



Design 9  
TPD4500

Daniel Fremming  
Sebastian Oliver Dahl  
Fall 2023

# Design for Robotic Fabrication

Industrial design, 5th year  
NTNU, Trondheim



## Abstract

The project aims to investigate and demonstrate the value of robotic fabrication as a design and fabrication tool. The project can be divided into three phases. The first being calibration and communication. By ensuring that the robot is properly set up and communicating some basic operation procedures, we hope to leave behind information that will make the particular robot that was used more approachable for students in the future.

By introducing a new software approach that has not been used previously in the Lukas workshop, we have simplified the process of generating accurate RAPID code. This approach is demonstrated with robotically fabricated prototypes. We also developed a surface topology that is constrained by the hemispherical geometry of the ball mill.

Finally, we have created vases that mirror our state-of-the-art reference, the AESTUS VASES by Oliver David Krieg. We achieved this by mimicking the wavelike surface patterns as well as mixing them with our own original surface topologies. The fabrication of these vases required that we employed the positioner that has, to our knowledge, not previously been used in the Lukas Workshop.

**No 0.**

# **Introduction**

The background and point of departure of the project,  
along with the ambition and motivation.

**Design for Robotic Fabrication**  
Wooden vase project

**Table of contents**

<b>No. 0</b> 00-08	<b>Introduction</b> Context and reference
<b>No. 1</b> 09 - 16	<b>Hardware setup</b> Robot system
<b>No. 2</b> 17- 26	<b>Software setup</b> Digital setup and surface topologies
<b>No. 3</b> 26 - 44	<b>Vases</b> Vase designs and robotic milling

## Collaborative parts

Project supervisor  
**Jóhannes Sigurjónsson**  
Department of Design  
at NTNU

Associate Professor and former Study Program Leader at the Department of Design, Faculty of Architecture and Design at NTNU, Trondheim, Norway.

---

Co-supervisor  
**Lars Tingelstad**  
Department of  
Mechanical and  
Industrial Engineering  
at NTNU

Lars Tingelstad works as an Associate Professor of Robotic Production at the Department of Mechanical and Industrial Engineering at NTNU, Trondheim, Norway. He is the leader of the Robotics and Automation research group.

---

Project advisor  
**Ricardo Mayor**  
Fabrication expert /  
MAA Faculty at IAAC

Ricardo Mayor Luque is a Master in Architecture Superior Technical School of Architecture of Alcalá de Henares in Madrid and a Master Degree in Advanced Architecture from IAAC (Spain) with a specialization in digital and robotic fabrication.

## Students



**Daniel Fremming**  
Industrial design engineering at NTNU, 5th year

Interaction design student with background in computer science.  
Key interests: AI, Technology, GUI, Human-Computer interaction



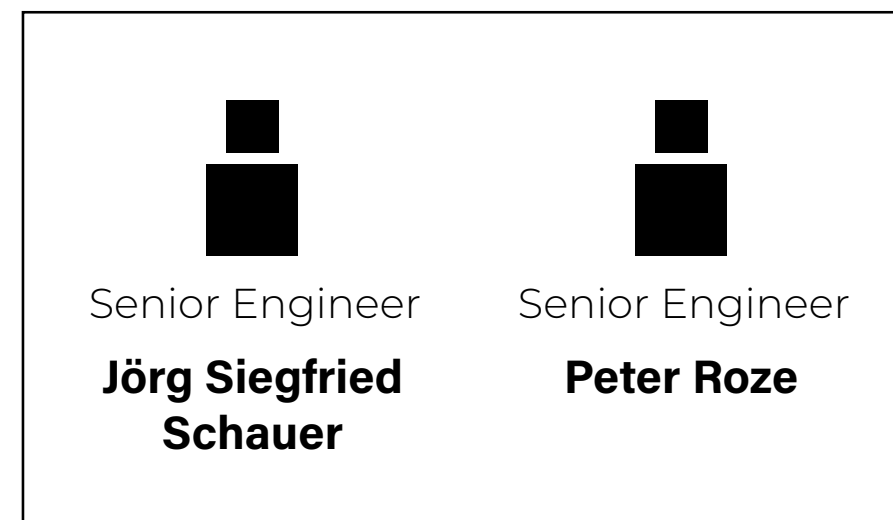
**Sebastian Oliver Dahl**  
Industrial design engineering at NTNU, 5th year

Design student specializing in computational design with a background in mechanical engineering. Graduated from IAAC, Barcelona during spring of 2023.

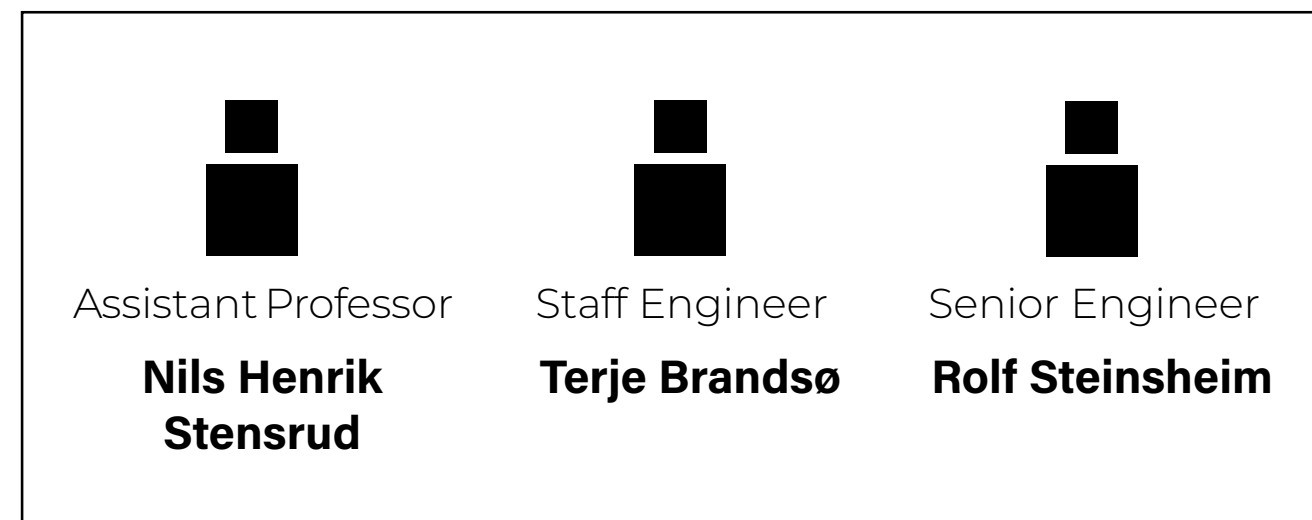
## Supportive network

This project was supported technically by these specialists. We remain thankful for their efforts.

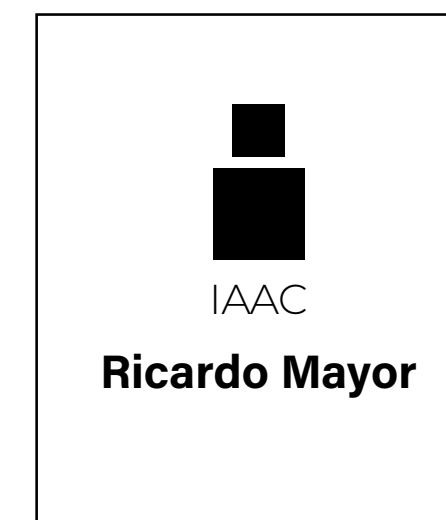
Lukas Workshop (NTNU)



Design Workshop (NTNU)



Robotic fabrication software support



Grasshopper support (NTNU)

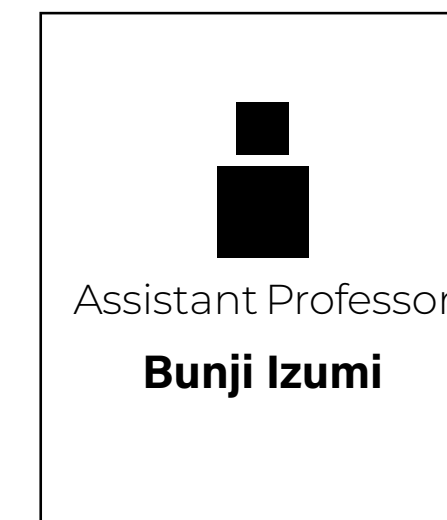


ABB robot hardware support (NTNU)

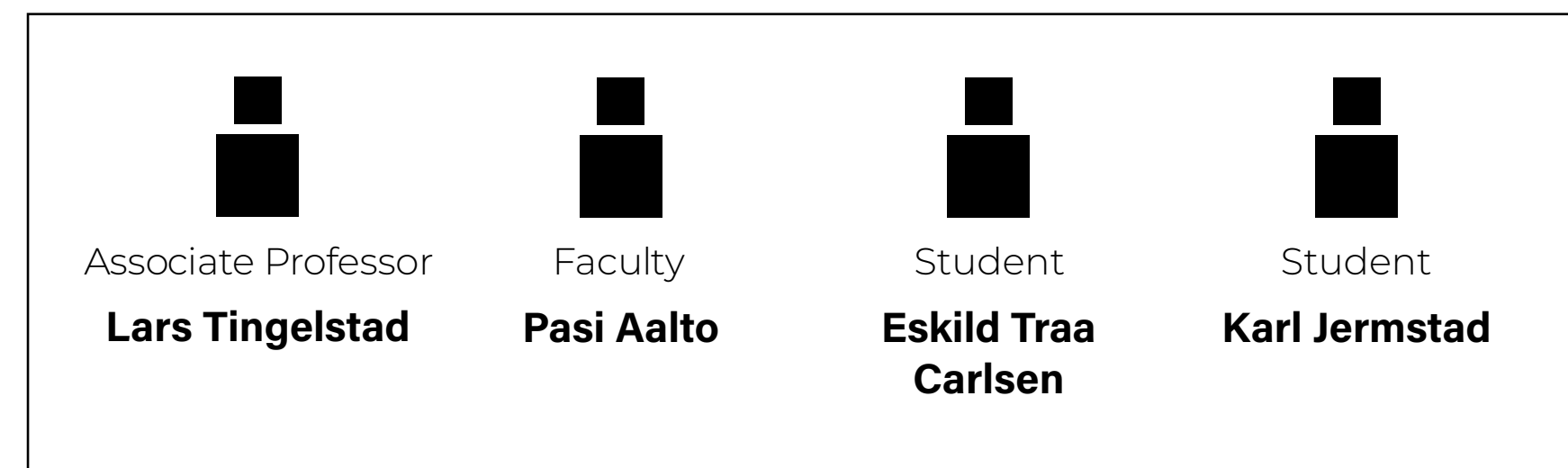
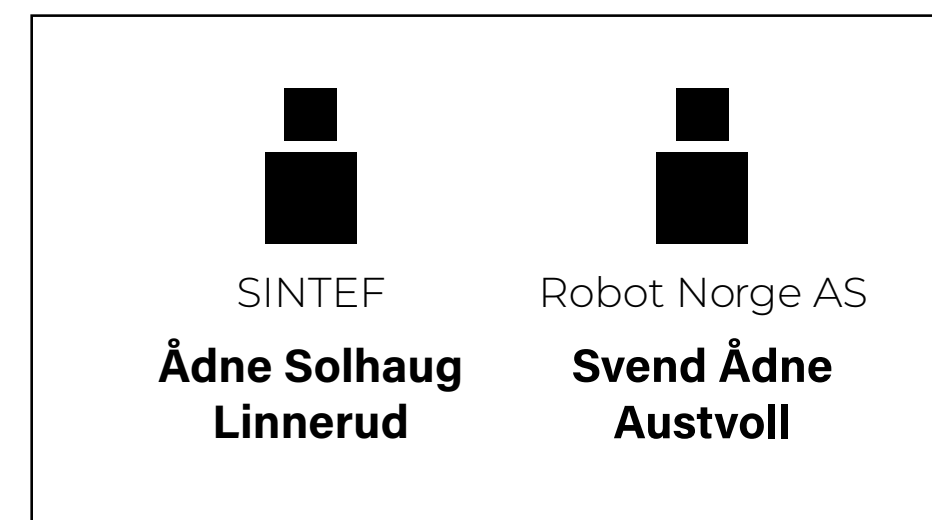
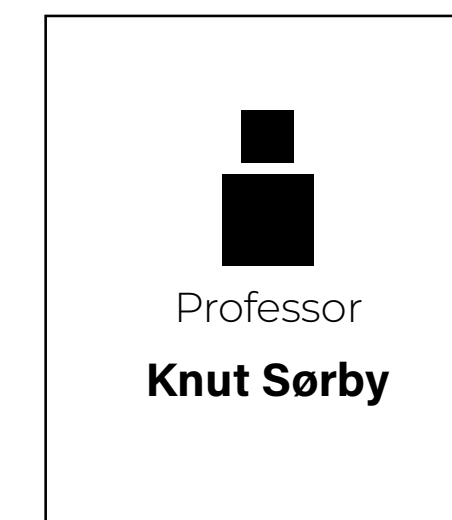


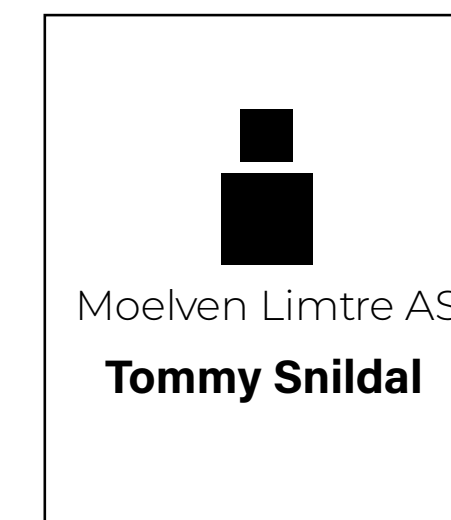
ABB robot hardware support



Milling head expertise (NTNU)



Provider of leftover materials



## Reference

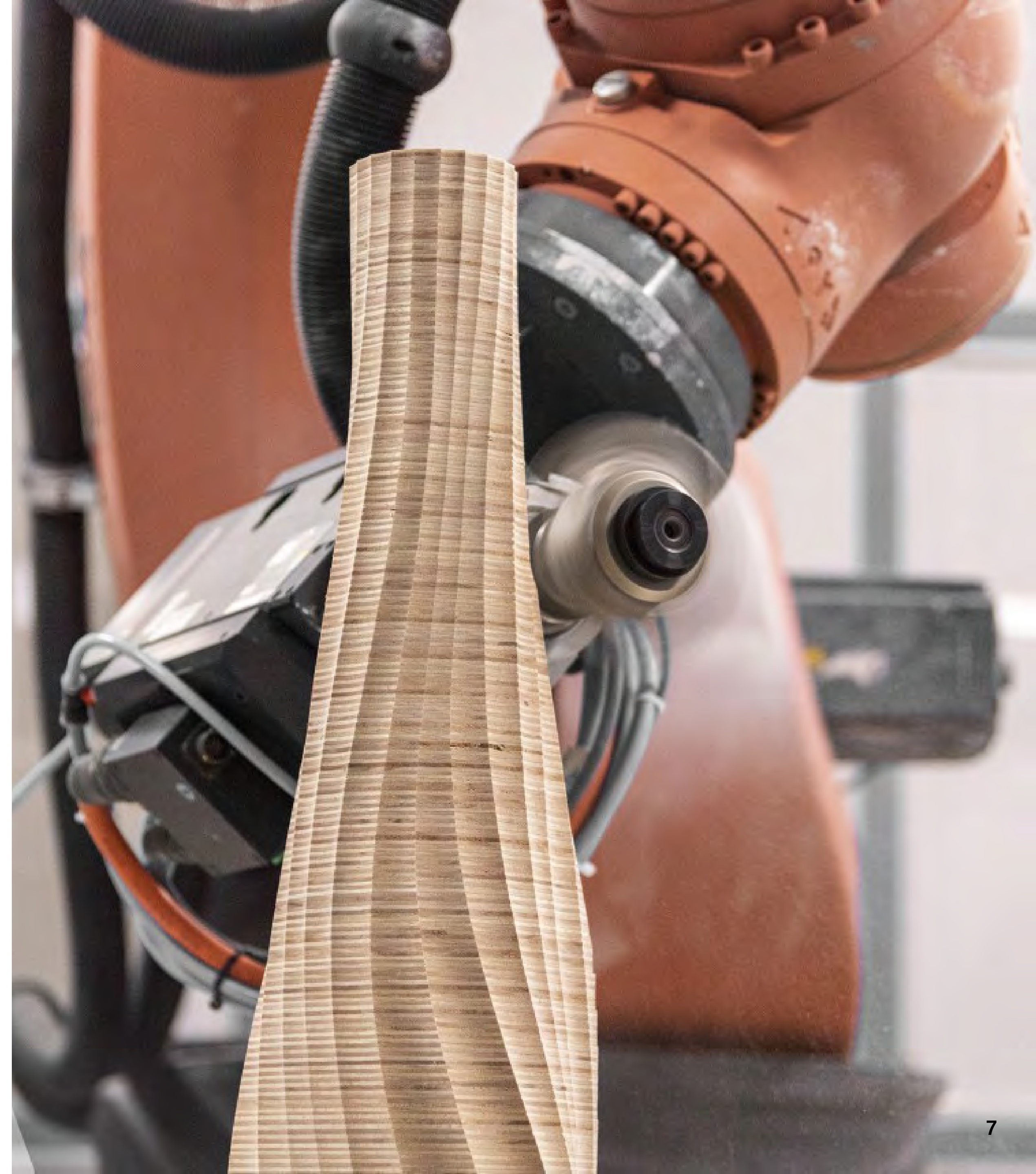
### AESTUS VASES

AESTUS is the first product design project from German architecture and wood researcher Oliver David Krieg. Building on almost ten years of experience in digital manufacturing of complex wood structures, AESTUS seeks to express the forces of formation in robotic fabrication, and to explore the dynamics of human, material and machine.

Carved from hundreds of layers of wood by an industrial robot, the vases capture the fluidity of the machine's movements in the depth of the wooden texture.



**Project Year** : 2018  
**Designed by** : Oliver David Krieg  
**Location** : Canada



**The ambition of the project was to apply robotic fabrication as a design tool, and in doing this familiarize the process of employing an industrial 6-axis arm for creative purposes.**

**Knowing that the AESTUS Vases were fabricated by professionals with skill levels way beyond our own, we decided that it would be a sufficient challenge to replecate their efforts.**



# No 1.

# Hardware setup

This part explains the setup of the tools and also provides a walkthrough of how to operate them.

## ABB IRB 4600

ABB Ltd. is a Swedish-Swiss multinational corporation headquartered in Västerås, Sweden, and Zürich, Switzerland. It is one of the major manufacturers of industrial robots globally.

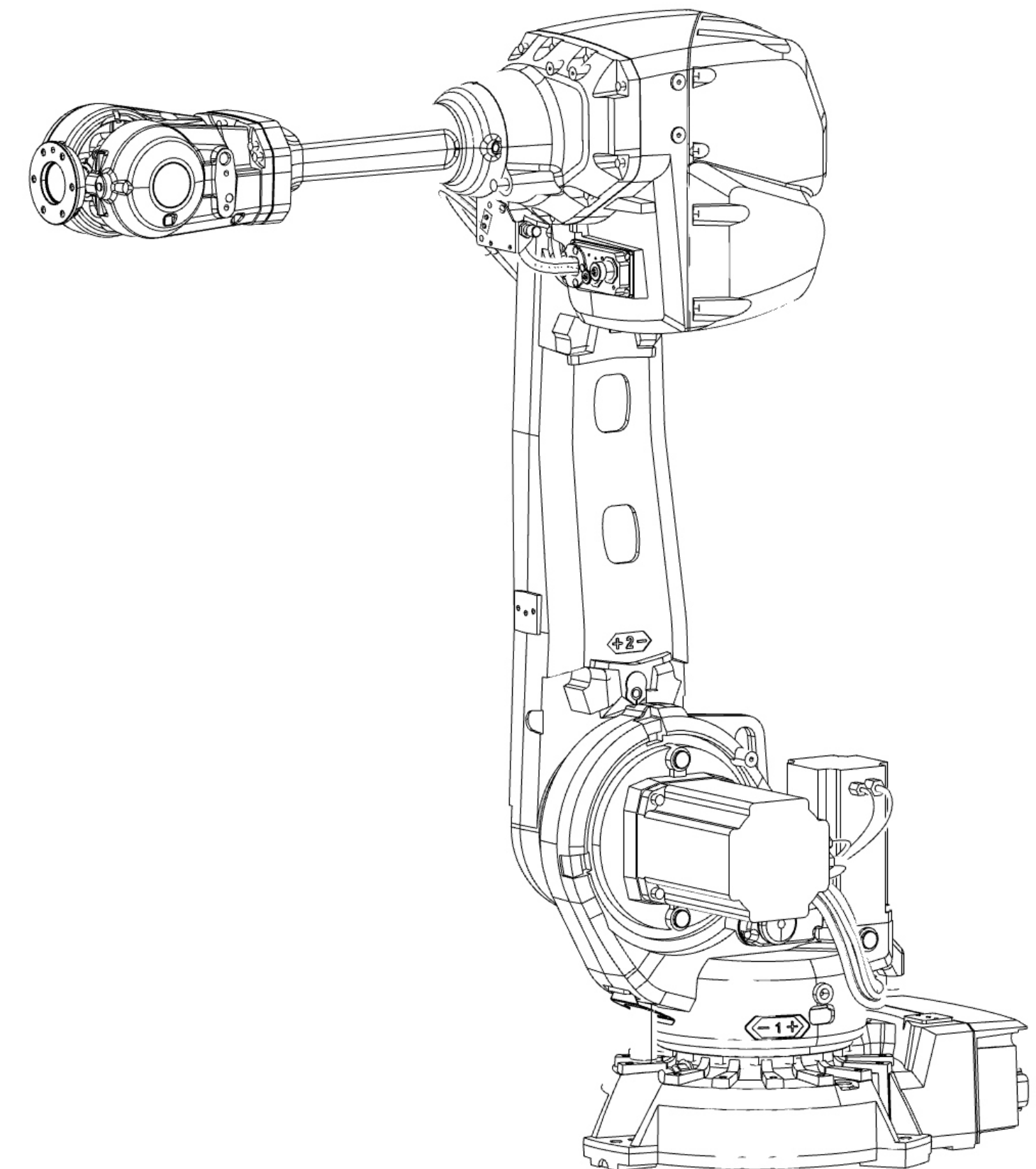
The ABB IRB 4600 is a highly productive general purpose robot optimized for short cycle times where compact robots can help create high density cells. While the automotive industry is a primary market for the IRB 4600 series, these robots have also been used for creative purposes by places like ETH Zurich's Institute of Technology in Architecture, one of the leading places for robotic fabrication in architecture.

### Main applications

- Arc Welding
- Assembly
- Material Handling
- Machine Tending
- Material Removal
- Cleaning/Spraying
- Dispensing
- Packing
- Laser Cutting
- Laser Welding

### Variants

- IRB 4600-60/2.05
- IRB 4600-45/2.05
- IRB 4600-40/2.55
- IRB 4600-20/2.50



## ABB IRB 4600-60/2.05 setup

The specific robot used in our project was the ABB IRB4600 60 kg 2.05 version 6-axis industrial robot. The lab is located in the Lukas workshop in Tapirbygget at NTNU in Trondheim. An IRBP L-300 positioner is located opposite a support axis. The robot has a spindle mounted on the 6th axis. The robot, the external axis and the spindle are all connected to the IRC5 Controller which is enabling them to act together

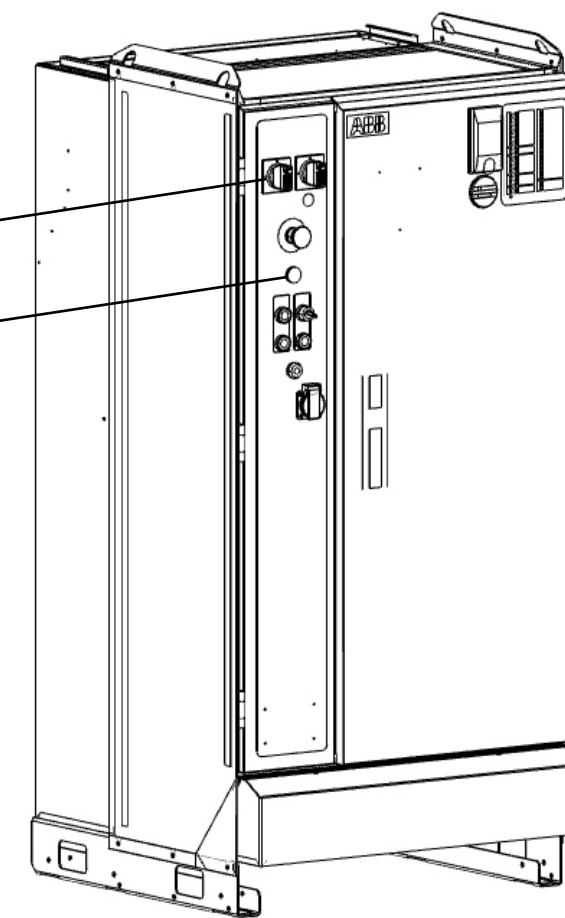
### Setup

- IRB 4600-60/2.05 robot
- IRBP L-300 positioner + support
- IRC5 Controller

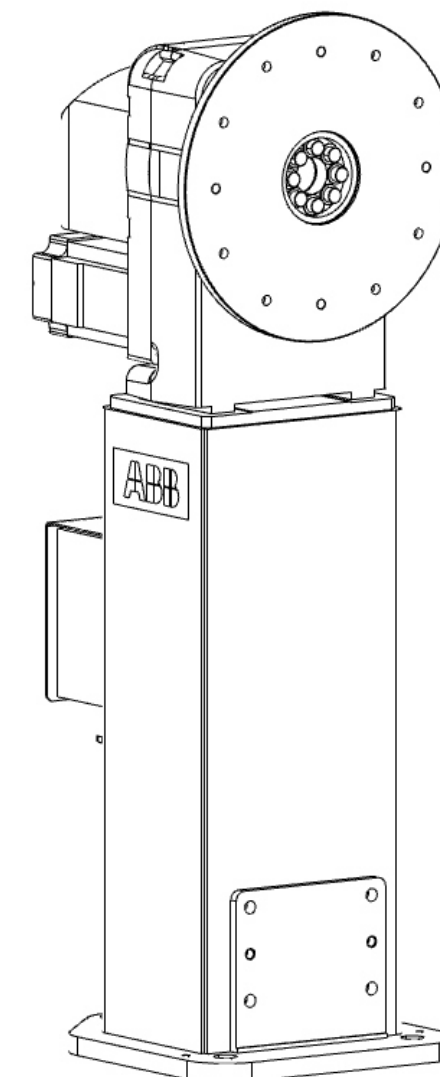
**Location:** Lukas workshop  
NTNU, Trondheim

### Startup

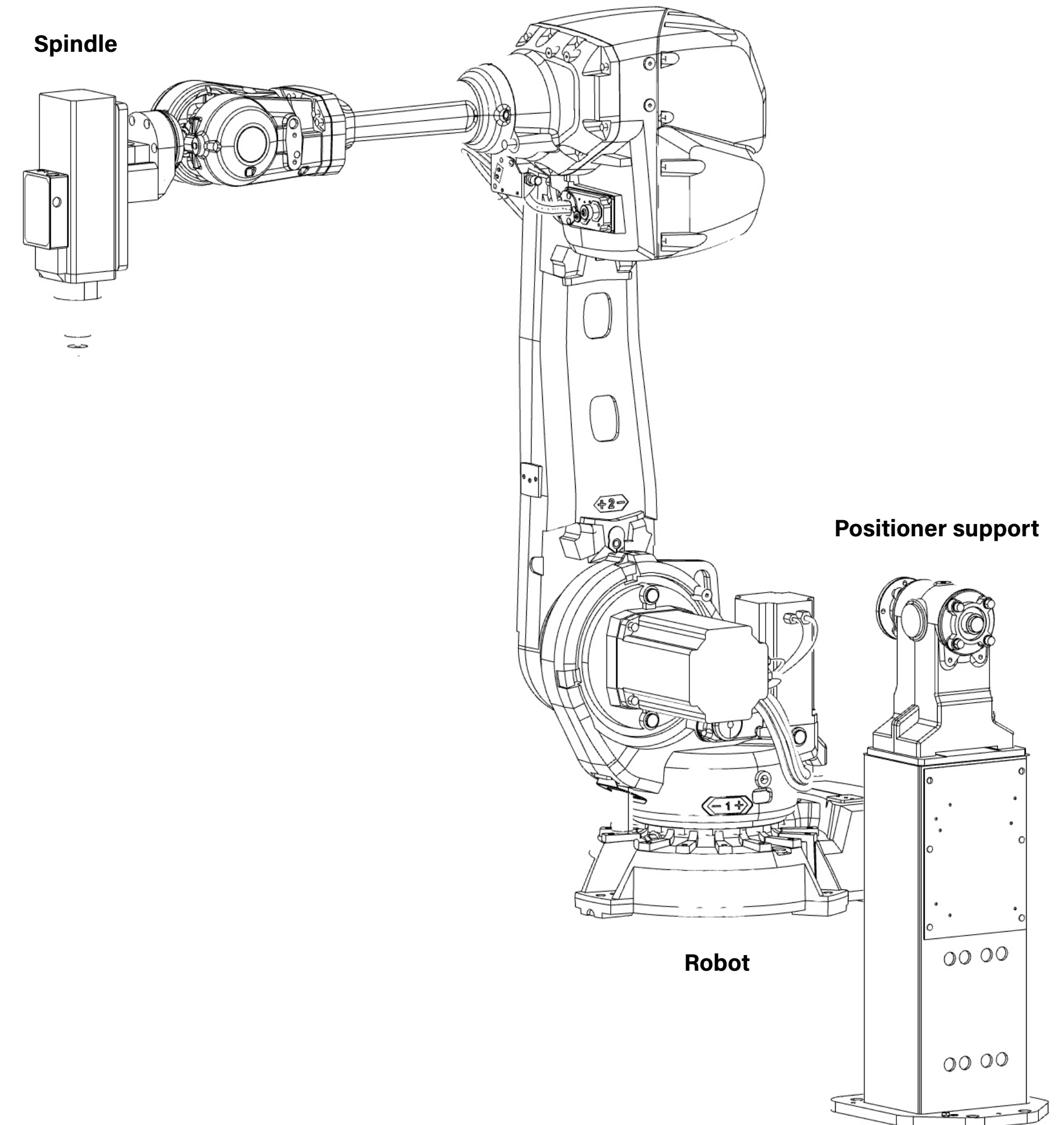
1. Flick this switch
2. Push this white pulsing button
3. Wait for the the startup procedure to complete
4. Grab the Flexpendant and start interacting with the robot

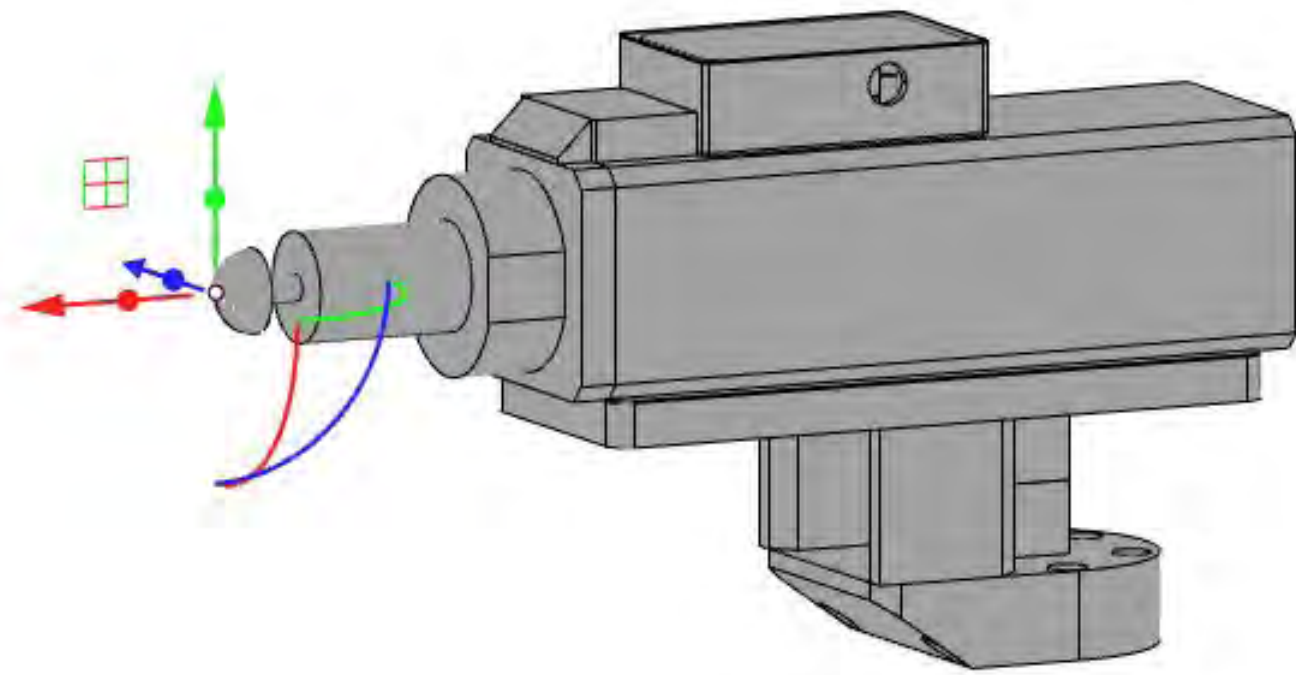


**IRC5 Controller**



**Positioner**





**Tool center point (TCP):** The TCP is located at the tip of the tool that is mounted on the spindle. This exact point is essential for generating the correct RAPID scripts, and the orientation of the x-, y- and z-axis also matter.

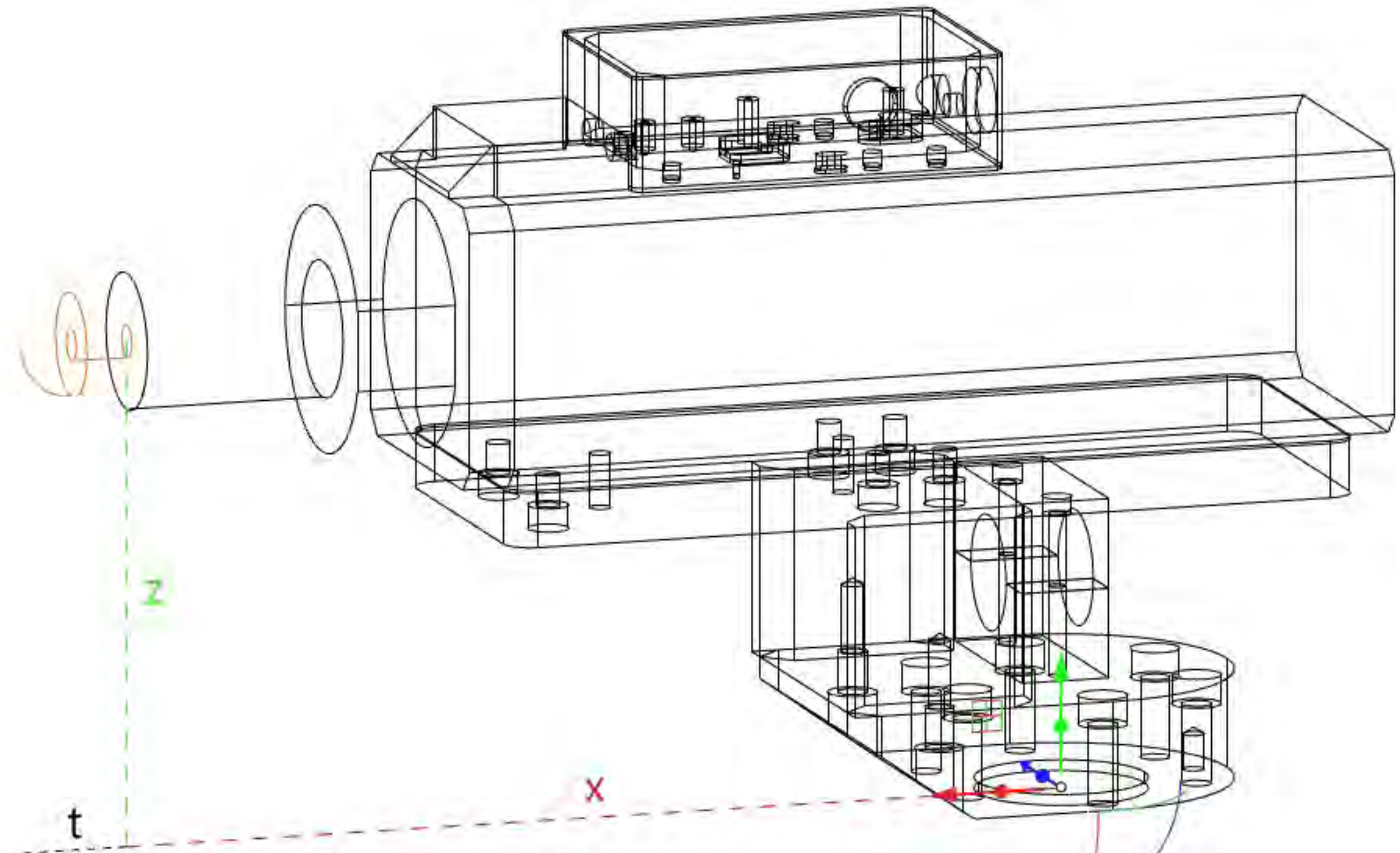
## Tool center point (TCP)

TCP: [353.50+t, 0.00, 186.12]

The TCP plane of the tool needs to hold the correct corresponding values with the physical tool in the workshop in order for the robot to work as expected. It is always located in relation to origo at [0,0,0]. The spindle geometry CAD file is precise and provides a fairly correct starting point. The z-value will not change no matter what kind of millbit you attach, because the TCP will always be at the center of the spinning cylinder on the spindle. The z-value must, in this case, always be  $z=186.12$ . The y-value will not change either, as the spindle is placed symmetrically on the robot, thus the y-value will always be  $y=0$ . However, although the x-value has a base value of  $x=353.5$  the total value will depend on the length of the tool:  $t$ , that is attached. In the case of our ball mill tool we have a total x-value of  $x=393.5$

## Spindle geometry

The spindle geometry has been changed from the original CAD in order to match the physical tool. The upper part of the spindle was originally flipped 180 degrees, but this was changed by previous users of the arm as an effort to reduce error margins. Although it is not vital that the spindle geometry matches the TCP in order to make RAPID scripts, it will make the visualized movement in the simulation correspond much more with the real movement. Thus, while it is possible to create RAPID code with only the TCP-plane as a tool-input, it is always preferable to also attach the tool geometry



## Tool calibration

In order to confirm that the TCP-coordinates are correct, a calibration procedure is necessary. There are two categories of calibration that are possible to conduct. One is zero correction and the other is tool calibration. L. Tingelstad, our technical advisor, could read from the robot that the last zero correction was performed by ABB technicians, and therefore advised against any zero correction calibration. Thus we decided to only conduct tool calibration

We jogged the robot into a position, in our case 100 mm above the fixed table that is placed in the lab, and while in this position we could read an approximate x-, y- and z-values from the flexpendant. We marked this point with a wooden stick that would easily break if the robot went too far. Then we created scripts where the tool would approach this point from different angles. Initially our TCP-values were quite off, but after adjusting the x-value up and down we reduced the error margins to acceptable levels.

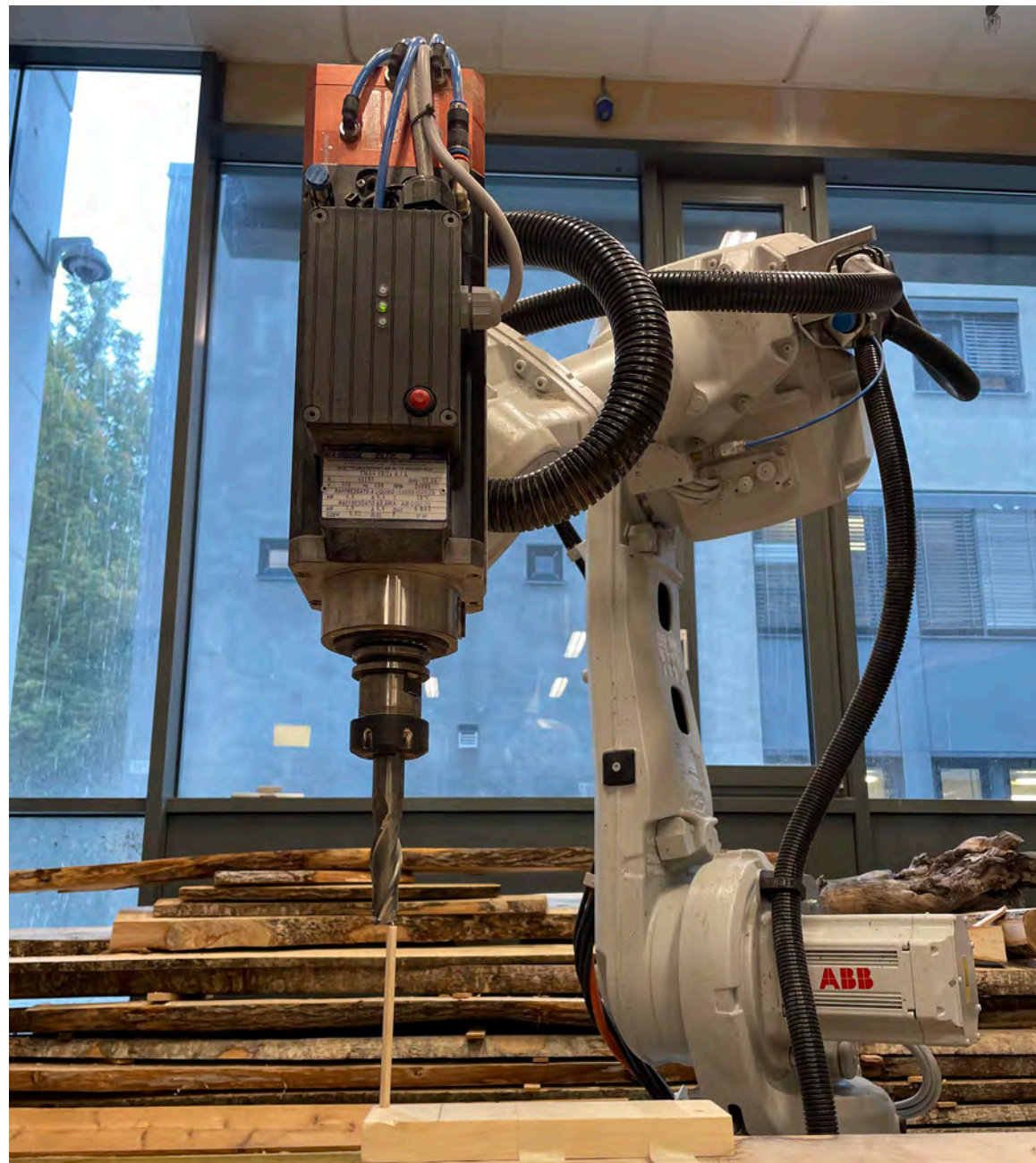


Figure 1: Approach from above

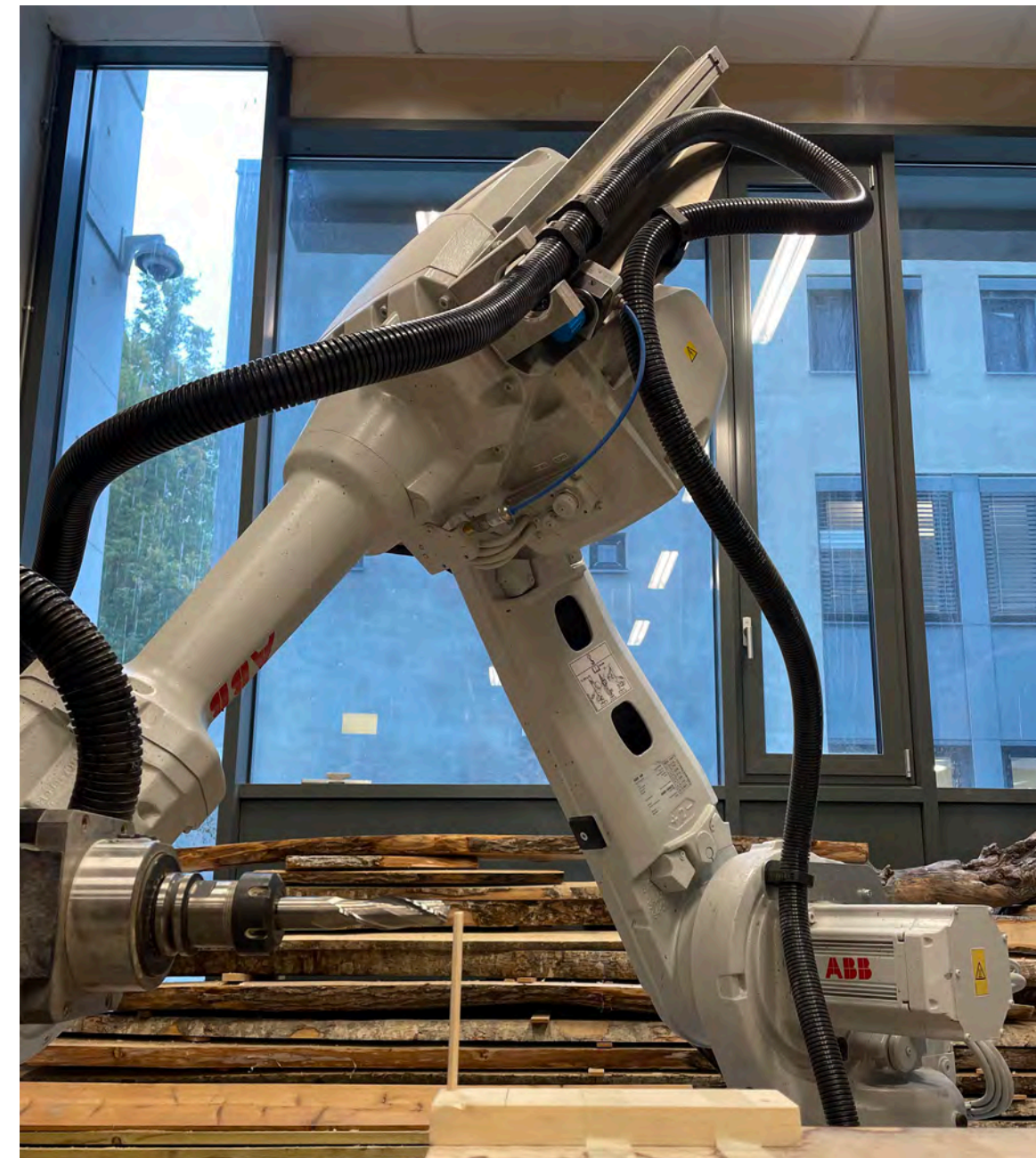


Figure 2: Approach from the left.



Figure 3: Approach from behind.



Figure 4: Sideways approach from above.

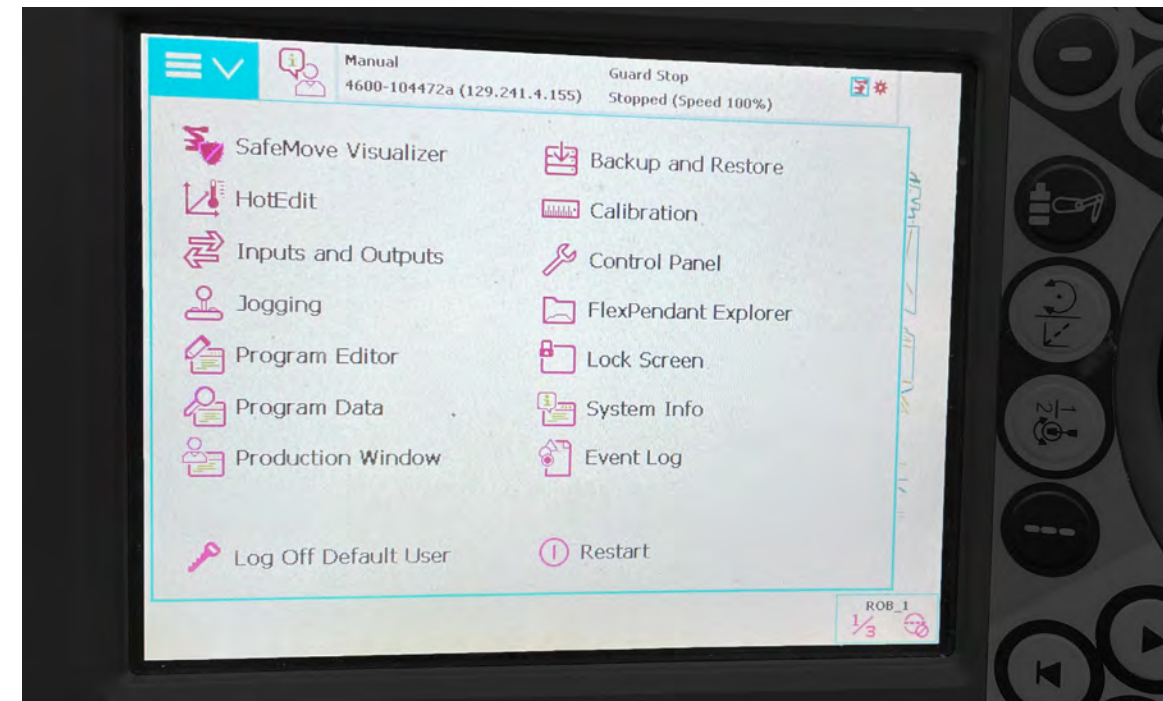
# Flexpendant walkthrough

The flexpendant is the external screen that is connected to the IRC5 Controller which is used to control the robot. Here are some basic procedures.

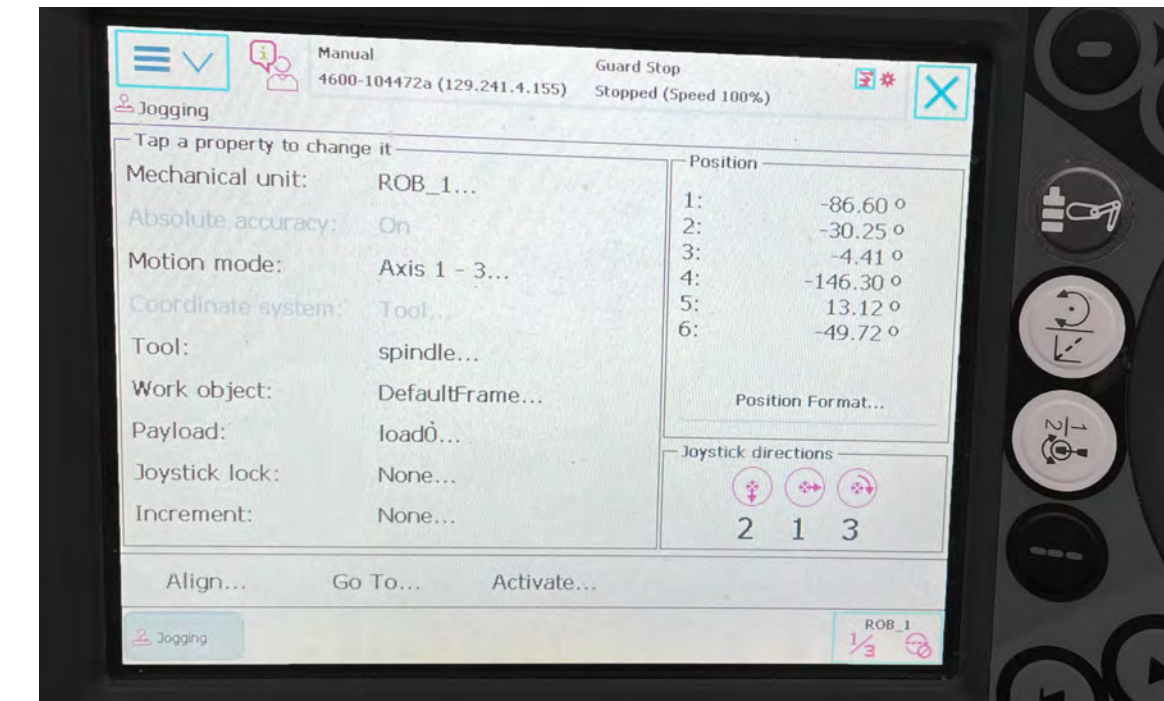
**Note:** The long button behind the joystick must be lightly, but firmly pressed in order for the robot to run. It is the "manual mode" for industrial robots, and whenever the button is released the power gets cut and the robot halts.



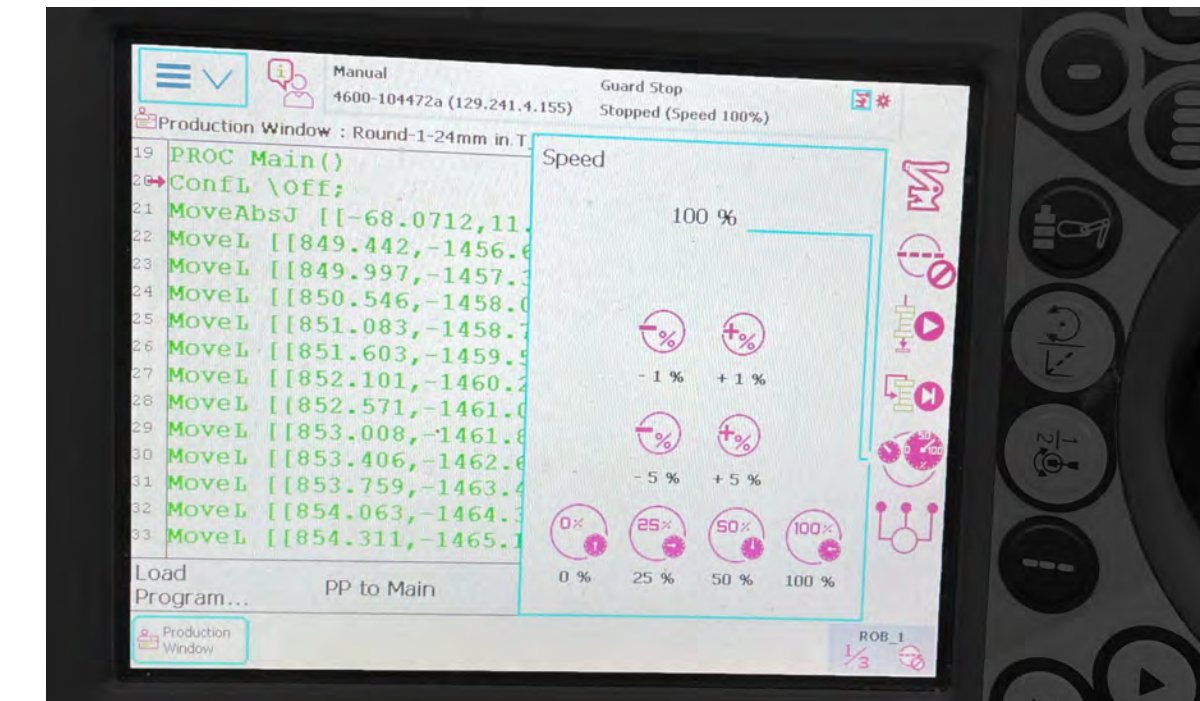
**Starting screen:** Press the menu in the upper corner



**Main menu:** This is the main menu



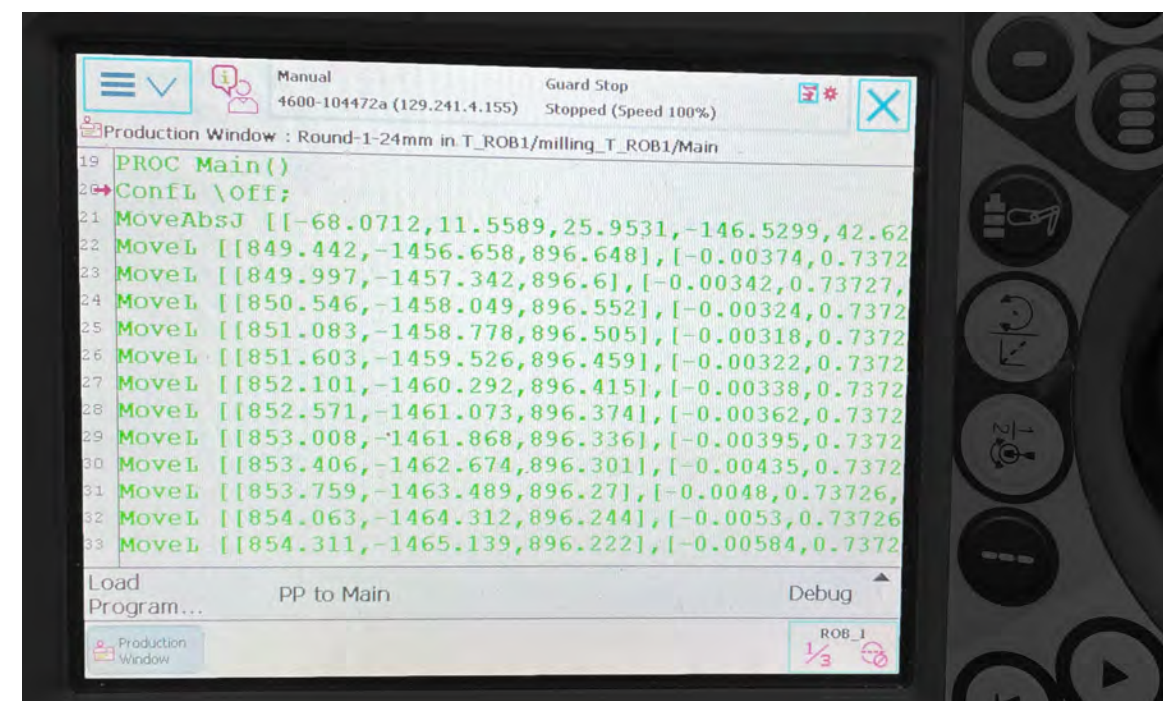
**Jogging:** Jog the robot using the joystick and switch between different axis using the three middle buttons.



**Control robot speed:** The operational speed of the robot can be altered during running of the code.

## Running scripts

Scripts are run by selecting "Production window" from the main menu. Then select "Load program" and respond "Yes" on the pop-up window that appears. Next select ".prg" in the dropdown menu and find your file from your connected USB stick. Alternatively you can upload the script via ABB Robotstudio as long as a computer is synced with the IRC5 Controller.

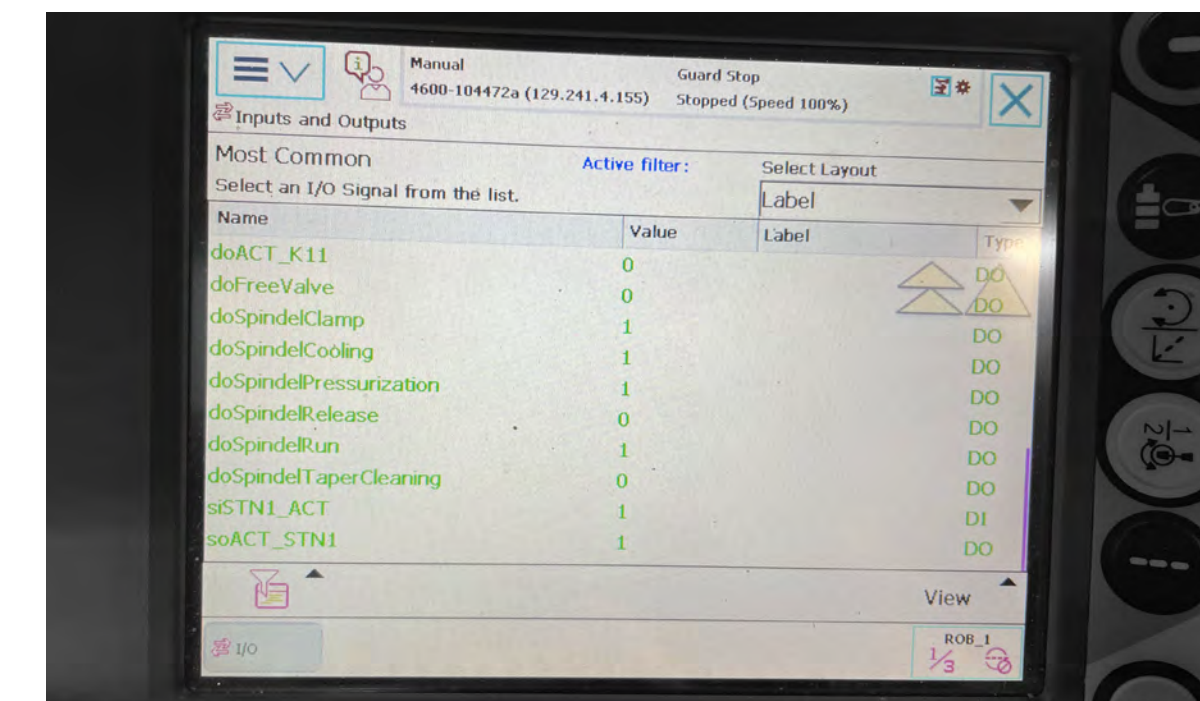


## Activating spindle

The I/O switches controls the spindle. First, set the spindle speed (we have used 15-20 000 rpm). Then change only the following values from 0 to 1:

- doSpindleClamp **1**
- doSpindleCooling **1**
- doSpindlePressurization **1**
- doSpindleRun **1**

Make sure doSpindleRelease is set to **zero**.





## State of the robotic lab at the Lukas workshop

The space around the robot tends to get somewhat cluttered due to rare use. The rare use also results in delusion of competency and specific knowledge regarding the operating procedures of this particular robotic arm. We hope to leave some of the knowledge we have obtained in a way so that it will be an easier and faster process for others who will use the lab in the future

Certain errors that arise can lead to frustration and may halt all progress for periods of time. For instance, before we were able to run any kind of RAPID code, we needed to fix an error that had appeared inside the IRC5 system code. In this case there was a typo "spindl" in the code that needed to be corrected to "spindle". The process of solving this one issue took ~ 3 weeks and we would have been unable to do it without help from Ådne Linnerud (SINTEF).

## Insight - Point of departure

Starting out with this project we had no clue as to how we could operate the robot arm in the workshop and what to make out of the setup there. Therefore, in phase 1 getting in touch with people that could show us the ropes was key for our project to continue. This was done through correspondences and meetings with key people, either providing us with new information or redirecting us to other people of interest. Incrementally, as our network of people expanded so did our knowledge. In parallel with the outreach we kept track of new acquired knowledge and how it can be used for our project.

Establishing a network of people took first priority since we have never done anything like this before and felt like we were in deep waters, therefore, we thought, it would be nice to have a network of people to reach out to if we were stuck or unsure about some of the problems we were facing. Phase 1 includes therefore an overview of people we deem necessary to be aware of next time executing a robot fabrication project. In addition, phase 1 lists a comprehensive description of the hardware setup and some of the components. This reflects some of the pitfalls we encountered during this phase.





# No 2.

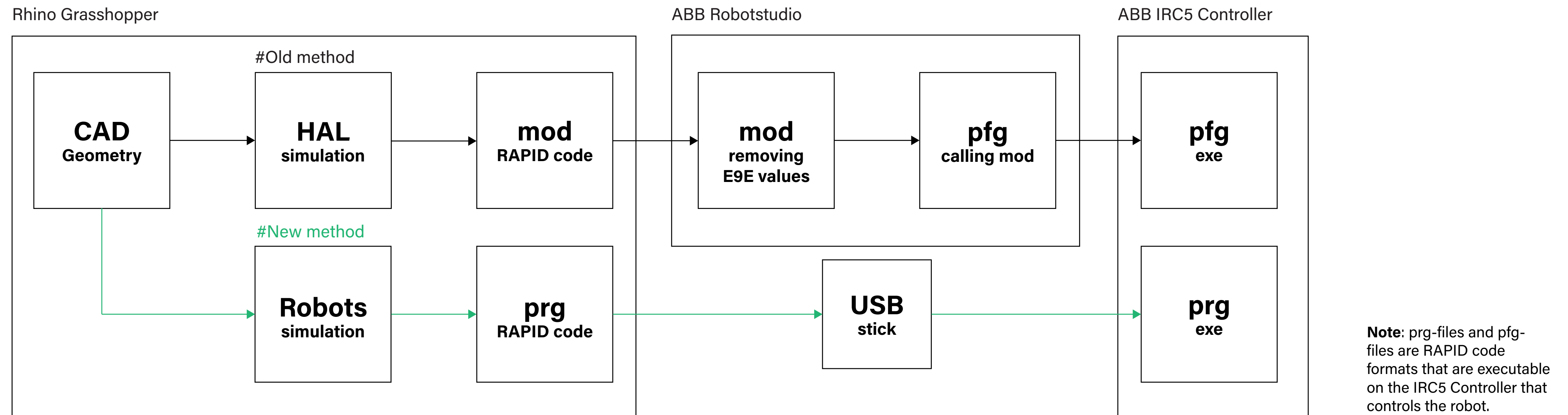
# Software setup

During this phase we defined the principles for the surface topologies and conducted some tests.

## Robots plugin

Robots is a plugin for Rhino's Grasshopper visual programming interface. It allows you to create and simulate robot programs for ABB, KUKA, UR, and Staubli robots. It is compatible with Rhino 7 and higher for Windows and MacOS.

Other simulation plugins exist, such as HAL that has been previously used to generate RAPID code for the robot that we are using. However, we have found the Robots plugin to offer a more effective and straight-forward while avoiding many of the issues that were reported by previous users of the robot.



# Programs

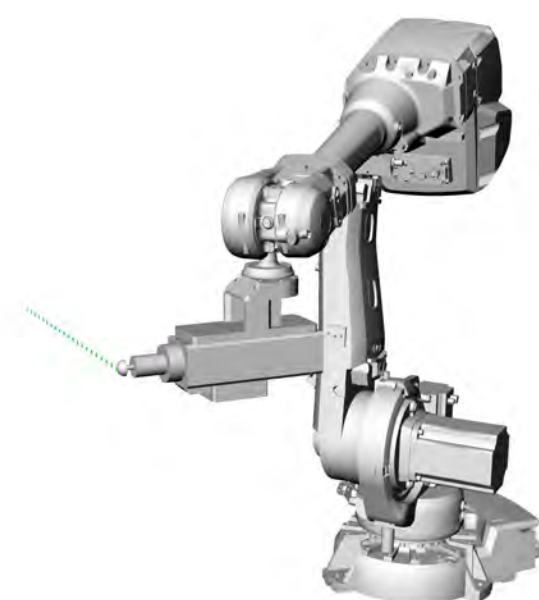
After attaching the tool with the TCP coordinates as well as the tool mesh and also attaching the correct robot model, the only thing missing is to connect a program.

The program is a list of planes, rather than a list of points, as points can be approached in an infinite amount of ways. The plane includes orientation, and the plugin will always use the z-axis as the entry direction.

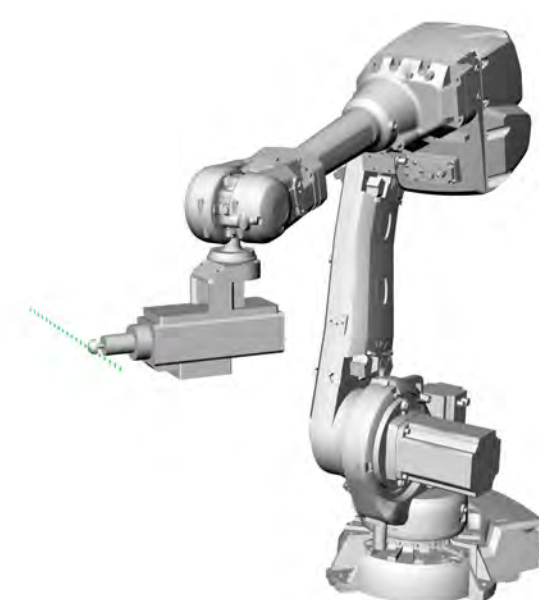
The illustrations show three simulations of identical programs with 21 planes with the only difference between them being the orientations of the planes.

### Program 3: Listed coordinates with orientation

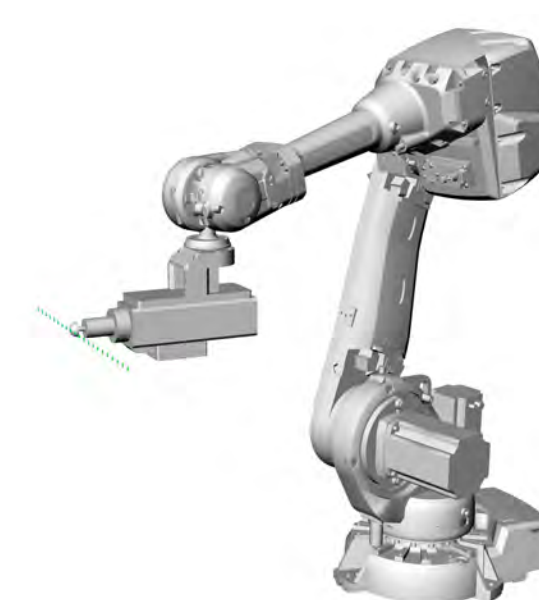
```
{0}
0 o(1230.00,711.00,1012.00) z(0.00,0.00,1.00)
1 o(1230.00,675.45,1010.35) z(0.04,-0.17,0.98)
2 o(1230.00,639.90,1008.70) z(0.10,-0.33,0.94)
3 o(1230.00,604.35,1007.05) z(0.17,-0.47,0.86)
4 o(1230.00,568.80,1005.40) z(0.26,-0.59,0.76)
5 o(1230.00,533.25,1003.75) z(0.35,-0.68,0.64)
6 o(1230.00,497.70,1002.10) z(0.45,-0.74,0.50)
7 o(1230.00,462.15,1000.45) z(0.55,-0.76,0.36)
8 o(1230.00,426.60,998.80) z(0.64,-0.74,0.21)
9 o(1230.00,391.05,997.15) z(0.72,-0.69,0.06)
10 o(1230.00,355.50,995.50) z(0.80,-0.60,-0.08)
11 o(1230.00,319.95,993.85) z(0.85,-0.49,-0.20)
12 o(1230.00,284.40,992.20) z(0.89,-0.34,-0.30)
13 o(1230.00,248.85,990.55) z(0.91,-0.19,-0.37)
14 o(1230.00,213.30,988.90) z(0.91,-0.01,-0.42)
15 o(1230.00,177.75,987.25) z(0.89,0.16,-0.44)
16 o(1230.00,142.20,985.60) z(0.84,0.33,-0.42)
17 o(1230.00,106.65,983.95) z(0.79,0.49,-0.37)
18 o(1230.00,71.10,982.30) z(0.71,0.64,-0.30)
19 o(1230.00,35.55,980.65) z(0.63,0.75,-0.20)
20 o(1230.00,0.00,979.00) z(0.53,0.84,-0.08)
```



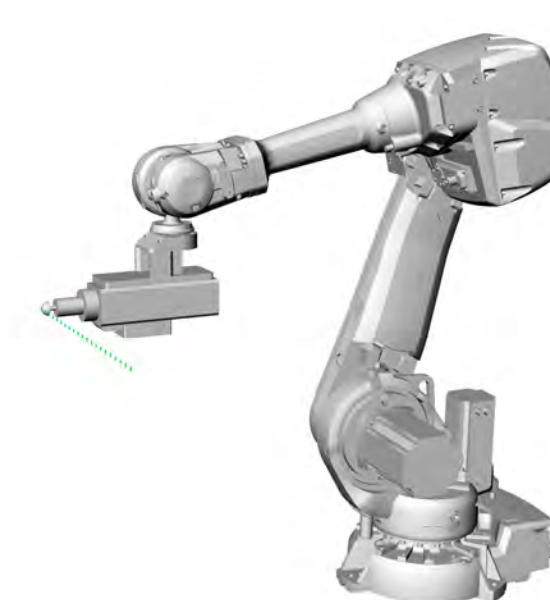
Program 1: Beginning



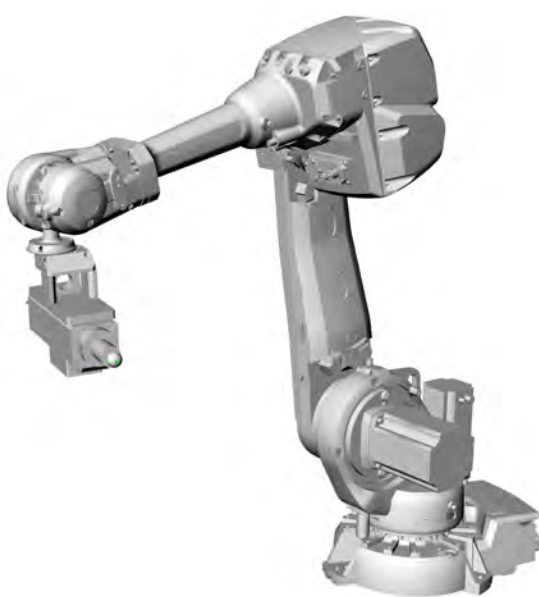
Program 1: at plane 7



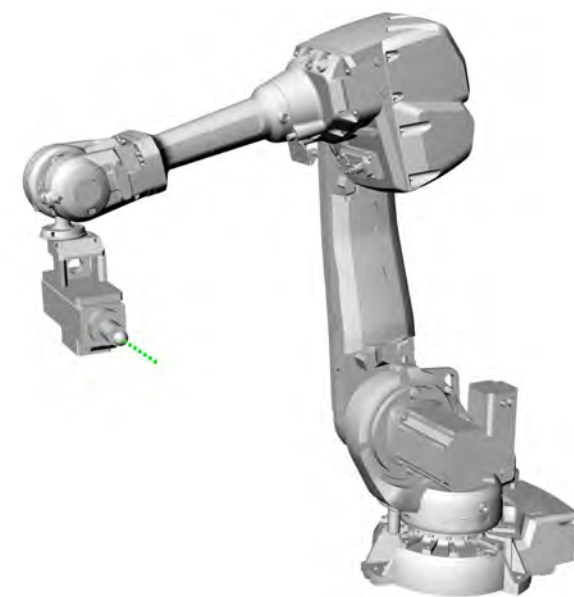
Program 1: at plane 14



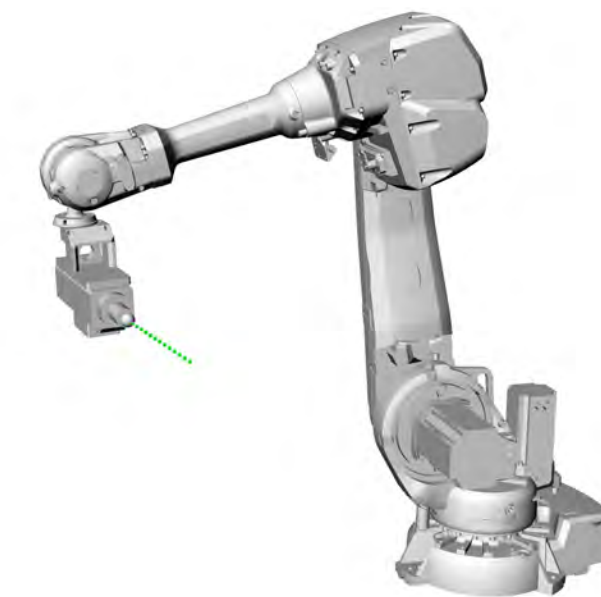
Program 1: Completed



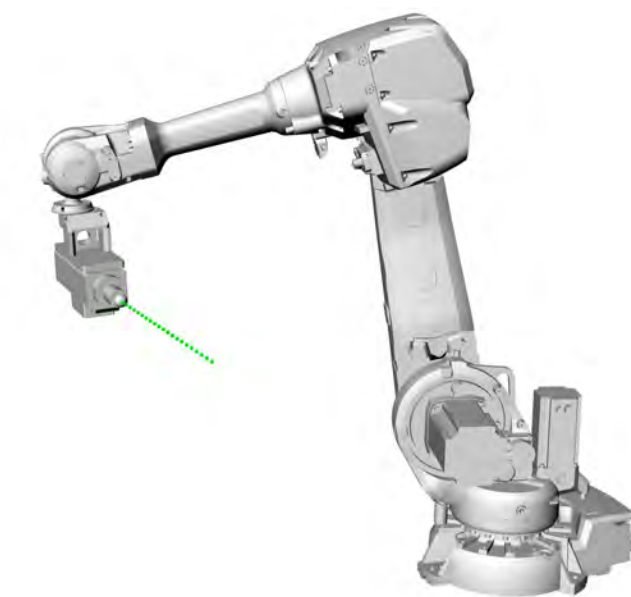
Program 2: Beginning - 0 coordinates



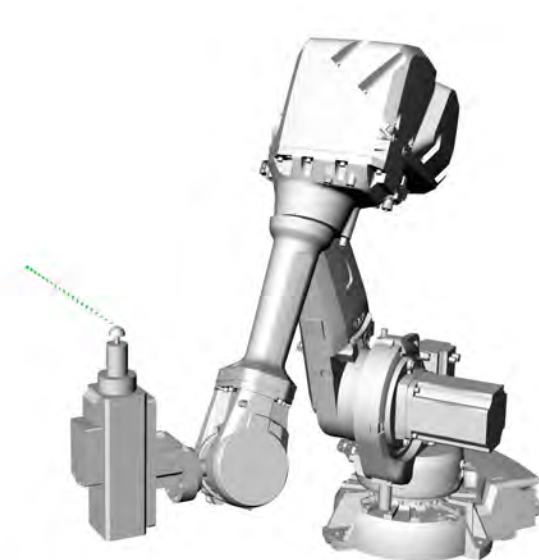
Program 2: at plane 7



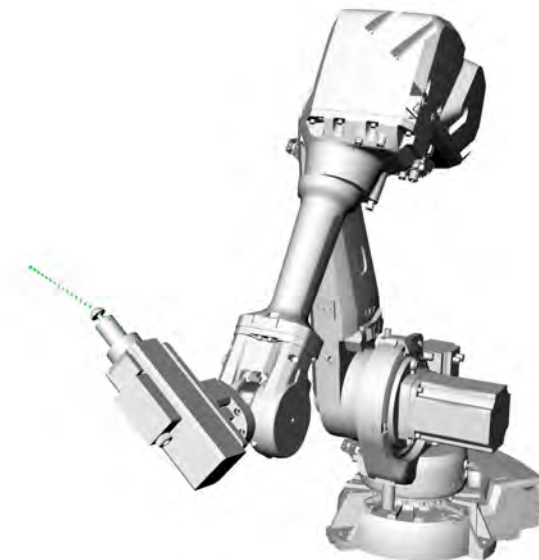
Program 2: at plane 14



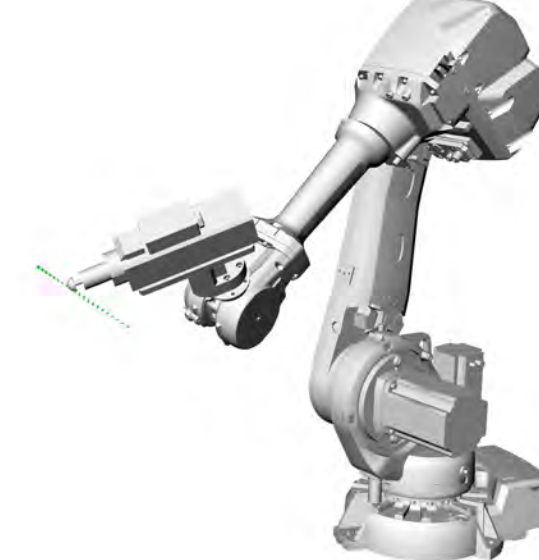
Program 2: Completed



Program 3: Beginning - 0 coordinates



Program 3: at plane 7



Program 3: at plane 14



Program 3: Completed

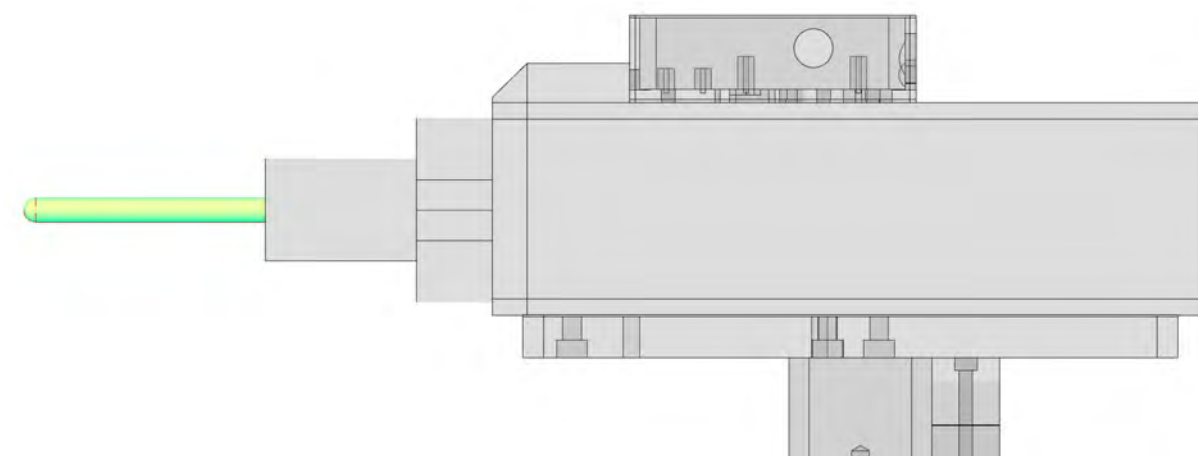
## Geometric approach

The computational strategy was to create geometries that would both display and provide milling curves for the robot.

When creating surface geometries in this project, we decided that we will only use ball mills, or nose mills when doing surface milling. Thus, the only thing that can affect the surface patterns will be constrained by the radius of the milling tool. Therefore the surface is always a result of circles, or hemispheres with radiuses equal to that of the particular milling tool that is used.

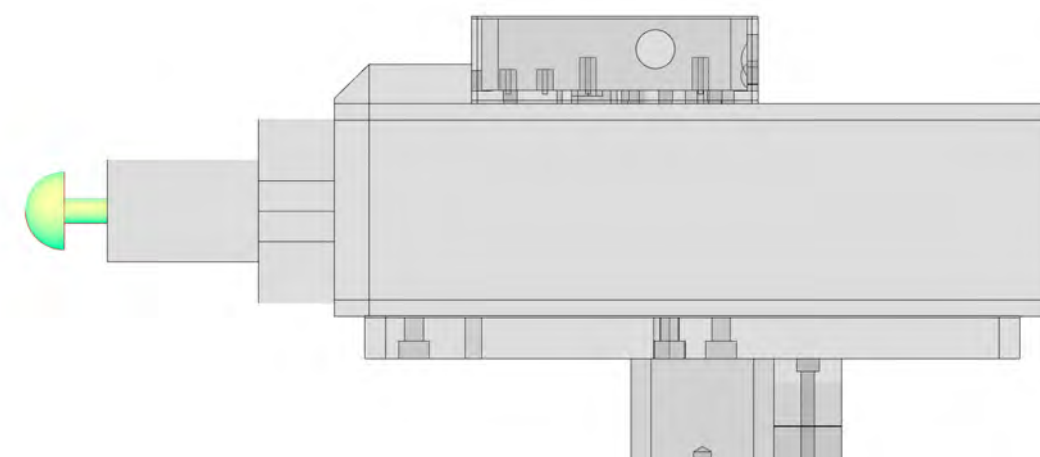
### Tool 1

**Type:** ballmill  
**Tool diameter:** 12 mm  
**Fastening diameter:** 12 mm  
**Tool length:** 118 mm  
**Brand:** unknown



### Tool 2

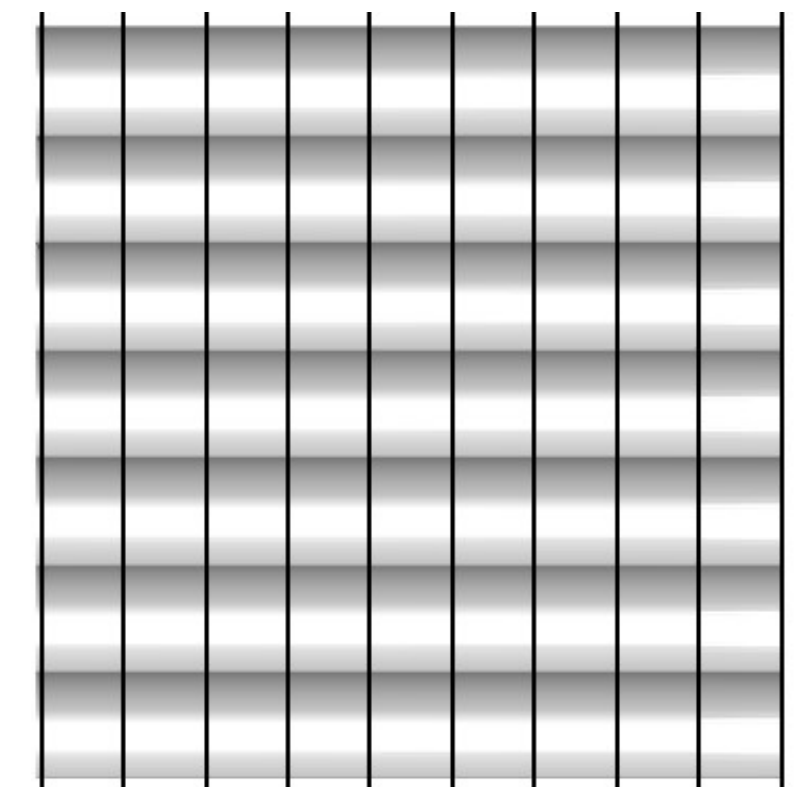
**Type:** ballmill  
**Tool diameter:** 38.1 mm  
**Fastening diameter:** 12 mm  
**Tool length:** 40 mm  
**Brand:** CMT



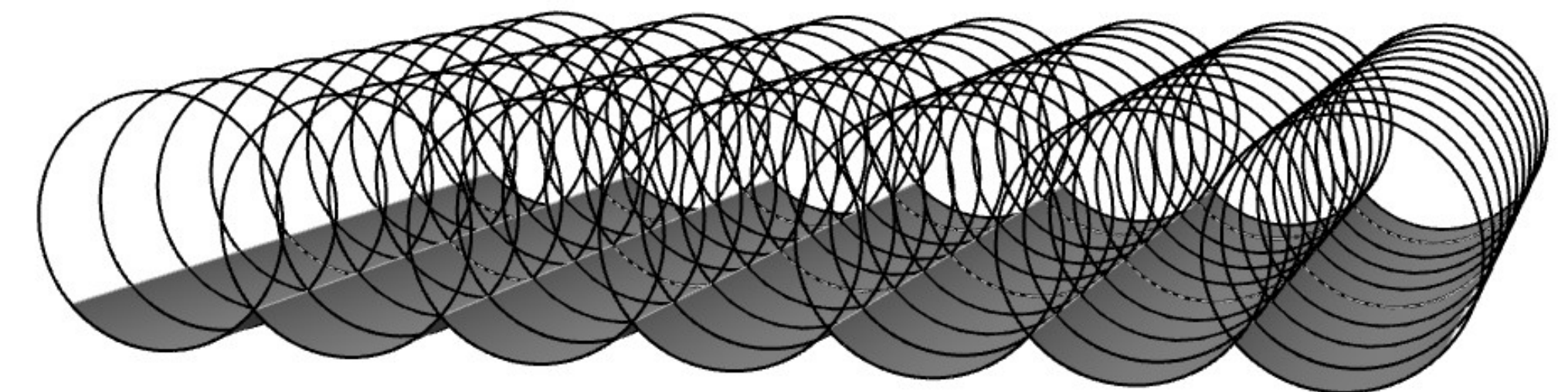
Tool 1 geometry: Front view



Tool 1 geometry: Side view



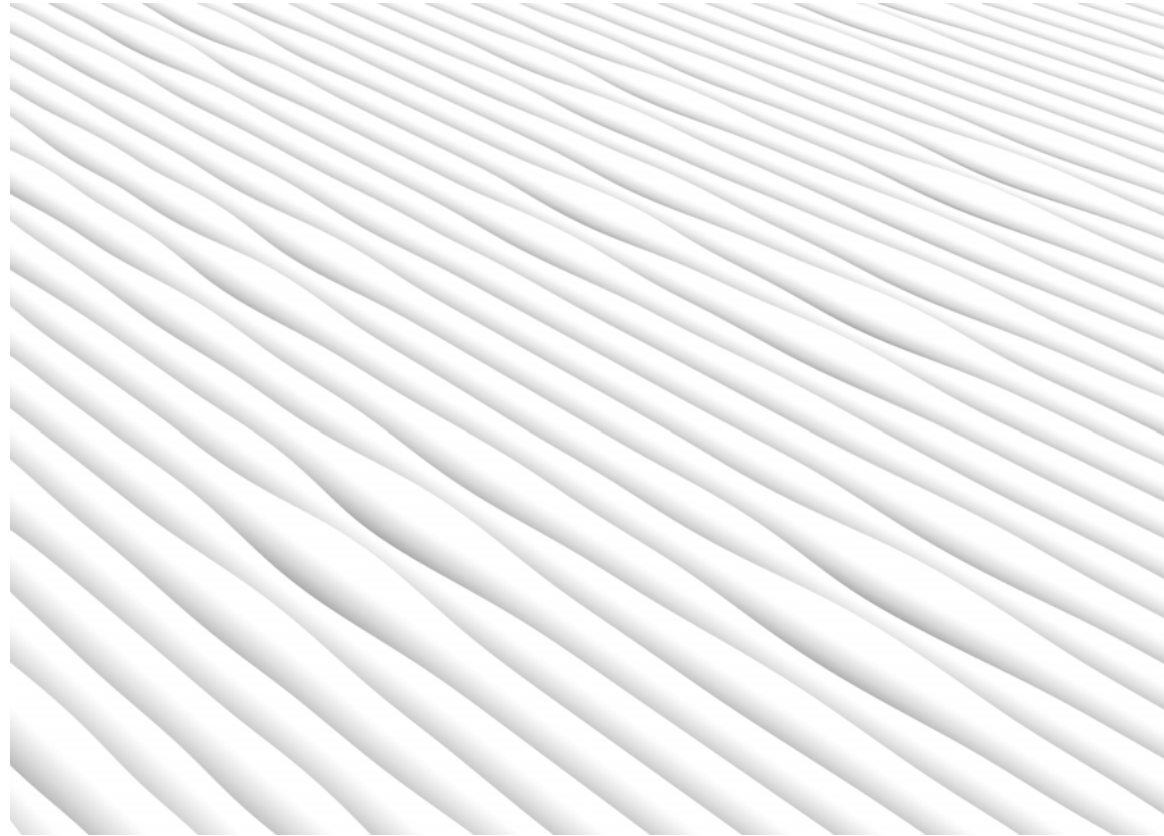
Tool 1 geometry: Top view



Tool 1 geometry: Perspective view

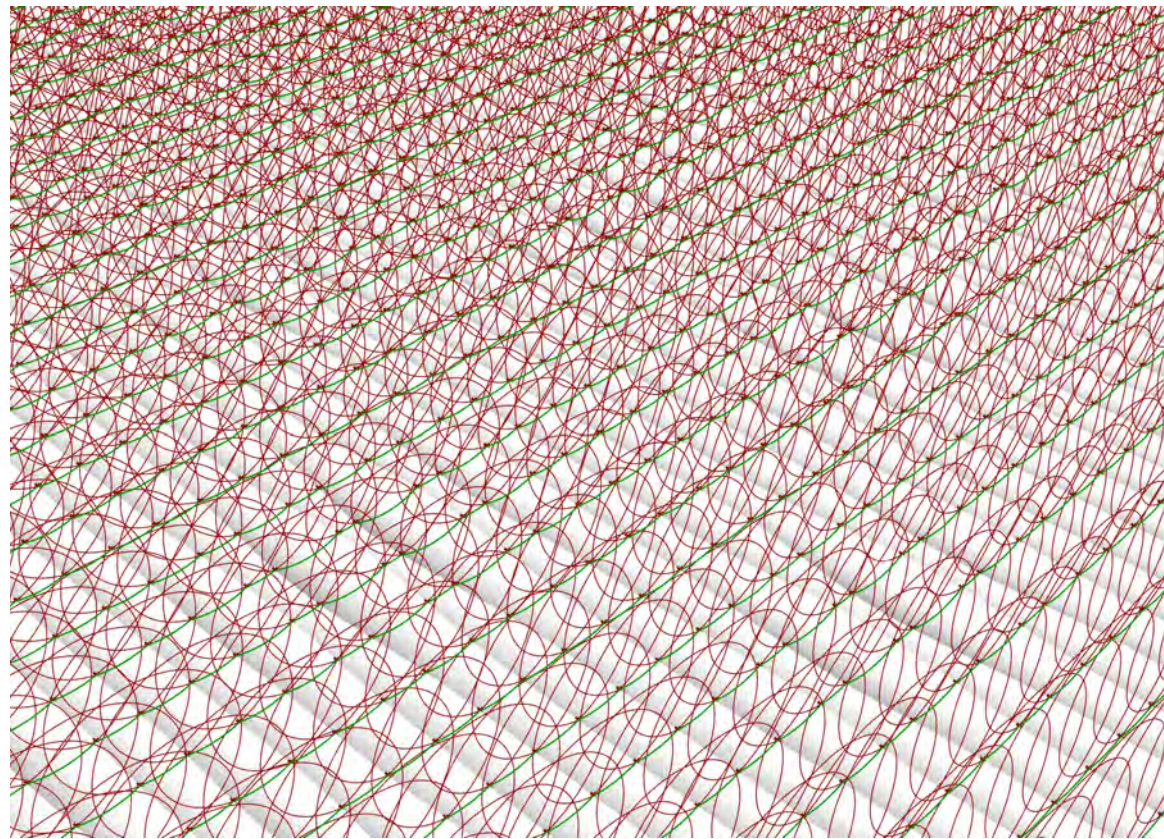
## Alternating waves surface process

In order to test if our calibration efforts had succeeded, we designed a fairly detailed surface with no more than 2 mm of maximum depth. If the output is correct we have an indication that the error margins are acceptable for future vase designs



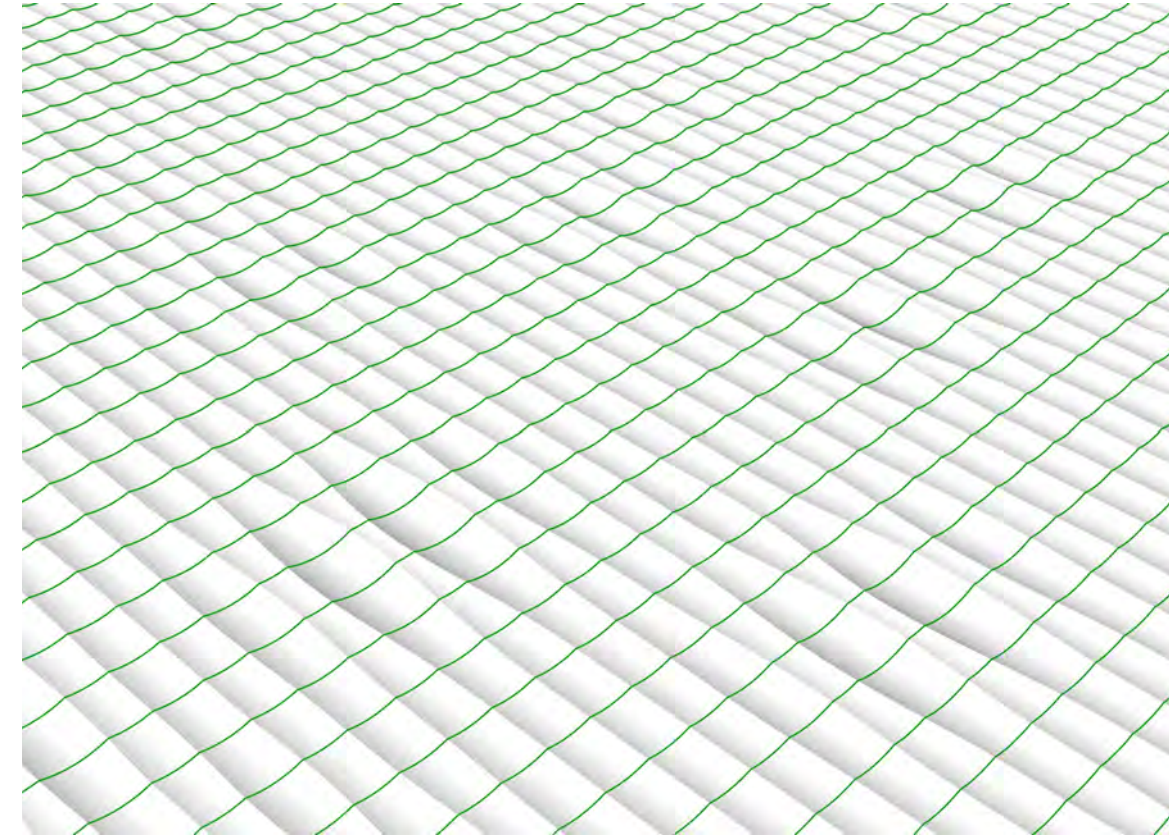
### Lofted curves

The final visual representation of the panel surface



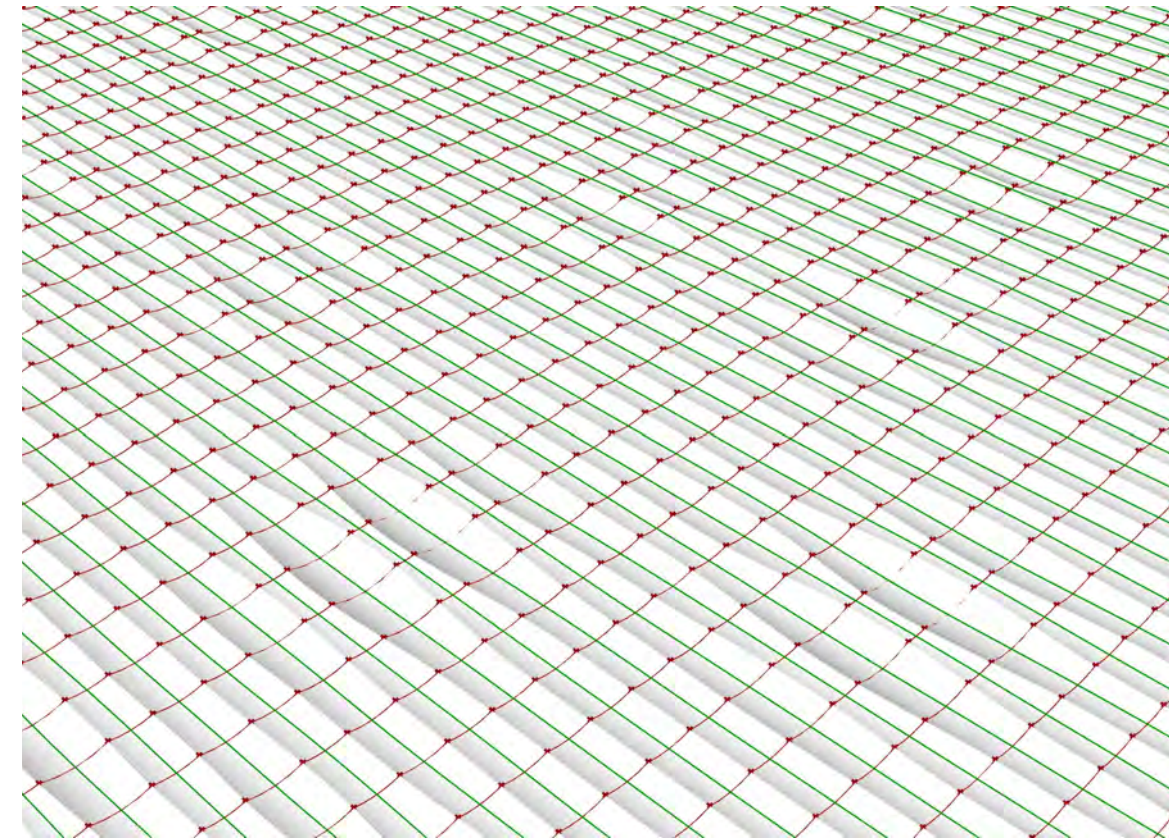
### Circle split

Circles with radius equal to that of the ballmill that will cut the pattern are split at each point and then selected away.



### Joint cuves

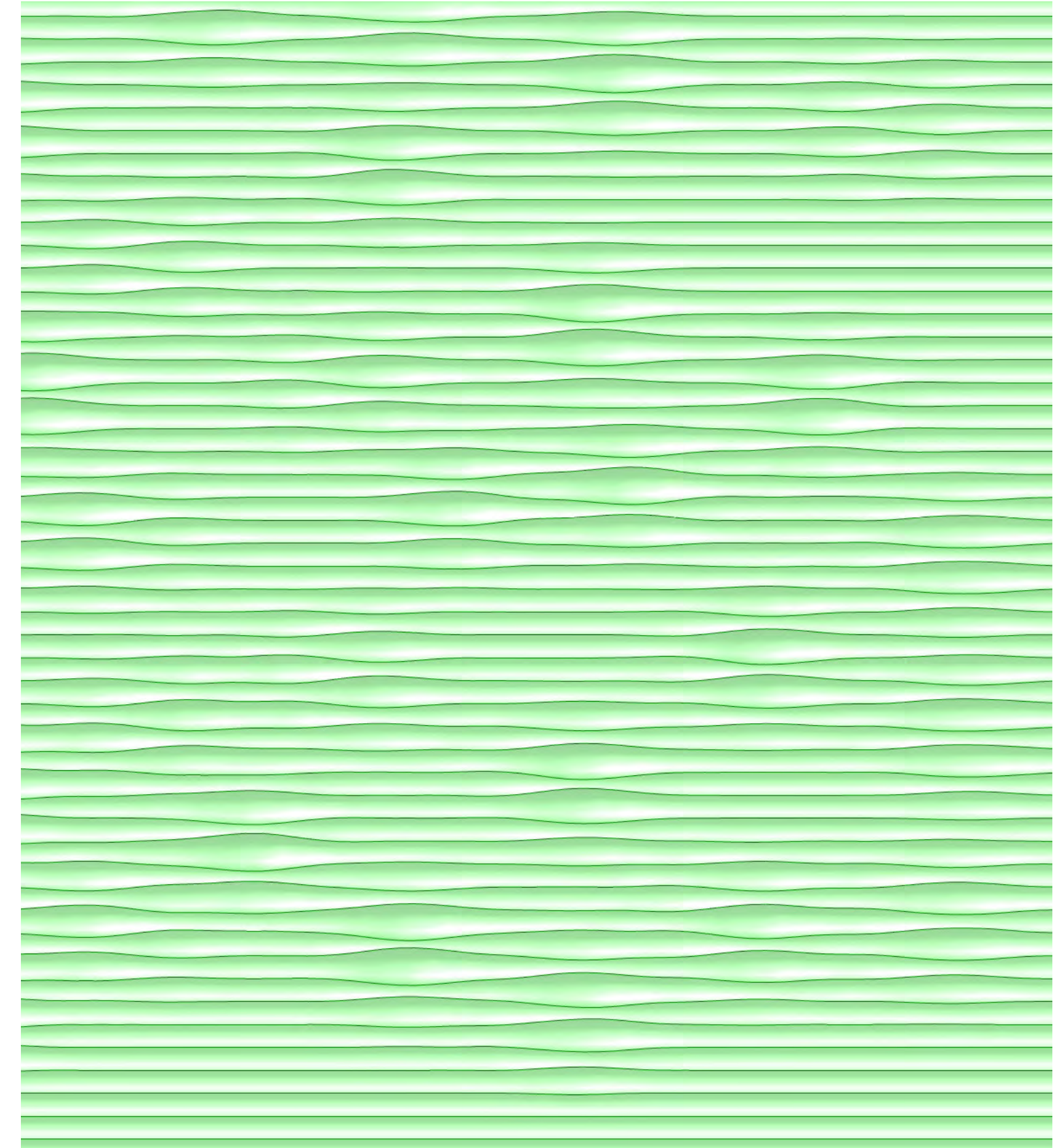
Curves that are joined and arranged into the correct order before the lofting happens



### Cutting paths found at curve middle poitns

By connecting the middle points in a nurbs curve the cutting paths that will make this particular panels far

## Panel surface output

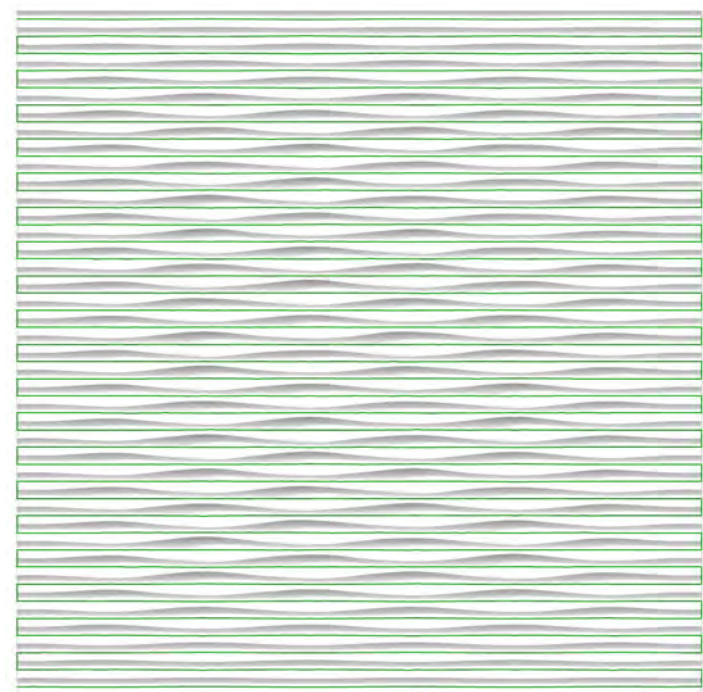


### Visual output

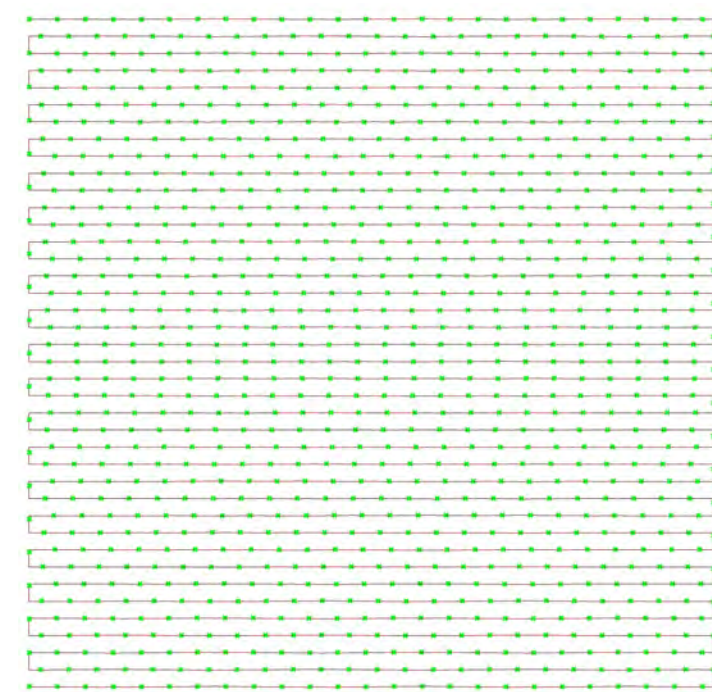
This is the final visual output viewed from top view

## Plane generation

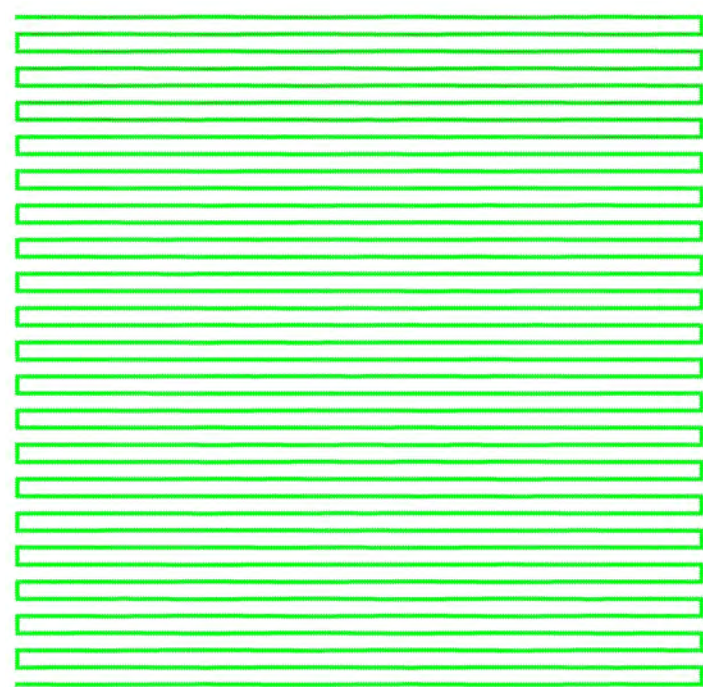
After modeling the desired geometry, as well as creating the curve that the robot should follow, the next step is to convert this curve into TCP-coordinates that the robot can follow. This can be achieved by dividing the curve into points with equal distance inbetween. In order to ensure a high resolution of details, we can increase the number of points, thus reducing the distance in between each point.



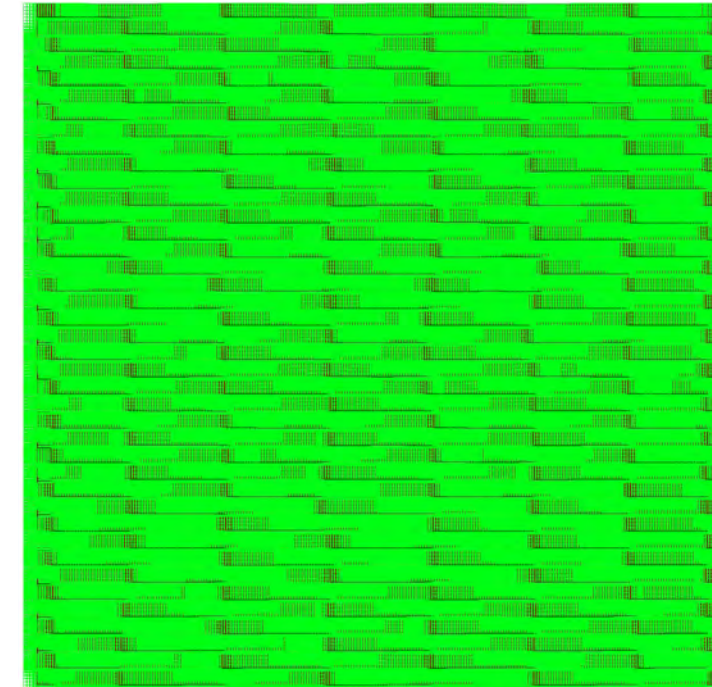
**Step 1:** Continuous curve



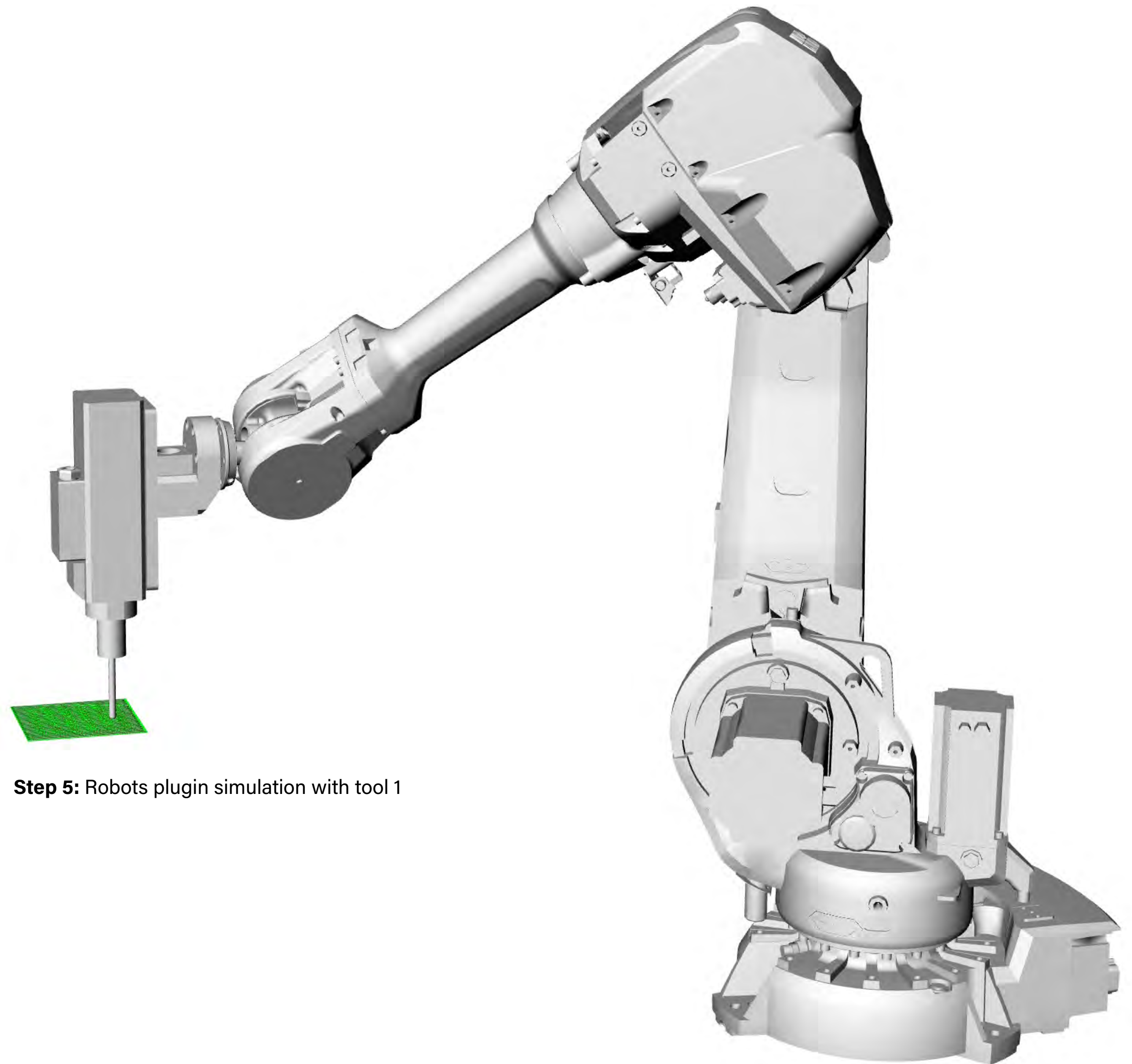
**Step 2:** Curve divided into 1000 points



**Step 3:** Increase to 10 000 points



**Step 4:** Convert points into 10 000 XY-oriented planes

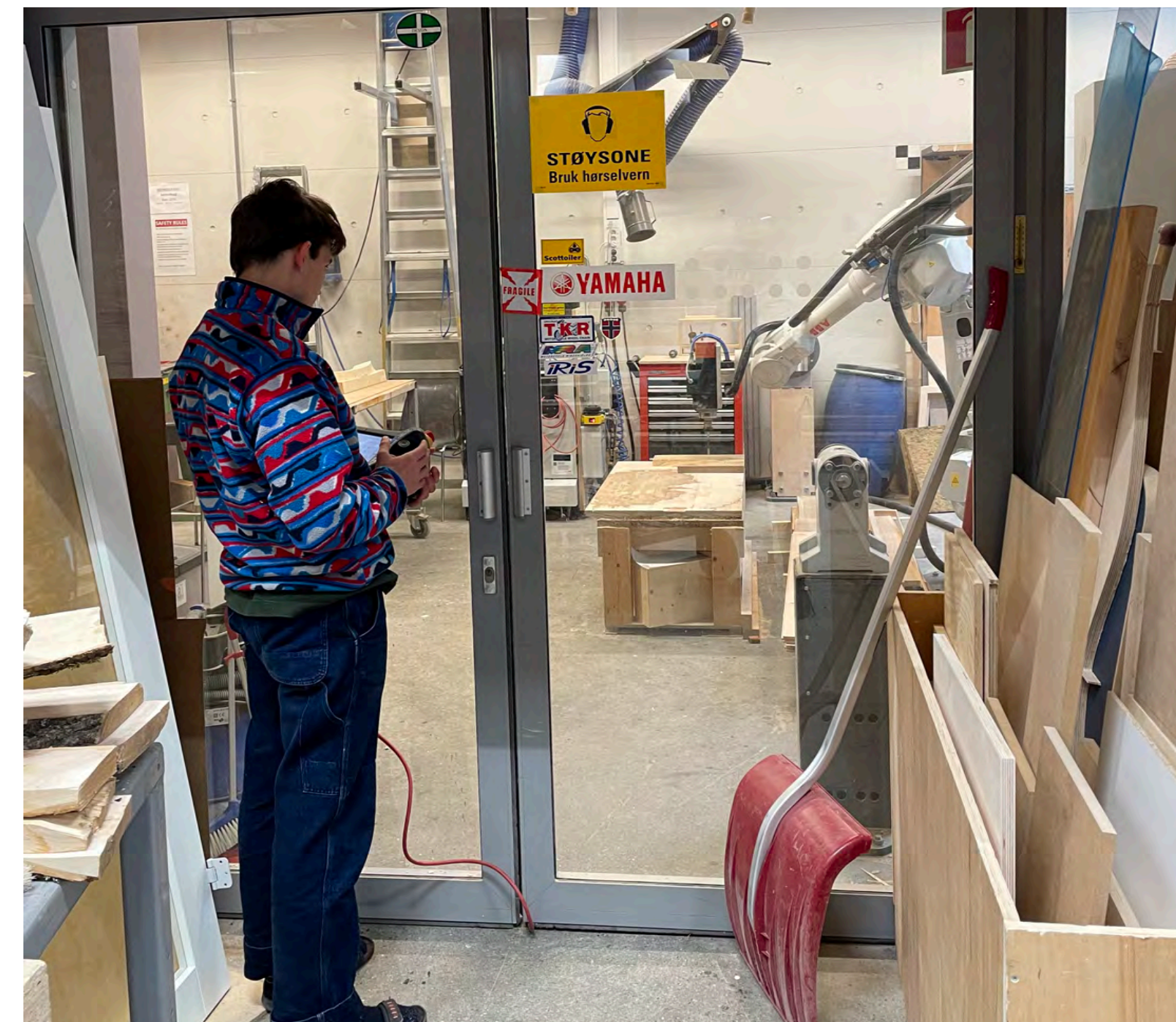


**Step 5:** Robots plugin simulation with tool 1



## Milling

After securing the workpiece to the table and confirming that the starting point was correct we ran the surface milling files. The milling file mainly contained one continuous curve, divided into 14000 TCP planes, that the robot would follow.



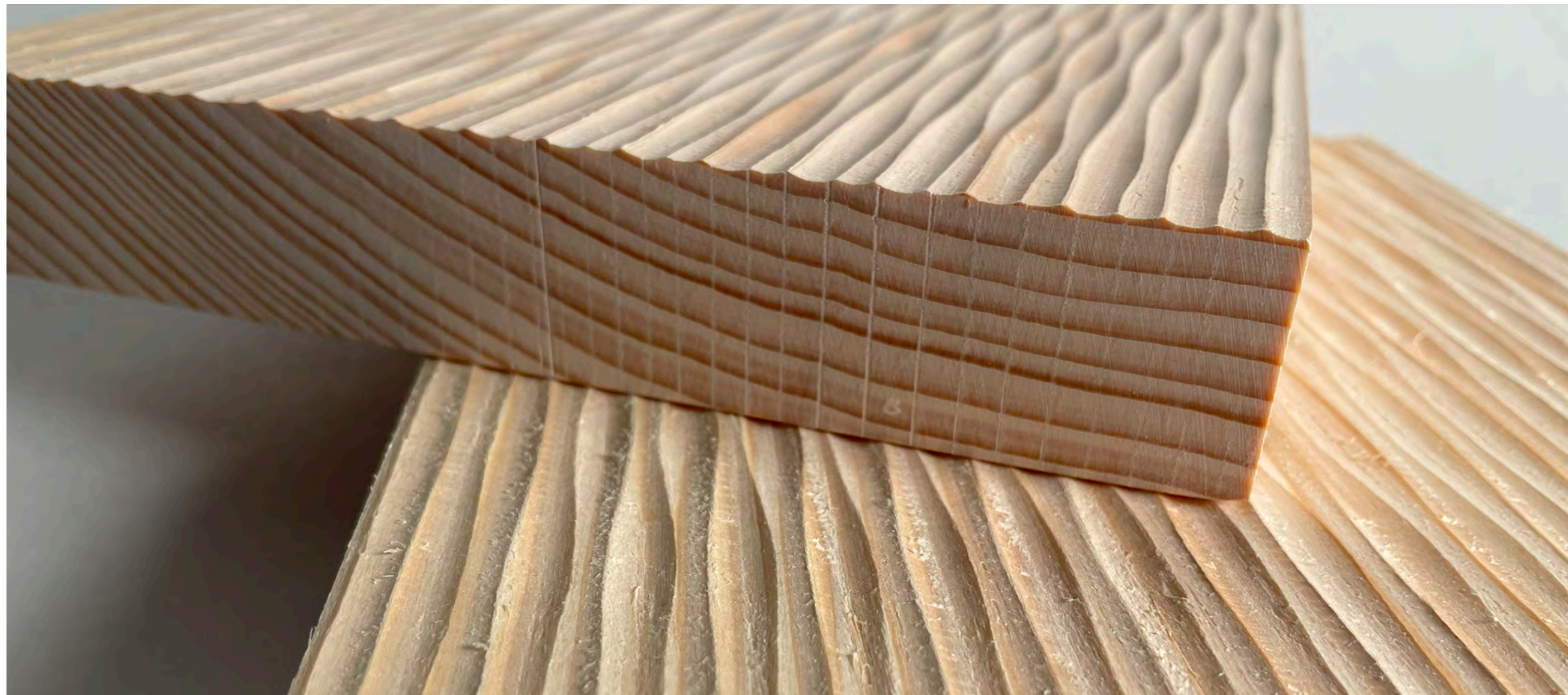
**Figure 4 - Milling**

The RAPID files are run from the Flexpendant from behind the protective glass for safety reasons.



**Milled surface**

Closeup of surface milled at  
10 000 RPM



**Material surfaces**

Milled at 10 000 and 15 000 RPM. Unsurprisingly, there was an  
increase in surface quality with the higher spindle speed.

Testing panels were milled in order to observe the precision of the relatively large robotic arm in contact with a rather detailed surface. The results allowed us to conclude that the precision is sufficient for our project.

**Material:** Sitka spruce

**Dimensions:** 188 x 188 mm

**Thickness:** 30 mm

**Tool diameter:** 12 mm

**Tool type:** Ballnose mill

**Spindle speed:** 10 -15 000 RPM

**Milling direction:** Along fibers

**Milling height difference:** +/- 2 mm







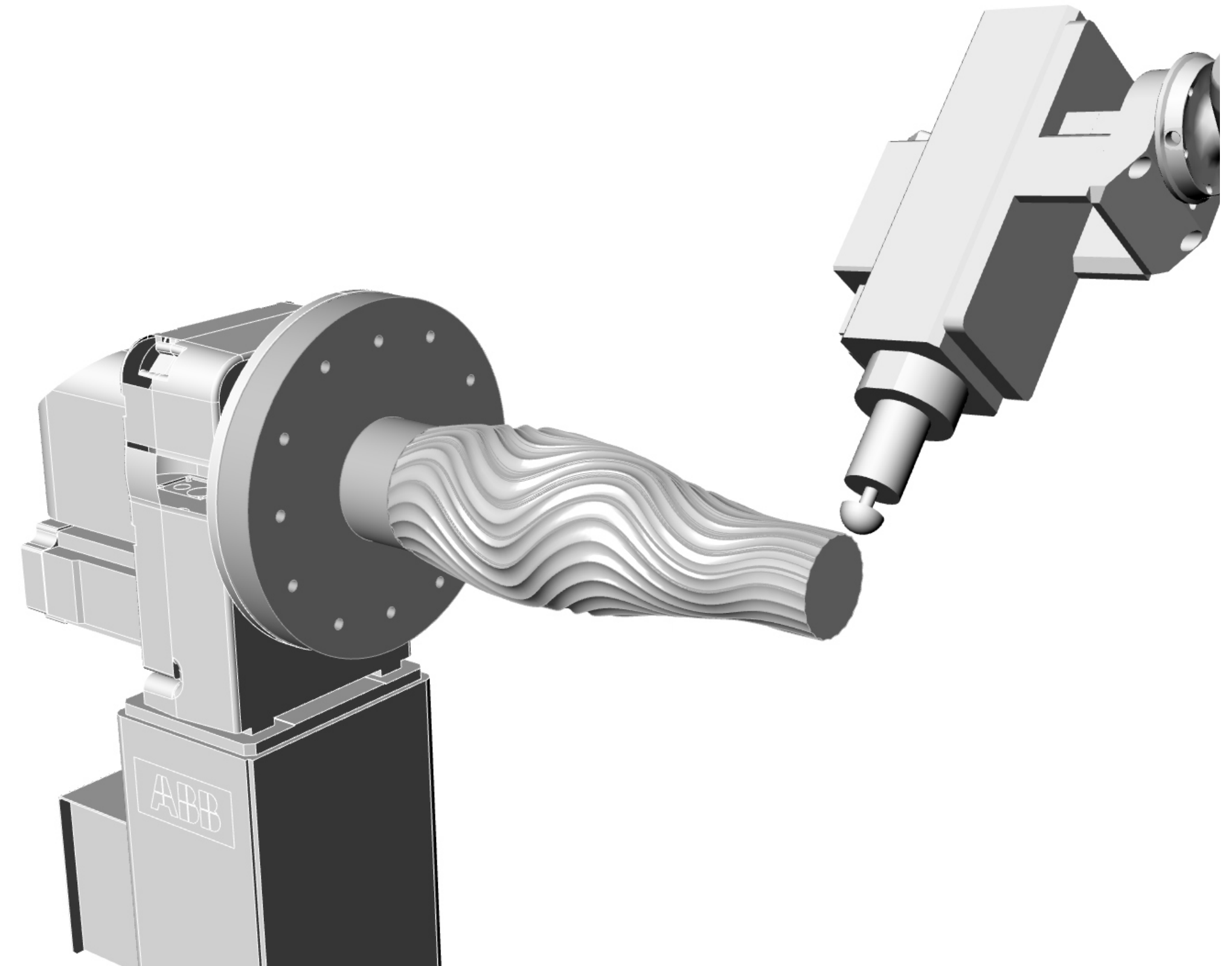
**Alternating waves surface pattern**

The surface pattern becomes increasingly visible when exposed to contrasting lights.



## Insight - Surface samples

Before we could start using the robot for surface milling we had to check the robot arm and the digital model of the arm and end of arm tool (eoa). The inspection of the articulated arm disclosed inaccuracy when operated as it lacked service, where the latter was mentioned briefly in dialogues with the crew in Lukas' workshop. This can become a huge disadvantage for future use, since it can become unusable if ignored. As for digital resources, we were provided with a model of the eoa, but it proved to be oriented wrong. This took time to correct, but once corrected made the workflow better. Further corrections of the digital model for future use is therefore not required, but we still include the necessary coordinates for the ones eager to try it out for the purpose of robotic fabrication. As for the surface geometry which is milled, we include the parametric principles used for making the intricate patterns, which are made in Grasshopper - a Rhino plug-in. The approach of creating such patterns is diverse and we encourage people to use the mentioned software when using the robot in the Lukas' workshop. The reason for this statement is because the workshop is mainly used by architecture students and employees from the respective faculty. Hence, it is possible to get some help from them when working in Rhino and Grasshopper.



# No 3. Vases

During this phase we have designed vases and employed the positioner to fabricate them.

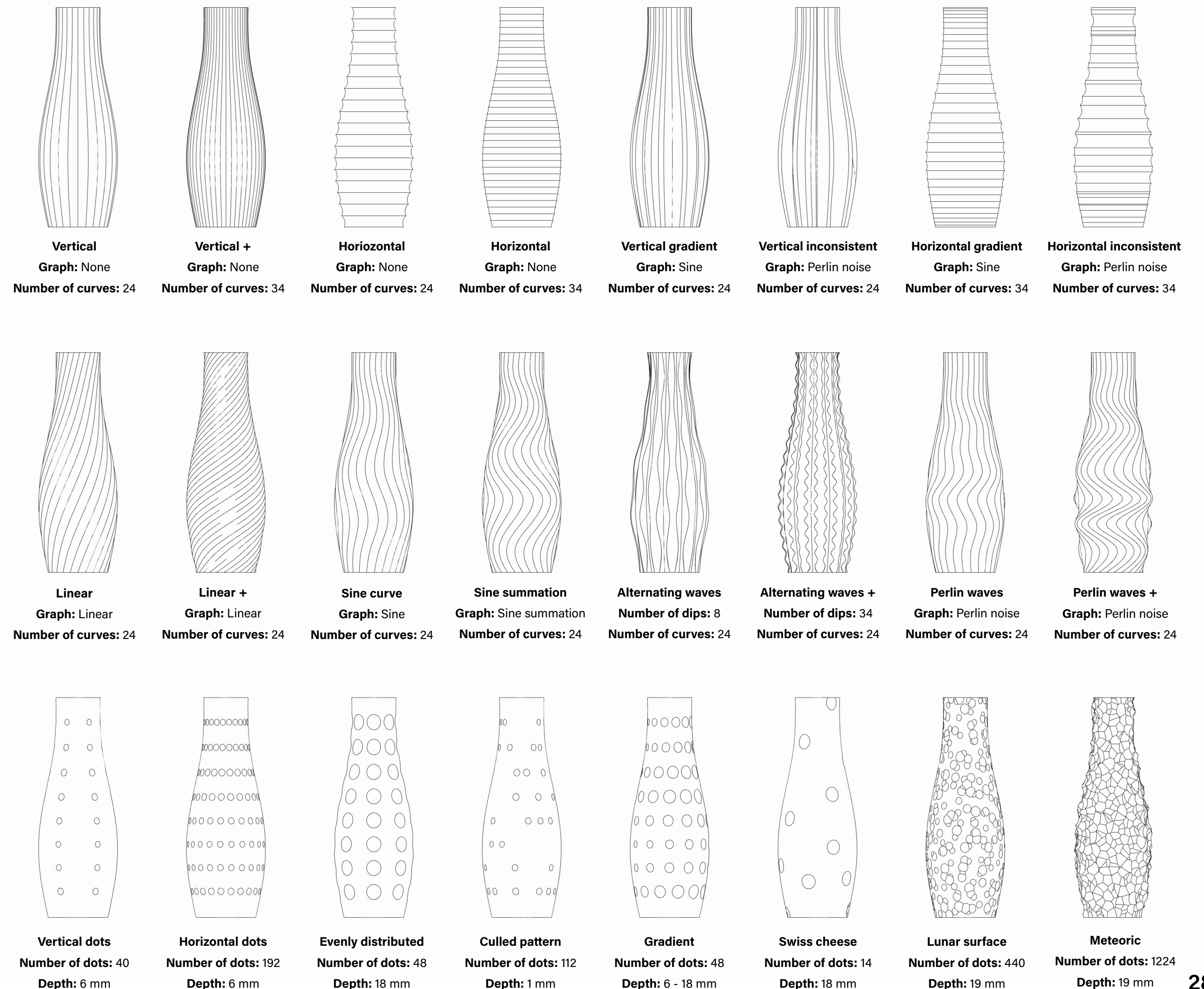
## Vase designs

Three series of vases were made in order to explore some of the different surface geometries that are possible given the restraint of the hemispherical ball mill.

**The lined vases** is a basic exploration of lined surface geometries that can be created by the ball mill. The lined vases illustrate the results of varying distances between each line on a curved surface and emphasizes directionality.

**The wavy vases** are our direct interpretation of the O. D. Kriegs vases. They also demonstrate different types of waves that are possible using this concept. We can also control the intensity of these wavelike surface patterns.

**The dotted vases** emphasize the fact that, given our geometrical constraints, we are not obliged to create continuous patterns. They might also be inconsistent and random. The dotted surface geometries also play with the span between the crater sizes that occur when the ball mill is breaking the surface of the workpiece at different depths.



## Calling the positioner

The positioner has never, to our knowledge, previously been used in the Lukas Workshop, so we could not rely on any local experience provided by our supportive network. With help from Ådne Solhaug Linnerud, we were able to identify the external axis command in the RAPID code that we already had developed.

```
MODULE milling_T_ROB1
VAR extjoint extj := [0,0,0,0,0,0];
VAR confdata conf := [0,0,0,0];
PERS tooldata spindle:=[TRUE,[[471.5,0,186.118],[0.5,0.5,0.5,0.5]],[0.001,[471.5,0,186.118],[1,0,0,0],0,0,0]];
TASK PERS wobjdata DefaultFrame:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
TASK PERS speeddata Speed000:=[100,180,5000,1080];
TASK PERS zonedata Zone000:=[FALSE,20,20,20,2,20,2];
PROC Main()
ConfL \Off;
MoveAbsJ [[-12.4576,7.3372,19.0036,-95.5985,78.8534,116.8875],extj],Speed000,Zone000,spindle;
MoveL [[1199,-586,627],[0,1,0,0],conf,extj],Speed000,Zone000,spindle \WObj:=DefaultFrame;
MoveL [[1199,-586,607],[0,1,0,0],conf,extj],Speed000,Zone000,spindle \WObj:=DefaultFrame;
MoveL [[1199,-586,587],[0,1,0,0],conf,extj],Speed000,Zone000,spindle \WObj:=DefaultFrame;
MoveL [[1199,-586,567],[0,1,0,0],conf,extj],Speed000,Zone000,spindle \WObj:=DefaultFrame;
MoveL [[1199,-586,547],[0,1,0,0],conf,extj],Speed000,Zone000,spindle \WObj:=DefaultFrame;
ENDPROC
ENDMODULE
```

It turns out, in the particular setup at the Lukas workshop, it is the 5th index of the value extjoint that defines the rotation of the positioner.

**VAR extjoint extj := [0, 0, 0, 0, 0, 0]**

Thus by changing this value between 0-360, we can decide the rotation of the positioner at each TCP-coordinate. This is because the variable extj is called in each line. While milling a cylindrical vase, we divide the milling sections into 24 equal parts and call a new variable when we need to rotate the cylinder 15 degrees.

```
MODULE milling_T_ROB1
VAR extjoint extj := [0,0,0,0,0,0];
VAR extjoint extj1 := [0,0,0,0,15,0];
VAR extjoint extj2 := [0,0,0,0,30,0];
VAR extjoint extj3 := [0,0,0,0,45,0];
VAR extjoint extj4 := [0,0,0,0,60,0];
VAR extjoint extj5 := [0,0,0,0,75,0];
VAR extjoint extj6 := [0,0,0,0,90,0];
VAR extjoint extj7 := [0,0,0,0,105,0];
VAR extjoint extj8 := [0,0,0,0,120,0];
VAR extjoint extj9 := [0,0,0,0,135,0];
VAR extjoint extj10 := [0,0,0,0,150,0];
VAR extjoint extj11 := [0,0,0,0,165,0];
VAR extjoint extj12 := [0,0,0,0,180,0];
VAR extjoint extj13 := [0,0,0,0,195,0];
VAR extjoint extj14 := [0,0,0,0,210,0];
VAR extjoint extj15 := [0,0,0,0,225,0];
VAR extjoint extj16 := [0,0,0,0,240,0];
VAR extjoint extj17 := [0,0,0,0,255,0];
VAR extjoint extj18 := [0,0,0,0,270,0];
VAR extjoint extj19 := [0,0,0,0,285,0];
VAR extjoint extj20 := [0,0,0,0,300,0];
VAR extjoint extj21 := [0,0,0,0,315,0];
VAR extjoint extj22 := [0,0,0,0,330,0];
VAR extjoint extj23 := [0,0,0,0,345,0];
VAR confdata conf := [0,0,0,0];
```

### RAPID code positioner variables

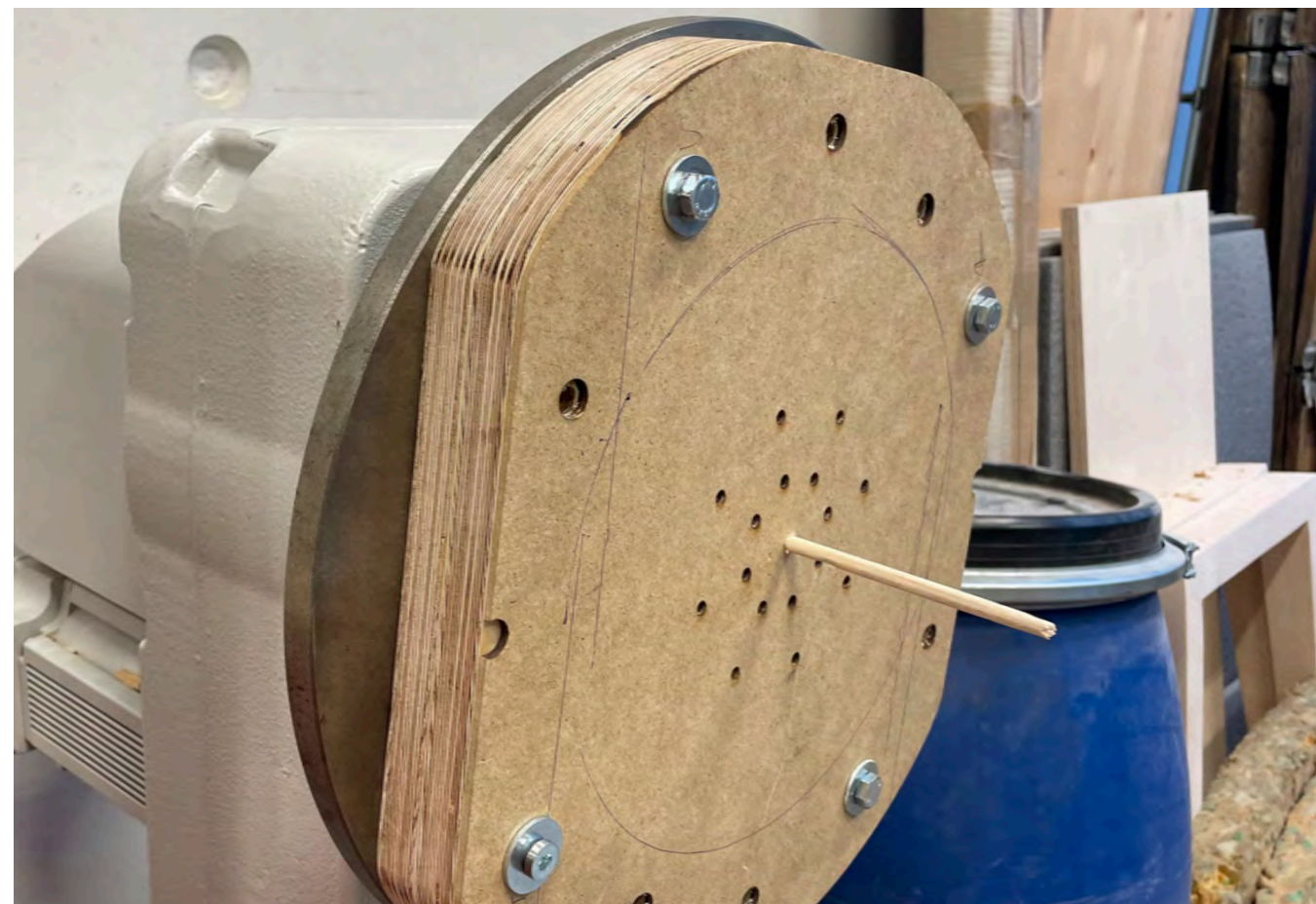
These variables are called each time the positioner rotates

## Positioner setup

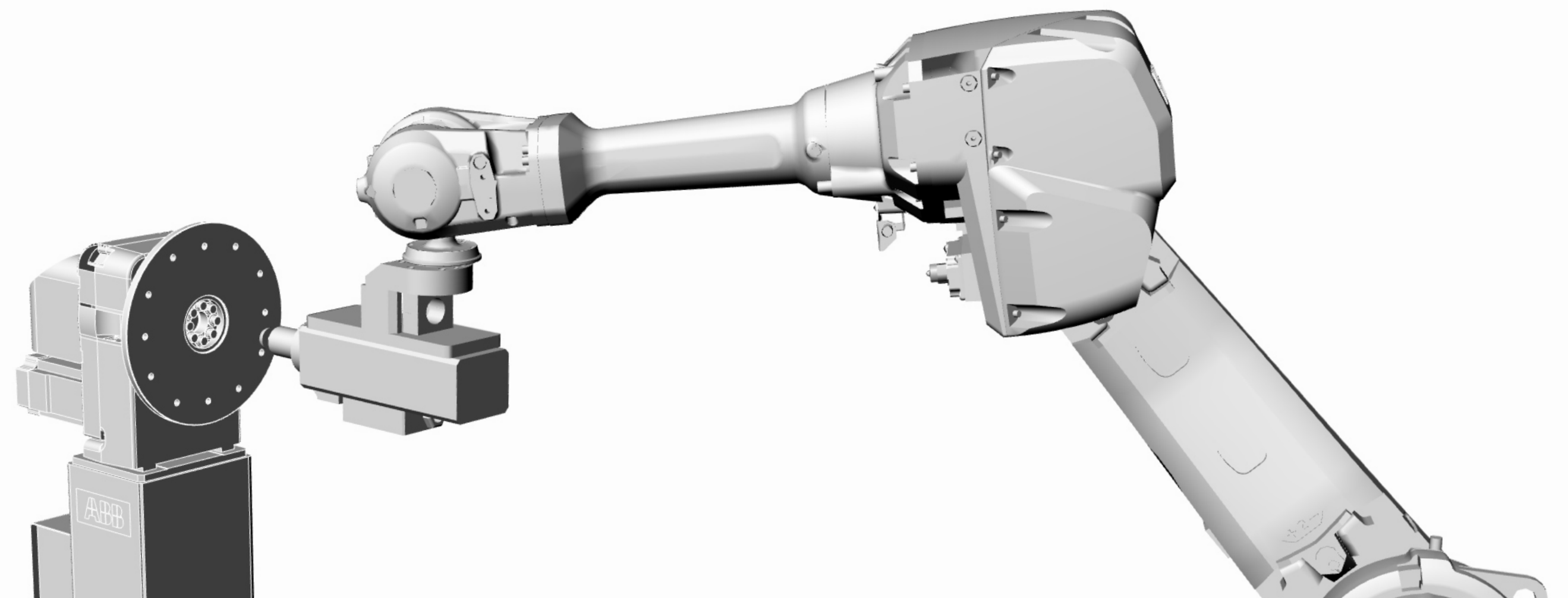
The coordinate needed to successfully operate the positioner is the center. We found this by jogging the robot and noting the cartesian coordinates. Then we ran a script that would cause the robot to approach this point + 100 mm in y direction (in order to avoid collision).

## Positioner fastening plate

In order to use the positioner we needed to make a fastening plate that would allow us to fasten our vases to the axis. To ensure correct measurements, we laser cut a 4 mm MDF plate based on the CAD file that was provided by ABB. Then we glued 2 layers of 15 mm pine plywood and shaped them after the MDF guide. Then we could fasten it in the correct place on the positioner using 4 bolts.



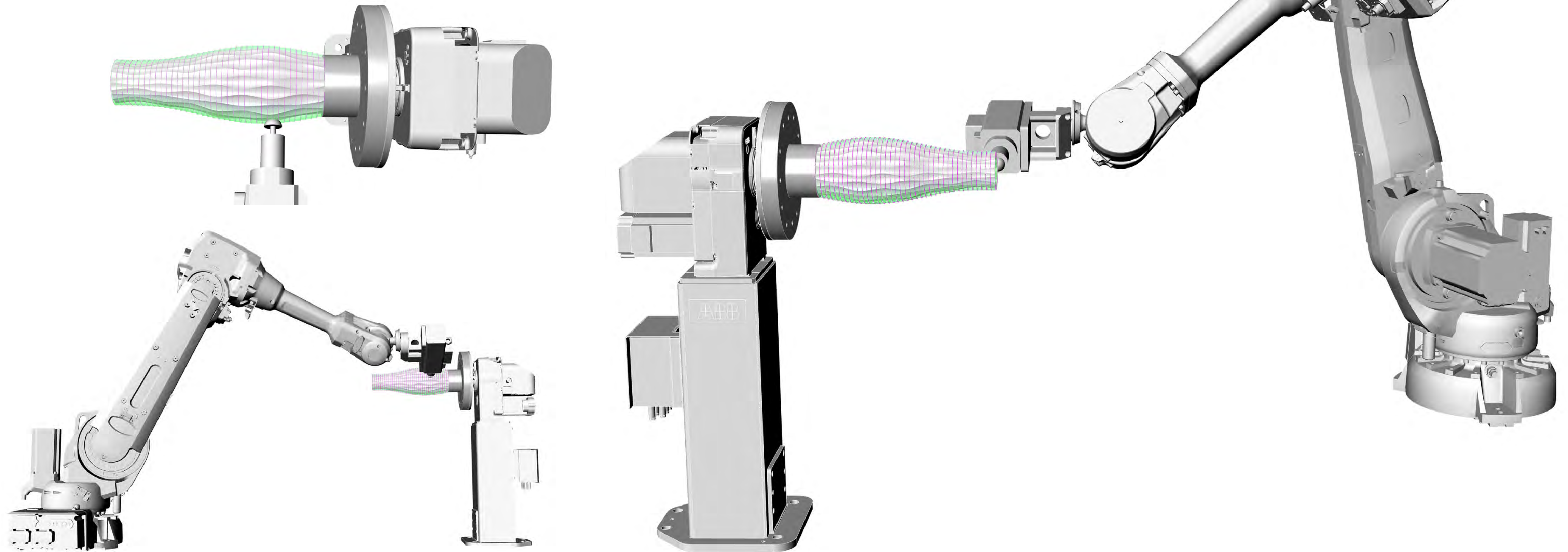
**Finding the center:** This was not correct the first time. We needed to correct the coordinates after testing the script a couple of times.



**Simulation:** In the simulation, the axis always hit the centre. The challenge was to correctly move the positioner geometry to the right place

## Fabrication strategy

The fabrication strategy was to mill each vase via offset curves of 3 mm each, one back and on forth. And then rotate the vase 15 degrees and mill the identical curve 24 times. In the case of the alternating pattern we needed to mill 2 different curves at the same time and then rotate 30 degrees and repeat this procedure 12 times.



**Simulation:** Confirms that the script preformas as expected

# Plywood vase process



## CNC milling

A total of 34x3 circular plates were cut from a 12x1220x2440 mm poplar plywood plate



## Numbering

Numbers were engraved on top to ensure that the plates were assembled in the correct order



## Three vases

Three vases with a height of 466 mm came from plywood plates with minimal waste material



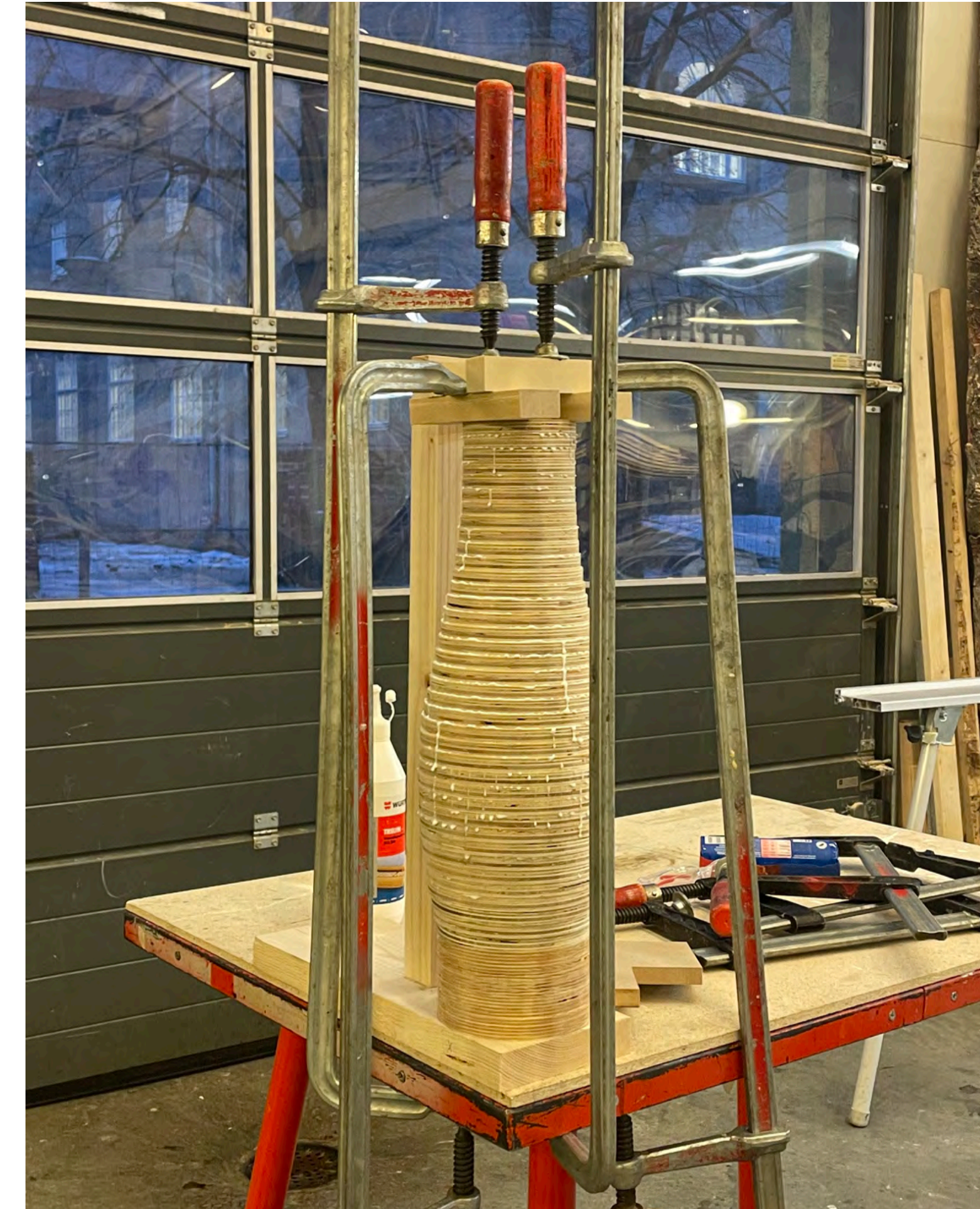
## Inner radius

The fixed inner radius made it possible to use a squared stick as a guide, locking each layer in place



## Gluing

Gluing was done in three stages per 8 layers



## Complete gluing

The final unmilled vase consisted of 34 disks of 12 mm plywood together with an offset base that of 4 layers of 15 mm plywood



# Glulam vase process



**Moelven visit**  
Moelven Limtre AS donated 8 units of birch glulam offcuts to our design project



**Glulams**  
The glulams that Moelven delivers are from sustainable and certified Norwegian and Swedish woods.



**Turning**  
Both vases were turned on the lathe. The one with vertical fiber direction was easier to work with.



**Sideways glulams**  
One of the glulam vases was glued with the wooden fiber direction horizontally in relation to the ground



**Vertical glulams**  
The other glulam vase was glued with the wooden fibers in the vertical direction

Milling





**Milling the vases was an interesting process in itself. A video can be found [here](#).**



**In total four vases were milled. Two made from glulam offcuts and two made from plywood.**





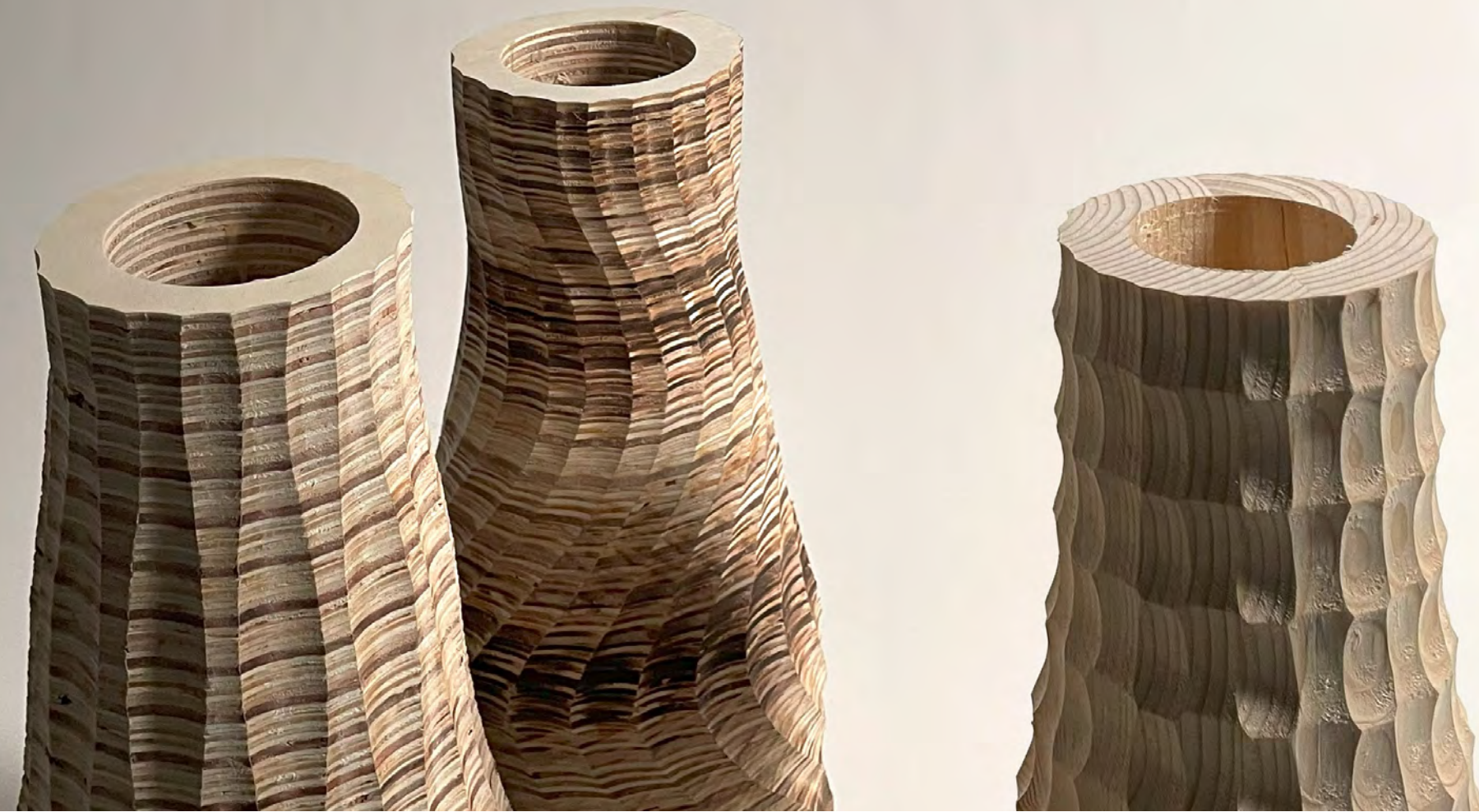
## Results

Being able to mill four vases allowed us to solve the final problem of surface finish. After noticing that the first two vases had black discoloration in some areas where the robotic arm slowed down due to complex movement, we aimed to avoid this when milling the final two vases.

We achieved this by selecting designs that would allow simplification of the robots movement while milling. This effort was successful as we did not see similar burned marks on the final two vases.

Finally, the alternating waves surface pattern became our own original take on robotic surface milling.





### Plywood vase #1

**Material:** Poplar plywood

**Material dimentions:**  
12x1220x2440 mm

**Number of glued units:** 37

**Height:** 466 mm

**Inner hole diameter:** 60 mm

**Design:** Alternating waves (8)

### Plywood vase #2

**Material:** Poplar plywood

**Material dimentions:**  
12x1220x2440 mm

**Number of glued units:** 37

**Height:** 466 mm

**Inner hole diameter:** 60 mm

**Design:** Sine summation

### Glulam vase #1

**Material:** Birch glulam

**Material dimentions:**  
600x200x300 mm

**Number of glued units:** 2

**Height:** 466 mm

**Inner hole diameter:** 60 mm

**Design:** Alternating waves+ (34)





## Conclusion

What started out as a lack of confidence when doing a robotic fabrication project, grew the further we progressed in the project and successes accumulated. We experimented with setups and jigs, learned new skills and became acquainted with new, enthusiastic and nice people, passionate about their field of work. This project has certainly opened our eyes to robotic fabrication. The journey of creating the four vases has proven to be rich in knowledge and has enhanced our understanding of robotic fabrication as a design tool.

As a final remark we want to acknowledge the value of the early and comprehensive desk research that helped us understand and examine potential disadvantages when doing robotic fabrication. It was crucial for our quick progression to be aware of other early applications of robotic fabrication within the context of product design in the creative industry.

This desk research can be divided into two papers:

“Robotic Carpentry - Product Design Related Potentials of Robotic Fabrication in Wooden Materials”, Sebastian Oliver Dahl, 2023, and “Robotic fabrication and its implications - finding pointers as to how designers can work around potential pitfalls”, Daniel Fremming, 2023.

The first explores the application of robotic fabrication, especially in the context of product design scale in the creative industry. The paper highlights several directions robotic fabrication has proven to be successful in the last decade, thus giving a forecast about a rise in popularity and usage of robotic fabrication, but also gives a relevant example of the AESTUS vase - the inspiration for our project. The last title addresses the implications of robotic fabrication through examples and explains how they can be circumvented.

# Thanks

A special thanks to Karl Jermstand and Eskild Traa Carlsen who went above and beyond to accelerate our process by sharing their experiences with the particular robot that was used.

A big thanks to Ådne Solhaug Linnerud who helped us overcome a critical error in the IRC5 Controller.

We are grateful towards all the workshop staff who have provided essential support throughout the whole project.

Thanks to Tommy Snildal and Moelven Limtre AS for their donation of wooden materials.

Last but not least, thanks to Jóhannes Sigurjónsson, Lars Tingelstad and Richardo Mayor for being excellent project supervisors who made this project possible in the first place.



