type	Value (mm | deg)
Normal (N)	100.000

Retract Type	Value (mm | deg)
Normal (N)	100.000

Path to tool offset: [X,Y,Z]mm | Rot[X,Y ,Z]deg - Fanuc/Motoman (default)
rotz(0)

Select algorithm: Minimum tool orientation change

Optimization parameters

Allow a tool Z rotation of +/- 180.00 deg by steps of 20.00 deg

Preferred joints for the start point

-180.0000 -90.0000 -90.0000 -90.0000 90.0000 -90.0000

□ Auto update

Path solved successfully.

Figure 49. Creating the printing program

When the program is working nominally, it is exported to a text file. Because RoboDK supports many robot manufacturers, the correct post processor has to be selected. This is done by right-clicking on “Print3D” and clicking “Select Post Processor”.

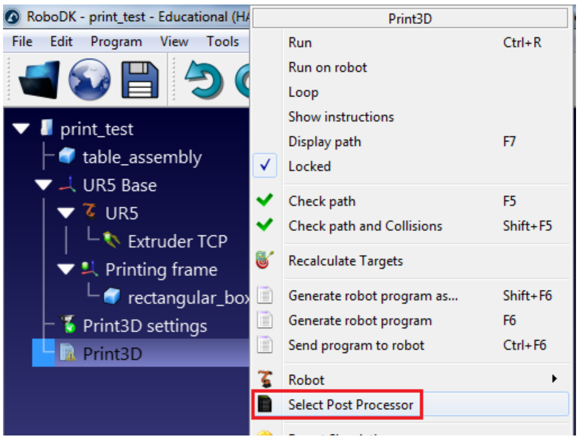


Figure 50. Opening the “Select Post Processor” window

In the window that opens, the Universal Robots post processor is selected.

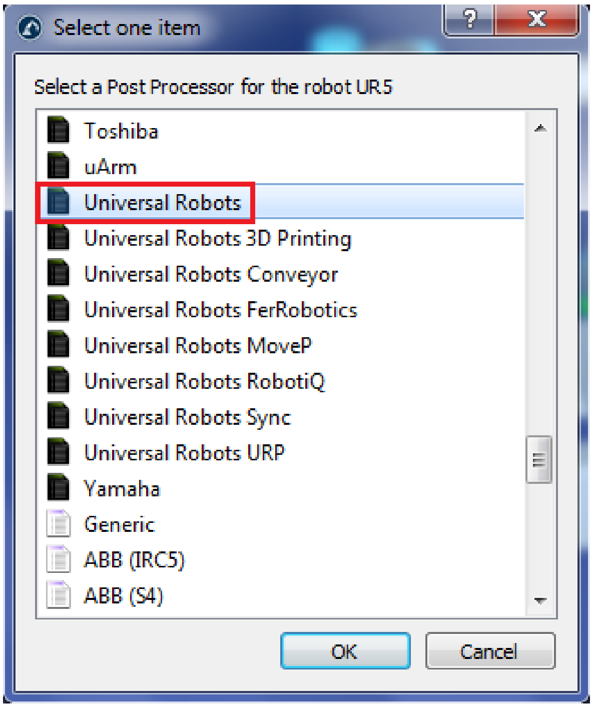


Figure 51. Selecting the Universal Robots post processor

Now the robot program can be exported. In the right-click menu the “Generate robot program as” is selected. The program is going to be saved as a .script file in the selected location.

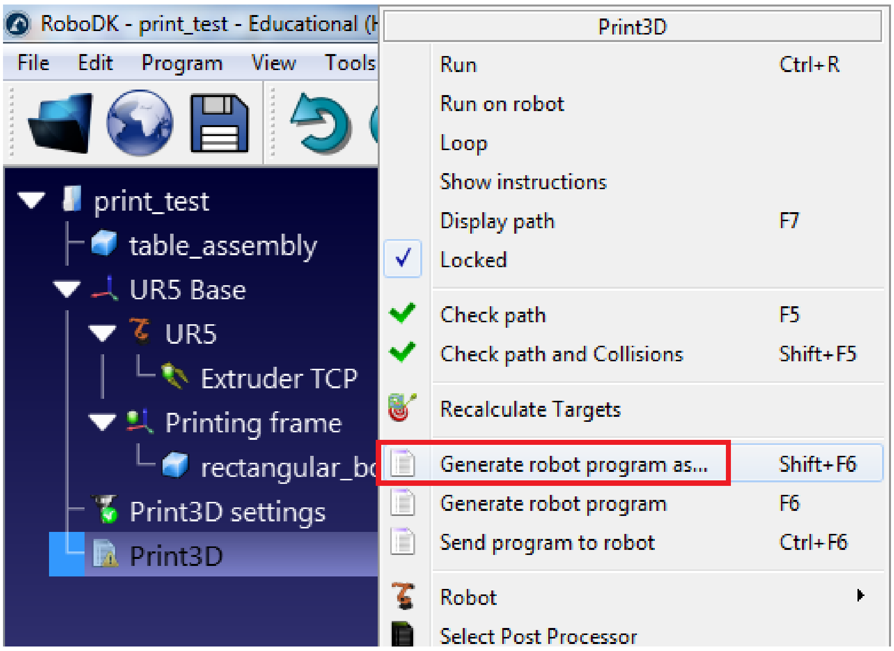
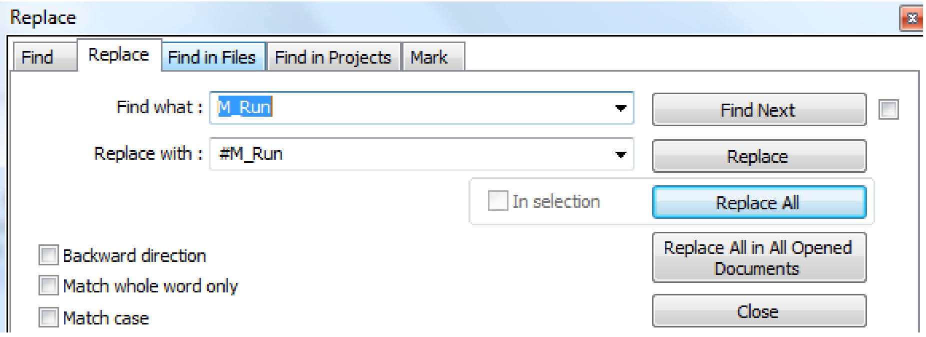


Figure 52. Exporting the robot program

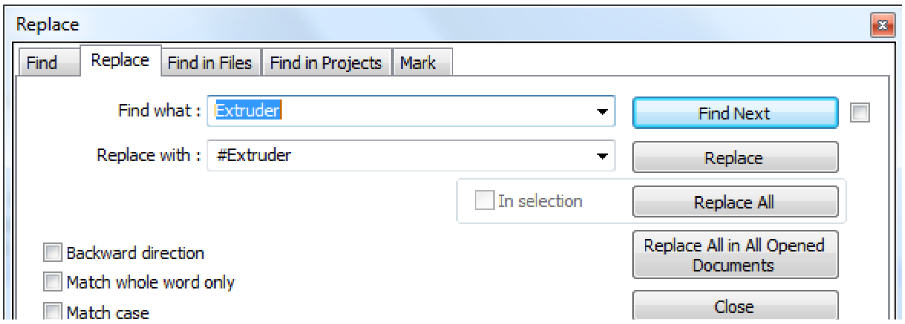
The exported files hold commands for the robot in text form. Among the commands are some that the robot is not able to interpret. These commands have to be commented out, so they do not cause an error. This is done by opening the file in any text editing software and using the Find/Replace tool.



Find what : M_Run

Replace with : #M_Run

Figure 53. Commenting out M codes



Find what : Extruder

Replace with : #Extruder

Figure 54. Commenting out extruder commands

4.2 Creating the robot program with URSim

Depending on its length, importing the .script file to a robot program can be a very long process. To avoid keeping to robot occupied for a long time, it is better to import it in URSim and then transfer the finished program to the robot. The added benefit is also that any unexpected behaviour can be detected in the URSim simulation.

The importing is done by saving the .script file on a USB stick and mounting that stick to the virtual machine where URSim is running. When the stick is successfully mounted, the software will offer to show the contents of the USB stick.

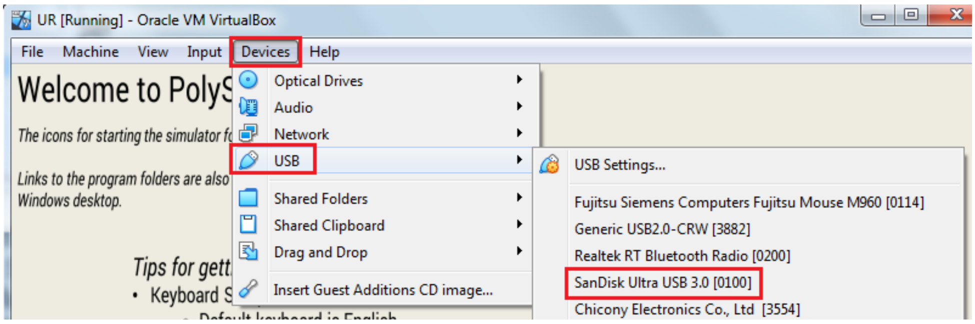
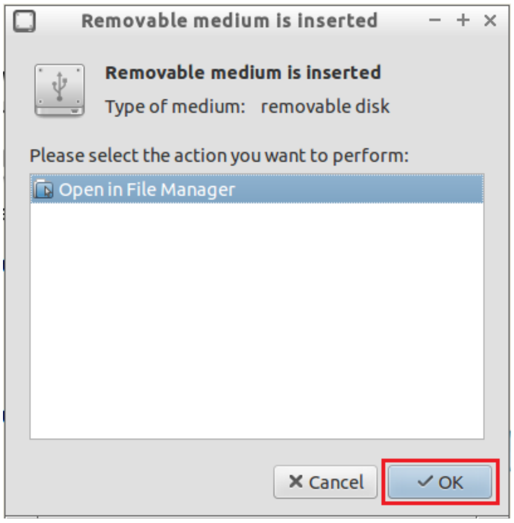

UR [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Welcome to PolyS
The icons for starting the simulator f
Links to the program folders are also
Windows desktop.
Tips for gett
• Keyboard S
Optical Drives
Audio
Network
USB
Shared Folders
Shared Clipboard
Drag and Drop
Insert Guest Additions CD image...
USB Settings...
Fujitsu Siemens Computers Fujitsu Mouse M960 [0114]
Generic USB2.0-CRW [3882]
Realtek RT Bluetooth Radio [0200]
SanDisk Ultra USB 3.0 [0100]
Chicony Electronics Co., Ltd [3554]

Figure 55. Mounting a USB stick to URSim


Removable medium is inserted

Removable medium is inserted

Type of medium: removable disk

Please select the action you want to perform:

Figure 56. Opening the USB stick

The .script file then has to be copied to the folder holding robot folders. Now it can be accessed from the user interface of the robot (inURSim).

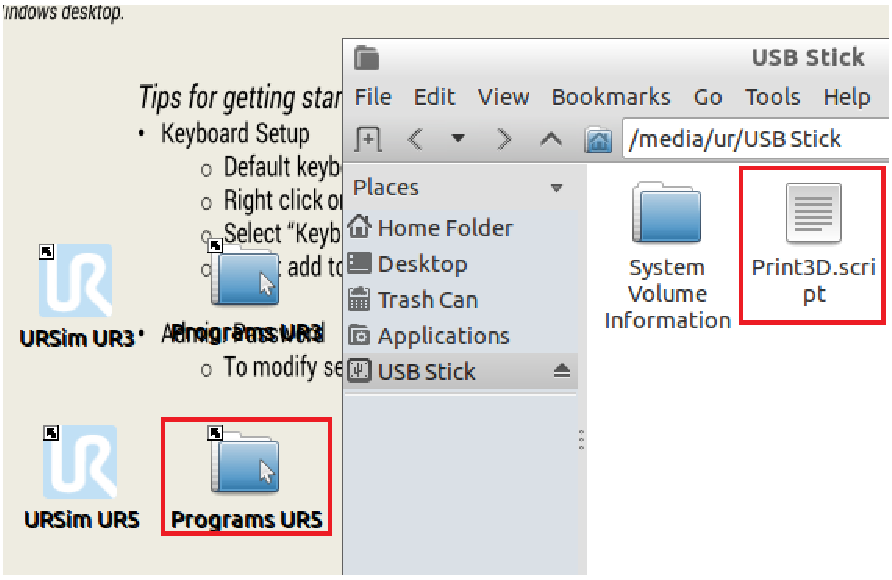
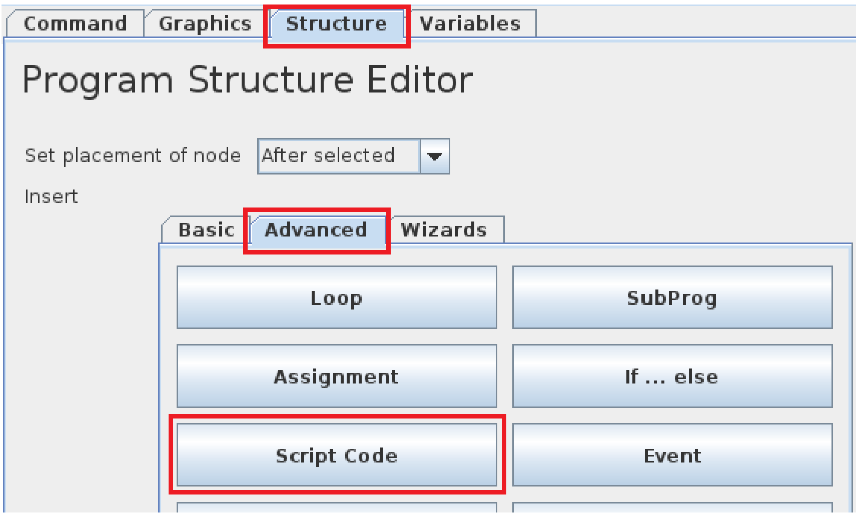


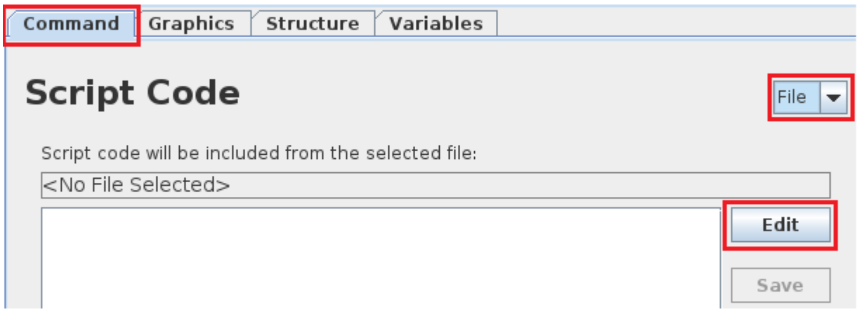
Figure 57. The .script file is copied to Programs UR5 folder

The URSim UR5 Polyscope simulation is then opened and an empty program is started. A “Script Code” command is added to the program and set to be read from a file. Using the “Edit” button, the File Editor is opened. The .script file is then imported using the “Open” button and the editor can be closed using the “Exit” button after that.



Set placement of node: After selected

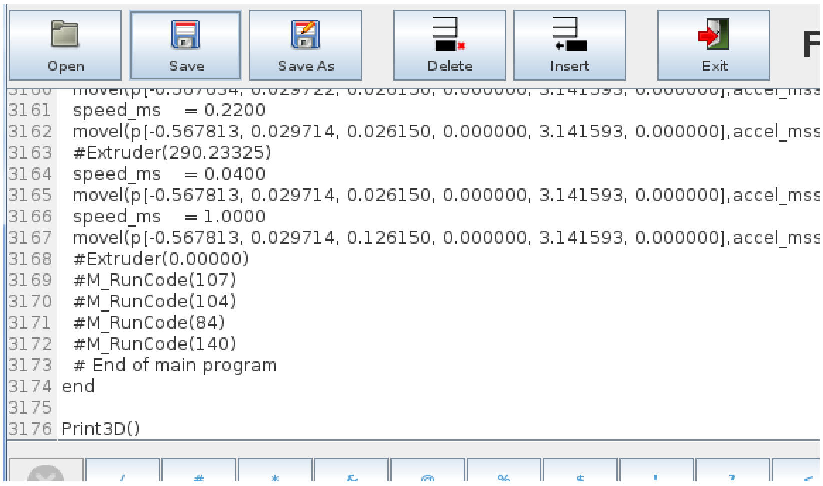
Figure 58. Adding “Script Code” to the program



Script code will be included from the selected file:

<No File Selected>

Figure 59. Importing the .script file



```
  speed_ms    = 0.2200
  movel(p[-0.567813, 0.029714, 0.026150, 0.000000, 3.141593, 0.000000],accel_mss
  #Extruder(290.23325)
  speed_ms    = 0.0400
  movel(p[-0.567813, 0.029714, 0.026150, 0.000000, 3.141593, 0.000000],accel_mss
  speed_ms    = 1.0000
  movel(p[-0.567813, 0.029714, 0.126150, 0.000000, 3.141593, 0.000000],accel_mss
  #Extruder(0.00000)
  #M_RunCode(107)
  #M_RunCode(104)
  #M_RunCode(84)
  #M_RunCode(140)
  # End of main program
end

Print3D()
```

Figure 60. File editor

The program can now be run to verify correct movements. When nominal operation is confirmed, the program can be saved and Polyscope closed. It is saved as a .urp file in the Programs UR5 folder. The .urp file should then be copied to the USB stick and the usb stick unmounted by repeating the procedure from Figure 55. After unmounting, the USB stick will appear in the native operating system again.

4.3 Uploading the program to the robot

To upload the program to the robot an USB stick with a "Magic file" is needed. This is a piece of software that automatically transfers the USB stick's contents on the robot. It can be downloaded from the Universal Robots website (Universal Robots, n.d.-b)

4.4 Testing the program on the robot

The program that has been made in URSim can be directly run on the robot. It is recommended that before the printing is initiated, a test run is performed with reduced speed and without extrusion.

4.5 3D printing

The operational sequence for 3D printing is as follows:

1. Start the cooling fan

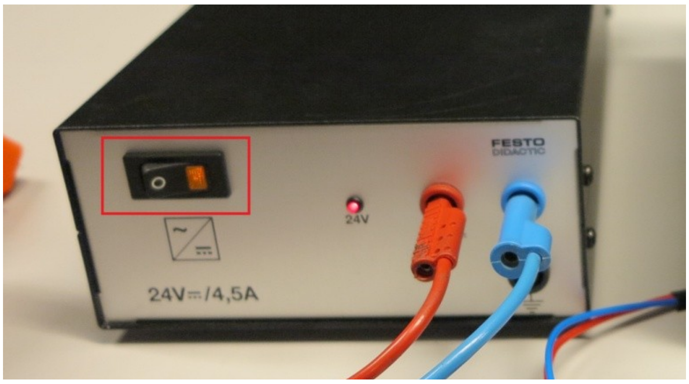


Figure 61. Turning on the fan's power supply

2. Start the heating of the hotend by pressing the control knob, navigating to “Print from SD” and opening the heating program.

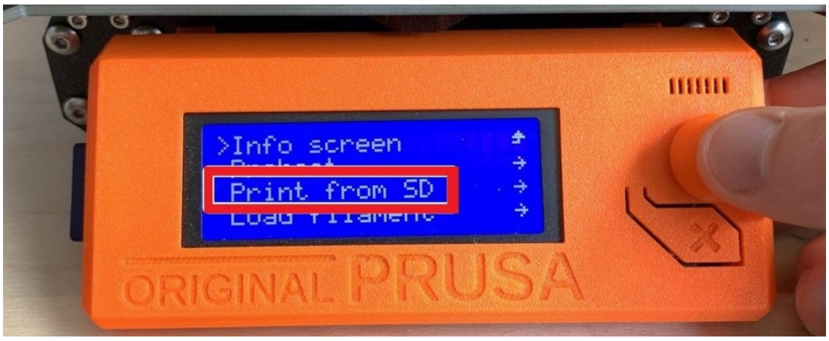


Figure 62. Starting the heating program on the printer (Lutz, 2020)

3. Wait to reach the target temperature.

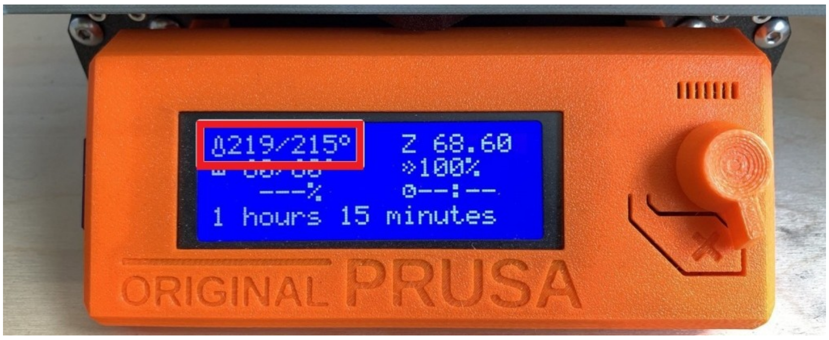
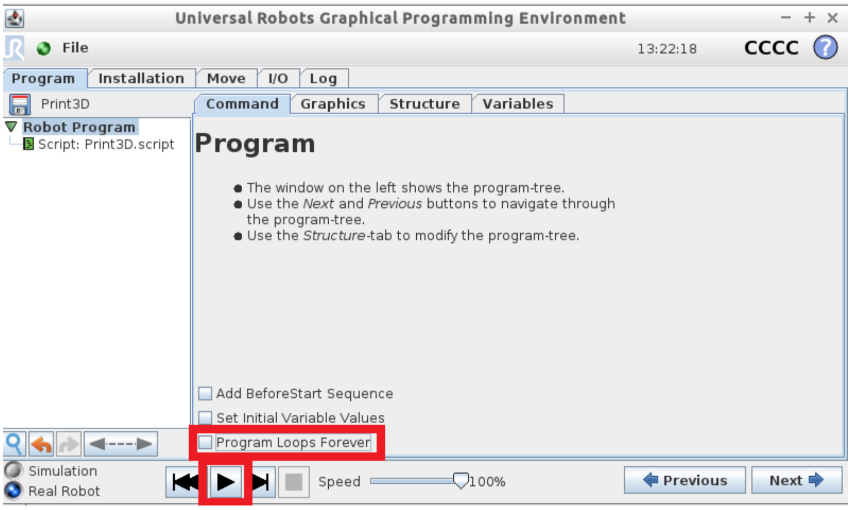


Figure 63. Printer showing the target and current temperatures (Lutz, 2020)

4. Start the robot program. Make sure the “Program Loops Forever” option is off.



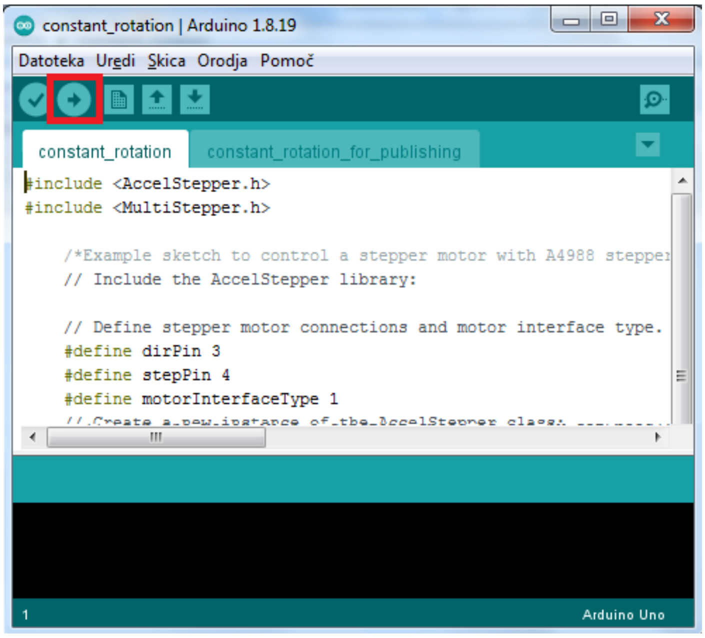
Program

- The window on the left shows the program-tree.
- Use the *Next* and *Previous* buttons to navigate through the program-tree.
- Use the *Structure*-tab to modify the program-tree.

□ Add BeforeStart Sequence
□ Set Initial Variable Values
□ Program Loops Forever

Figure 64. Starting the robot program

5. Start the extrusion


constant_rotation | Arduino 1.8.19
Datoteka Uredi Skica Orodja Pomoč
constant_rotation
constant_rotation_for_publishing
1
Arduino Uno

```
#include <AccelStepper.h>
#include <MultiStepper.h>

    /*Example sketch to control a stepper motor with A4988 stepper
    // Include the AccelStepper library:

    // Define stepper motor connections and motor interface type.
    #define dirPin 3
    #define stepPin 4
    #define motorInterfaceType 1
    // Create a new instance of the AccelStepper class:
```

Figure 65. Running an Arduino program from PC

6. Wait for the print to finish

7. Stop the extrusion by unplugging the Arduino’s USB connection to the PC.

8. Turn off the hotend by pressing the control button and navigating to “Stop print”.

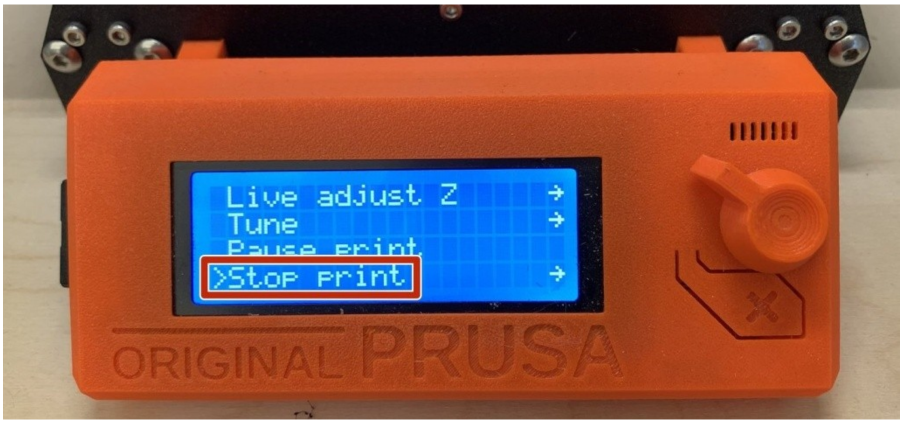


Figure 66. Turning off the hotend (Lutz, 2020)

9. Wait for the hotend to cool.

10. Turn off the cooling fan.

5 FINDINGS

5.1 Pen experiment

Printing simulation with a pen in place of the extruder proved the routing to be correct. It was drawing clean and consistent lines. Movements were also repeatable between different attempts.

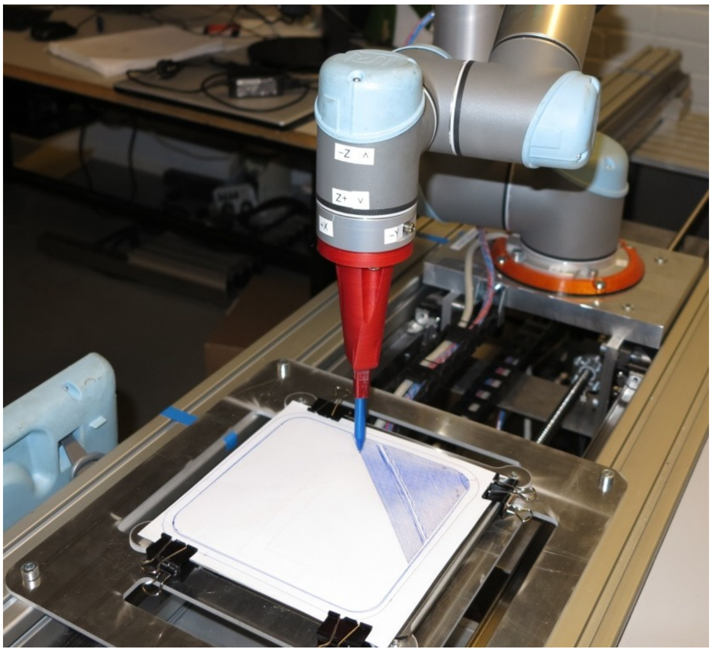


Figure 67. Pen experiment in progress

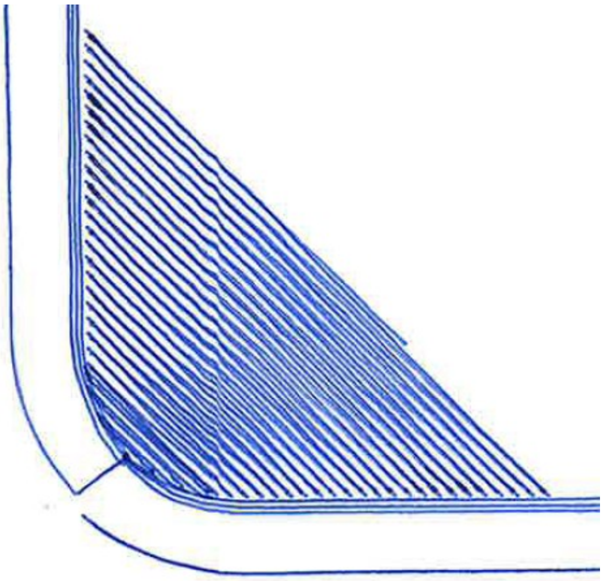


Figure 68. Pen experiment result

Disruption resistance was tested by pushing on the robot while it was drawing. This resulted in path deviations.

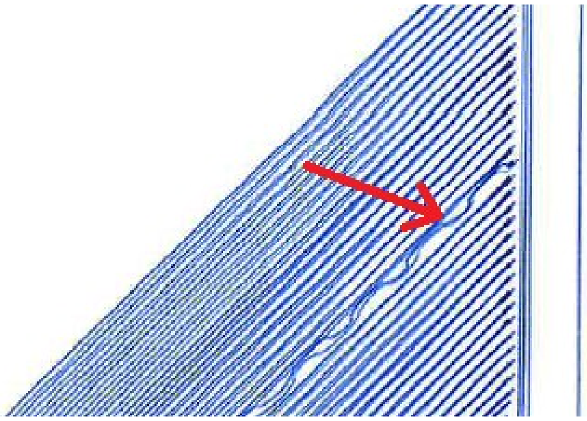


Figure 69. Deviations introduced by pushing the robot

5.2 3D printing experiment

Several test objects were printed with increasing geometry complexity and with fine-tuning the print settings.

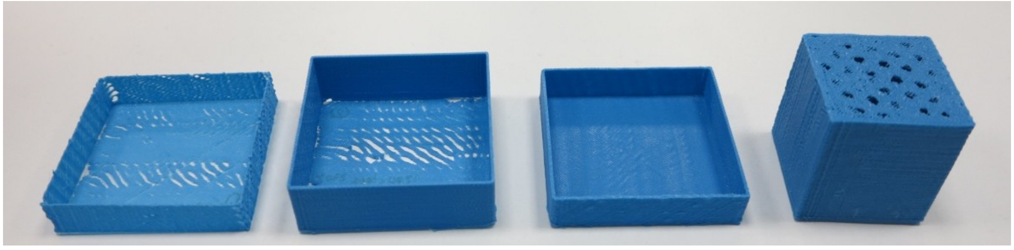


Figure 70. Test objects

The printing was successful although not as precise as on a dedicated printer.

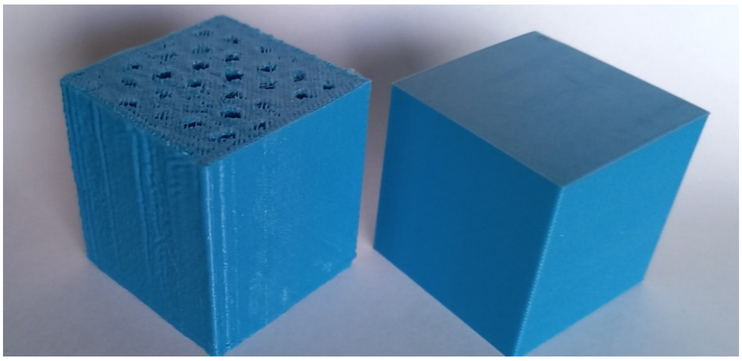


Figure 71. Left: Printed with a robot. Right: Printed on a 3D printer

Print quality also degraded in higher layers. This is likely due to error compounding.

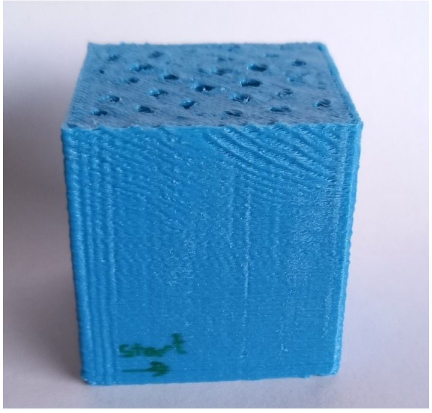


Figure 72. Quality degradation in higher layers

On all attempts stripe shaped voids appeared on the first layer. The cause for that is unclear.

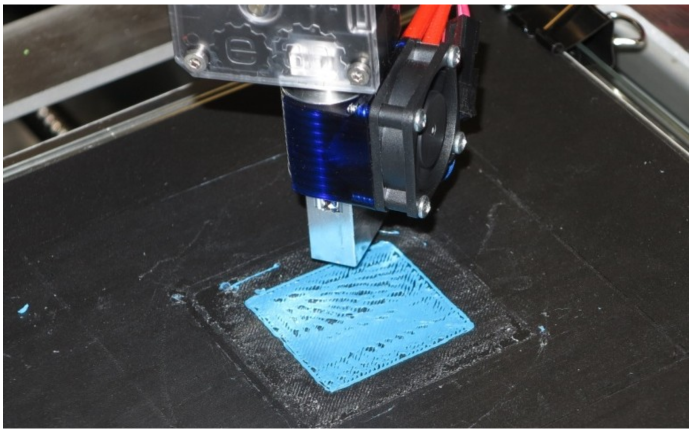


Figure 73. First layer of the print

The voids were closed after the third layer. This facilitated nominal base for the infill.

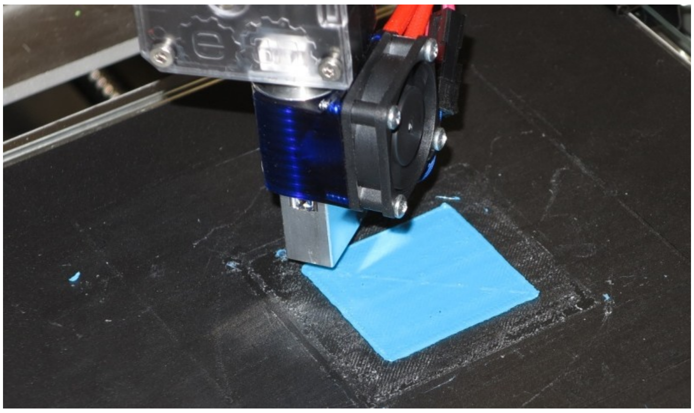


Figure 74. Third layer of the print

The infill printed mostly correctly with some voids that were the result of constant extrusion of the filament.

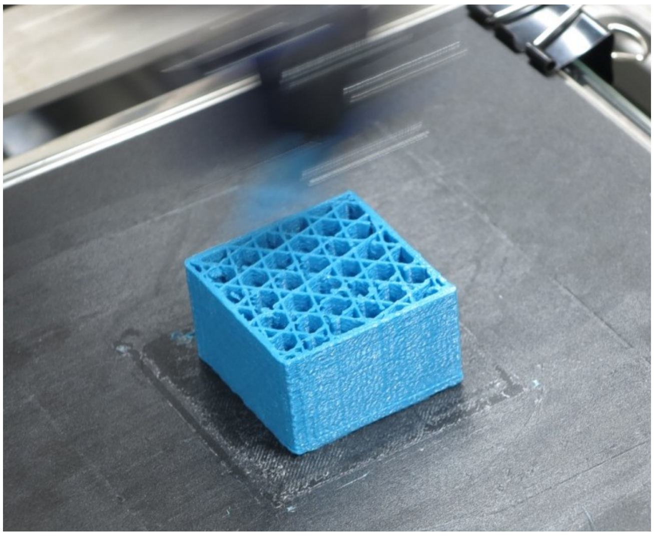


Figure 75. Midpoint of the print

The top layer of the print did not connect onto a closed surface. Also this was the result of constant extrusion and printing speeds, instead of being task specific.

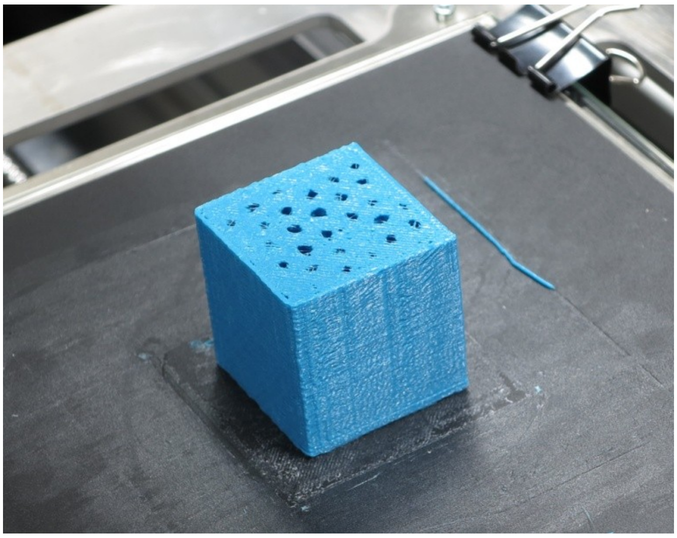


Figure 76. Finished print

6 CONCLUSION

All goals of the thesis were achieved. The final test setup ended up having a high number of components and steps. The main difficulty proved to be collecting the information needed for operating all the parts of the system. This report is hopefully going to be a valuable stepping stone for continued research by condensing all that information.

The most important upgrade to this work will be developing a way to control the extruder, as this was the main limitation of the thesis experiment. It is important to be able to use specific print and extrusion speeds for specific purposes. At the same time also the printing head control system should be simplified into a dedicated system.

Another necessary upgrade is to be able to control the robot directly from a computer. 3D printing requires a very large number of movement changes and the robot struggles to process such a long program. The other option is to edit the post-processor, to be more optimised for 3D printing.

It was observed that Universal Robots are relatively sensitive to vibrations and other interferences when this high precision is required. Although, with increasing print size, required precision usually decreases, so this might not pose a problem. One imperfection source is also the extruder mount, which should be redesigned with higher rigidity in mind.

References

Kauppila, I. (2021, May 7*). Robotic Arm 3D Printing – Platforms & Software.* All3DP. Retrieved May 9, 2022, from https://all3dp.com/1/robotic-arm-3d-printing-platforms-software/

Relativity Space. (n.d.). *Factory of the Future.* Retrieved May 9, 2022, from https://www.relativityspace.com/stargate

Schwaar, C. (2022, March 24). WAAM! *What Is Wire Arc Additive Manufacturing?*. All3DP. Retrieved May 9, 2022, from https://all3dp.com/1/waam-what-is-wire-arc-additive-manufacturing/

Nagami. (n.d.). *The Throne.* Retrieved May 9, 2022, from https://nagami.design/en/project/the-throne/

Stevenson, K. (2019, December 25). *What's Up At MX3D?.* Fabbaloo. Retrieved May 9, 2022, from https://www.fabbaloo.com/2019/12/whats-up-at-mx3d

Branch. (n.d.). *Products.* Retrieved May p, 2022, from https://branchtechnology.com/products/

3HTi. (n.d.). *EnvisionTEC Viridis 3D Printer: Create Custom Castings for Metal Foundries.* Retrieved May 9, 2022, from https://3hti.com/viridis-3d-printer/

Mensley, M. (2021, November 18). *3D Printer Extruder – The Ultimate Guide.* All3DP. Retrieved May 10, 2022, from https://all3dp.com/1/3d-printer-extruder-nozzle-guide/

Budmen. (n.d.). *3D Printing Crash Course.* Retrieved May 10, 2022, from https://budmen.com/support/3d-course/lesson-2/

MakeItFrom. (n.d.). *PLA vs. PTFE.* Retrieved May 10, 2022, from https://www.makeitfrom.com/compare/Polylactic-Acid-PLA-Polylactide/Polytetrafluoroethylene-PTFE

RoboDK. (n.d.). *Simulate Robot Applications.* Retrieved May 23, 2022, from https://robodk.com/

Additive-X. (2021, September 16). *What is Slicing Software, and what does it do?*. Retrieved May 10, 2022, from https://www.additive-x.com/blog/what-is-slicing-software-and-what-does-it-do/

RepRap. (2022, May 10). *G-code.* Retrieved May 24, 2022, from https://reprap.org/wiki/G-code

Slic3r. (n.d.). *About.* Retrieved May 23, 2022 from https://slic3r.org/about/

BuildTak 3D. (n.d.). *BuildTak Original 3D Printing Surface – Black*. Retrieved May 9, 2022, from https://buildtak.eu/collections/buildtak-products/products/buildtak-original-3d-printing-surface-25-pack-black

FileInfo. (2022, April 29). *.STEP File Extension.* Retrieved May 9, 2022, from https://fileinfo.com/extension/step

Pololu. (n.d.). *A4988 Stepper Motor Driver Carrier.* Retrieved May 3, 2022, from https://www.pololu.com/product/1182

Joy-IT. (n.d.). *Joy-IT Mega2560R3.* Retrieved May 4, 2022, from https://joy-it.net/en/products/ARD-Mega2560R3

E3D. (n.d.-a). *Volcano HotEnd*. Retrieved May 5, 2022, from https://e3d-online.com/products/volcano-hotend

E3D. (n.d.-b). *Titan Extruder.* Retrieved May 5, 2022, from https://e3d-online.com/collections/extruders/products/titan-extruder

E3D. (n.d.-c). *Motors.* Retrieved May 5, 2022, from https://e3d-online.com/products/motors

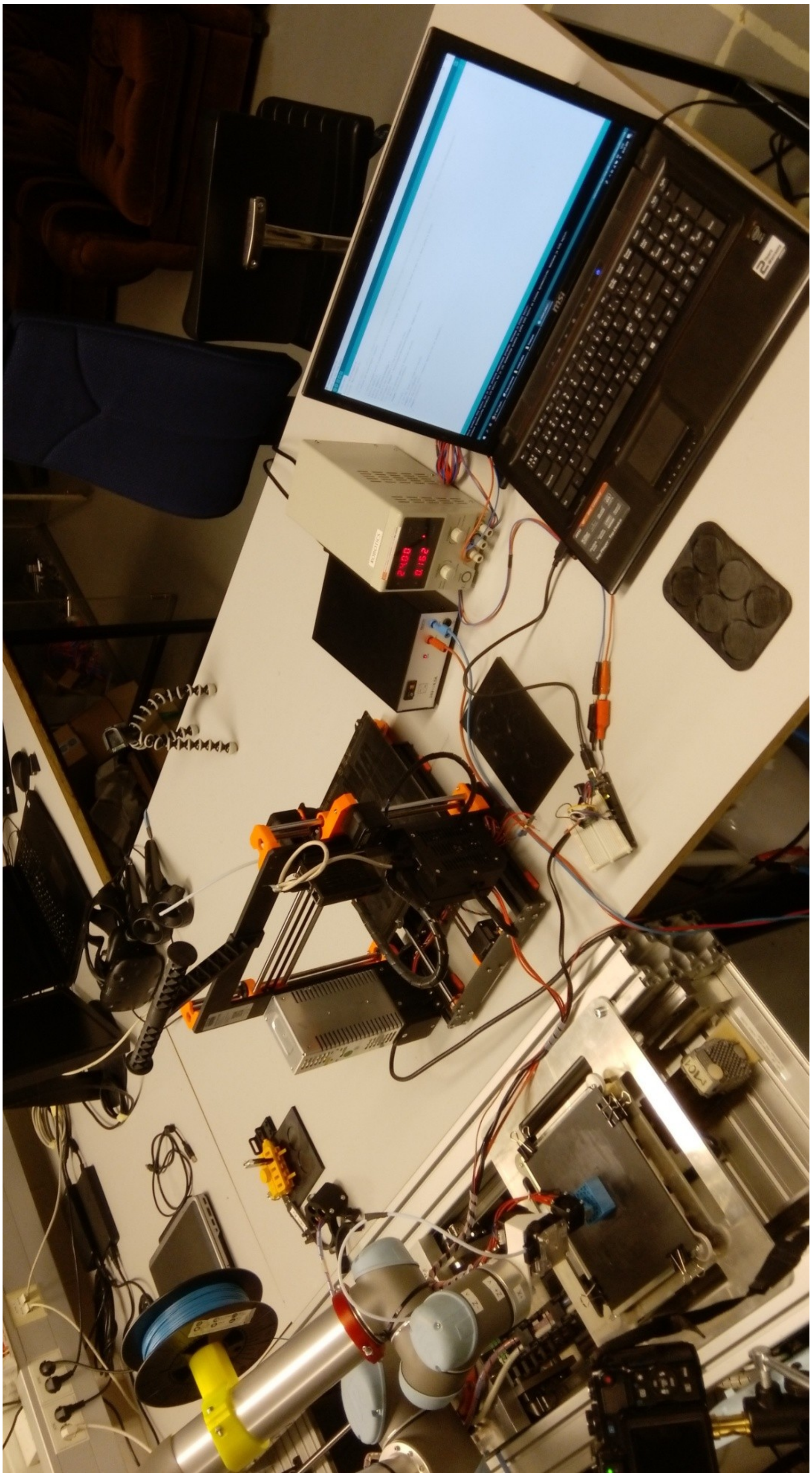
Marlin. (n.d.). *G-code Index*. Retrieved May 5, 2022, from https://marlinfw.org/meta/gcode/

Universal Robots (n.d.-a). *Downlaod.* Retrieved on May 18, 2022, from https://www.universal-robots.com/download/?filters[]=98759&query=

Universal Robots (n.d.-a). *Magic File - Upload All Programs from USB to robot.* Retrieved on May 18, 2022, from https://www.universal-robots.com/download/software-e-series/support/magic-files/magic-file-upload-all-programs-from-usb-to-robot/

Lutz, W. (2020, July 23). *3D Printing with the Prusa MK3 and Slic3r software.* MIT IDC. Retrieved May 12, 2022, from https://idc.dozuki.com/Guide/3D+Printing+with+the+Prusa+MK3+and+Slic3r+software/4

Appendix 1: Experiment setup



Appendix 2: Arduino program for running the extruder motor

```
#include <AccelStepper.h> // Include the AccelStepper library.Download from
           // https://www.airspayce.com/mikem/arduino/AccelStepper/
#include <MultiStepper.h> // Include the MultiStepper library. Included in
                          // AccelStepper

    // Define stepper motor connections and motor interface type.
    // Motor interface type must be set to 1 when using a driver:
    #define dirPin 3 // Rotation direction is on pin 3
    #define stepPin 4 // Step is on pin 4. Each signal to "STEP" is
                      // going to move the motor by one step.
    #define motorInterfaceType 1
    // Create a new instance of the AccelStepper class:
    AccelStepper stepper = AccelStepper(motorInterfaceType, stepPin,
dirPin);
    void setup() {
      // Set the maximum speed in steps per second:
      stepper.setMaxSpeed(20000);
    }
    void loop() {
      // Set the speed in steps per second:
      stepper.setSpeed(1200);
      // Step the motor with a constant speed as set by setSpeed():
      stepper.runSpeed(); // Starts the motor. To stop it, unplug the usb
    }
```

Appendix 3: Extrusion speed calculation

One step of the motor is 0.9 degrees
To complete one full rotation, it takes 400 steps
To increase smoothness we use 1/16 steps
To complete one full rotation it therefore takes 6400 steps

Extrusion speed (from Slic3r):

$L = 1.61$ m
$t = 807$ s
$v = \frac{L}{t} = 0.002$ m/s

Hob diameter: $D = 0.008$ m

Frequency of rotation of the large gear: $f_L = \frac{v}{\pi \cdot D} = 0.0796$ Hz

Gear ratio = 1:3

Frequency of rotation of the small gear: $f_S = 3 \cdot f_L = 0.2389$ Hz

Number of steps per second to achieve the desired speed: $N = 6400 \cdot f_S = 1529$ /s

Appendix 4: GCode program for heating the hotend

```
;Because the printer will timeout after a certain time of inactivity,
;the command blocks have to be repeated.
;The program is 30 minutes long. Add extra repetitions for longer prints.

M104 S230              ;Set temperature to 230°C
M109 S230              ;Wait for temperature to be 230°C
M300 S440 P200         ;Play tone, S defines tone frequency [Hz], P defines duration [ms]
M300 S660 P250
M300 S880 P300
G4 S600                ;Wait 600 seconds
M104 S228
M109 S228
M104 S230
M109 S230
M300 S440 P200
M300 S660 P250
M300 S880 P300
G4 S600
M104 S228
M109 S228
M104 S230
M109 S230
M300 S440 P200
M300 S660 P250
M300 S880 P300
G4 S600
M300 S440 P200         ;Longer alarm 1 min before the program ends
M300 S660 P250
M300 S880 P300
M300 S440 P200
M300 S660 P250
M300 S880 P300
M300 S440 P200
M300 S660 P250
M300 S880 P300
M300 S440 P200
M300 S660 P250
M300 S880 P300
M300 S440 P200
M300 S660 P250
M300 S880 P300
G4 60
M104 S0                ;Turns off the hotend
```