Technische Hochschule Augsburg

PbG: Polished by Gesture

Innovating Polishing with Hand-Tracking Controlled Robotics

By Felix Straub

Institution:	Augsburg Technical University of Applied Sciences
Course Instructor:	Prof. Dr. Michael Kipp
Course:	Interaction Engineering
Year:	WS2023/24

Abstract

This documentation is about "PbG: Polished by Gesture", an Interaction Engineering project with the goal to develop an innovative interaction with robots. The project is expanding a different research project, which is about creating an autonomous polishing tool.

As an inspiration the "KIPN" project at the Technical University of Augsburg, which tries to map pedestrian movements to a robot within a display window, was used. PbG aims to make robotics more user-friendly and accessible, especially for those without a programming background. By using the Leap Motion device for hand gesture tracking, users can control the robot and its attached polishing tool within a safety-assured cell to perform a polishing task.

The system offers an intuitive user experience, simplifying the complexity of robot control. However, user tests revealed challenges with precision during more complex tasks. Feedback highlighted the need for improvements in system responsiveness and user interface design to enhance the overall efficiency and user confidence.

Conclusively, while PbG shows promising way of interaction with robots, the prototype requires improvements in sensor data mapping and system responsiveness, as well as interaction design. The project lays the groundwork for future developments in intuitive gesture-based robotic controls.

Motivation

For my Master's in Applied Research, I am involved in a research project named "Collaborative Robotics in Trades". The goal of this project is to create an autonomous polishing tool with a robot that small businesses can use, especially those who cannot afford to implement a robot in their processes, due to the varying and unique products they make for their customers. So far editing the polished object after the initial processing is not possible. With the Interaction Engineering semester project, I am adding this useful feature to the research project.

During my studies with robots, I realized that without prior experience it is hard to integrate robots in the working processes. Because of that I wanted to seek out ways to make robotic technology more user-friendly and accessible for individuals who don’t have a lot of experience in programming. Robots offer numerous advantages, such as improved efficiency, enhanced safety, and creating a healthier work setting. By developing a complete application that can adapt to various products, I hope that it will encourage more small businesses to think about whether robots could be beneficial for them.

On a personal note, I'm interested in how to make robot control more intuitive. The current method of using control panels is not particularly efficient for programming new tasks for robots. I believe that by making the control of robots more straightforward, we can open new possibilities for their use in various industries.

Related Work

An ongoing research project called “KIPN” at the Technical University of Augsburg was used as an inspiration for the project. The goal of this project is to capture the movements of pedestrians and create a natural mapping between them and a robot, which is positioned inside a display window. This project aims to make research in general more accessible and approachable by directly showing the results to the public.

Concept

PbG is a system that utilizes hand gestures to enable users to easily control a robot. By moving and turning the hand the user can move the robot to different positions of the robot cell or set the orientation of the last robot joint. Furthermore, the user can use different gestures to set up different parameters which are important for the polishing process.

Setup

The main part of the system is the robot. For the prototype a Universal Robot 10e is used. On the robot tool holder, a OnRobot Sander tool is mounted for the polishing as well as a suction device to make sure the polishing dust is removed from the object. The robot is placed inside a robot cell to ensure a safe operation. Inside of the cell the objects to be polished are placed. In figure 1 a letter made from aluminium in the shape of a “C” is mounted as an example.

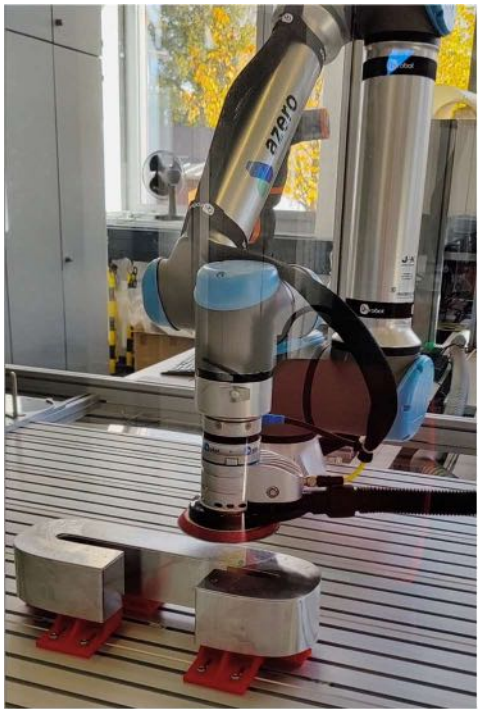

Figure 1: robot cell with object to be polished

The system utilizes a Leap Motion device to track the hand movement and to recognize different gestures. The device is placed right in front of the robot cell at an elevated position. This ensures that during the operation the robot is fully visible, as well as a comfortable hand movement position inside of the tracking zone of the Leap Motion. In figure 2 the device is

also placed right in front of the teach panel with an e-stop. This lets the operator stop the robot system at any time, in case of an emergency.

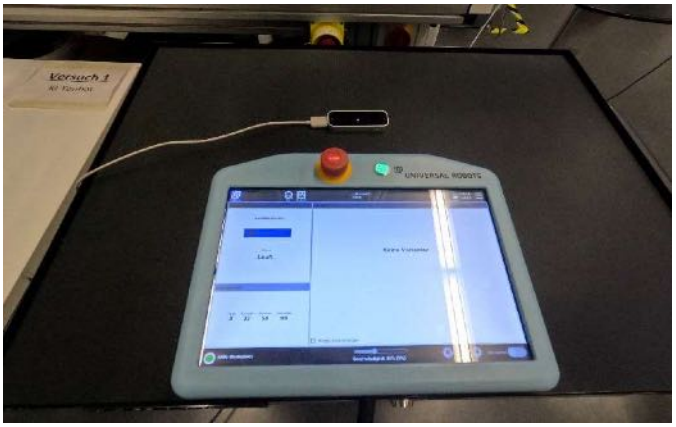

Figure 2: Leap Motion and Teach Panel

Right above the Leap Motion device in front of the robot cell there is a screen mounted to display the visual feedback of the system. At this position it is possible for the operator to look at the feedback screen to obtain the needed information without losing sight of the moving robot for too long.

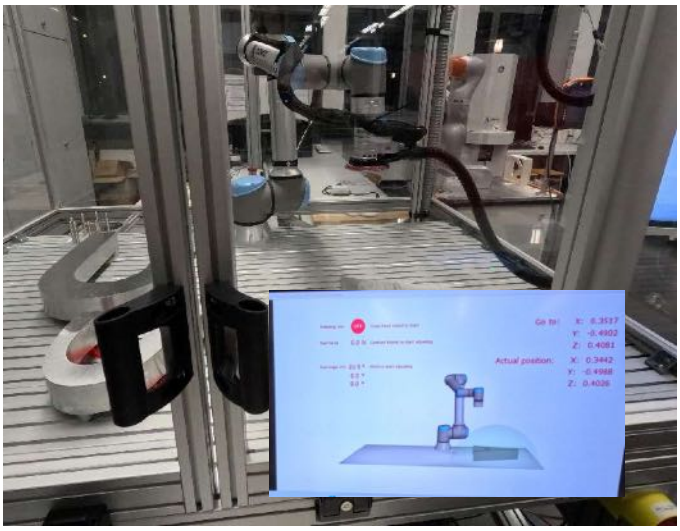

Figure 3: screen for visual feedback

A workstation, which can be seen in figure 4, is used to run the required software during the operation.

Figure 4: workstation

Interactions

The typical scenario of the system starts with the operator looking at the polished object and finding a spot which he does not like. Then he would mount the object back inside of the robot cell and start the PbG-system.

Base position

The robot is provided with a base position, which he positions itself at, whenever no hand is detected from the Leap Motion. The base position is a central position in the cell with enough distance to every object. With that a safe start of the hand tracking is always possible.

Robot movement

To start the robot movement the hand is to be placed at a central position above the Leap Motion sensor. There are three different directions to which the robot can move. The axis are shown in figure 5.

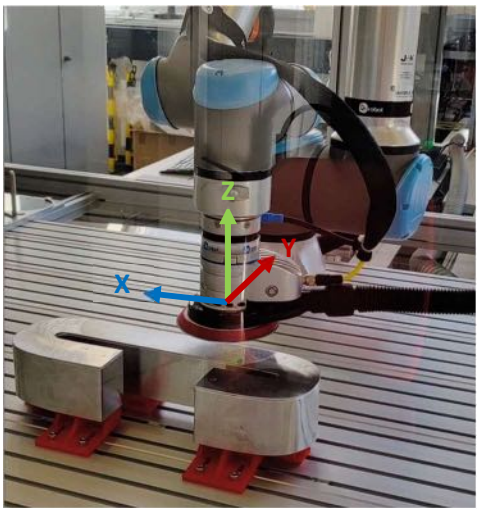

Figure 5: robot axis

To move along the x-axis the hand must be moved either left or right, to move along the y axis the hand must be moved either to the front or back and to move along the z axis the hand must be moved either up or down (see figure 6).

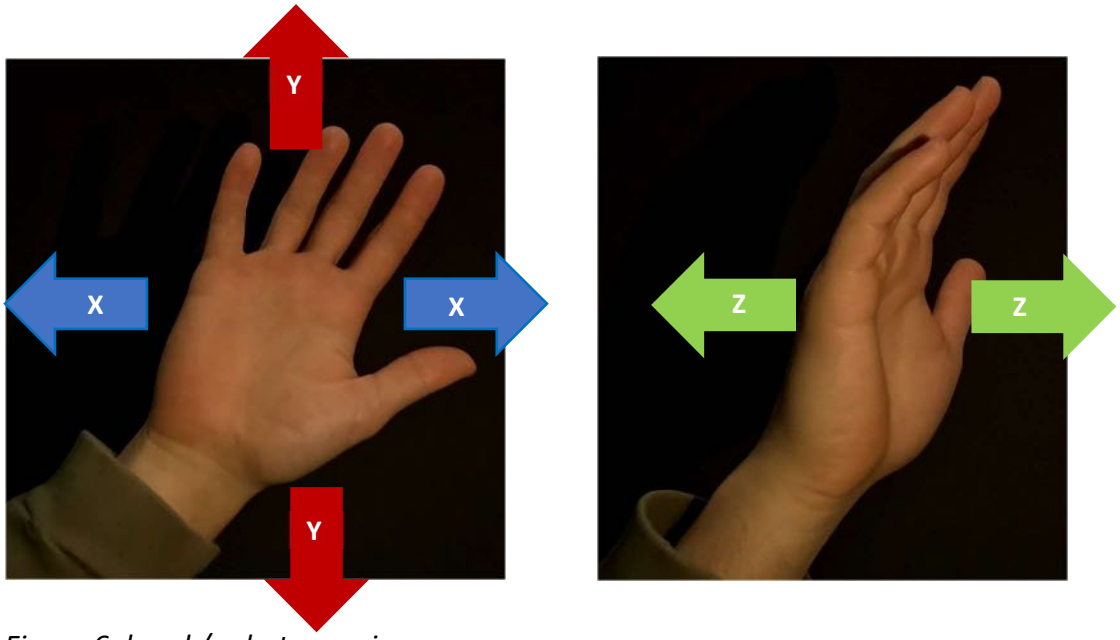

Figure 6: hand / robot mapping

Polishing Tool

Since the polishing tool cannot run the whole time due to loud noise and overheating issues, a gesture that activates or deactivates the tool is implemented. By closing the hand without the thumb (figure 7) the tool is activated when turned off or deactivated when running.

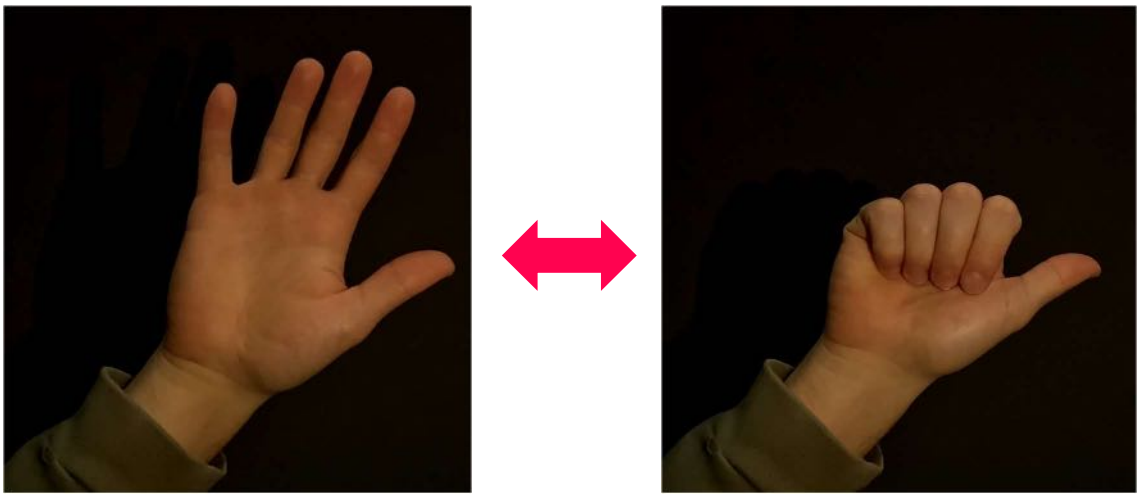

Figure 7: polishing tool activation

Tool Force

Another important parameter of the tool is the maximum force with which the robot can press onto the surface of the object to polish. To keep it at a stable level during operations the setup is triggered by a defined gesture (contract thumb) which can be seen in figure 8.

Figure 8: setup trigger tool force

To select the value of the tool force in between a preset range, the hand can either be fully closed to decrease the value with a preset decrement or fully opened to increase the value with a preset increment. In a neutral position of the hand the value stays constant (figure9). To set the value of the tool force the same gesture as before, seen in figure 8, is used.

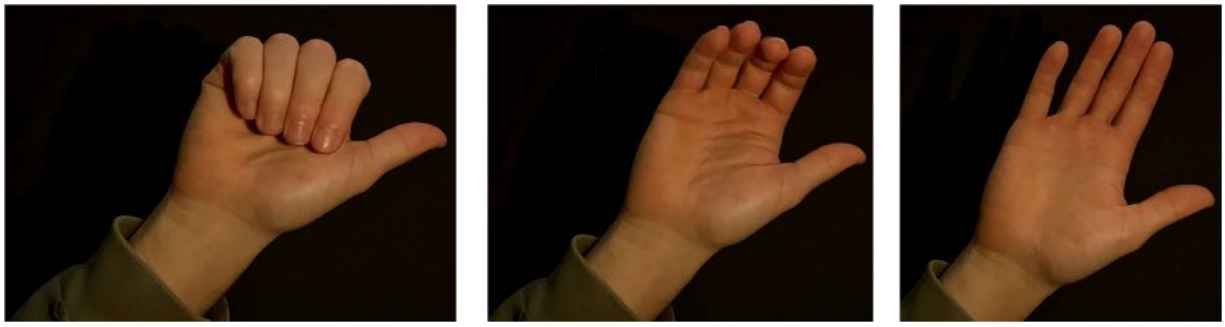

Figure 9: Decrement, Neutral, Increment

The decrementing gesture is similar to the activation gesture of the tool. Since the tool must be turned off in order to set the force value, that is not a problem. As soon as the setup of the force is triggered, the activation of the tool is not possible, vice versa.

Tool Angle

The rotational angle of the polishing tool can be set as well. As a first try the hand rotation around every axis was directly mapped to the rotation of the tool. This led to several problems though. Due to the instability of the hand a stable position of the tool was not possible anymore, hence the whole polishing process did not work properly. Therefore, the setup of the angle also must be triggered by a pinch-gesture, seen in figure 10.

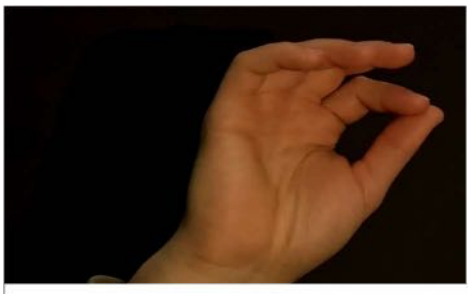

Figure 10: Tool Angle Setup Trigger

Since the hand rotation around one axis might influence the rotation around another axis the three axes are set separately. After the initial trigger through the pinch gesture, at first the x-axis can be set, after a second pinch, the y-axis can be set, after a third pinch the z-axis can be set and after the fourth pinch the values are confirmed.

The directions of the natural hand rotation are used to rotate the robot tool around its three axes (see figure 5).

Figure 11: Rotation Y-Axis: Increment, Neutral, Decrement

Figure 12: Rotation Z-Axis: Increment, Neutral, Decrement

Figure 13: Rotation X-Axis: Increment, Neutral, Decrement

Visual Feedback

Additionally, the system provides a visual feedback screen. On there every parameter to set is displayed (see figure 14). Every setting has its current value and an additional description what to do to change it. Triggering a gesture is signalled by a green bar in the background (see figure 18).

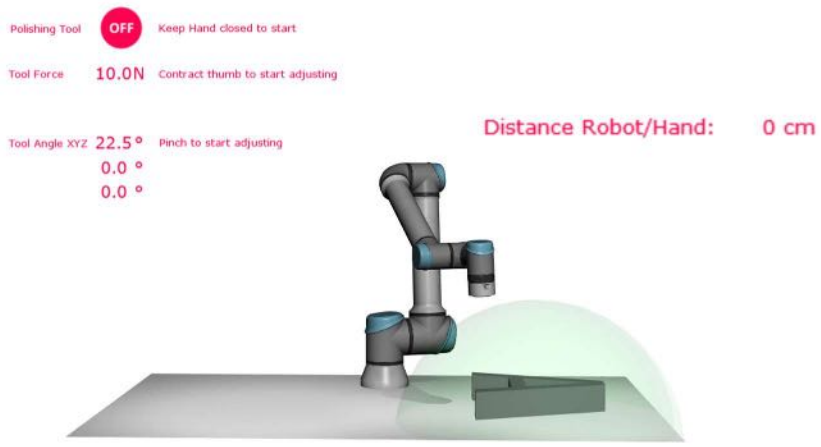

Figure 14: Visual Feedback System

As soon as a hand is detected by the Leap Motion device, the position, where the robot is supposed to go, is displayed in the simulation, indicated by the orange hand. The distance between the old and the new position is then calculated and displayed in the top right corner. The direction, which the robot arm is going to move, is indicated by a coloured arrow. With that feature the user always knows to which position the robot is going to next (see figure 15).

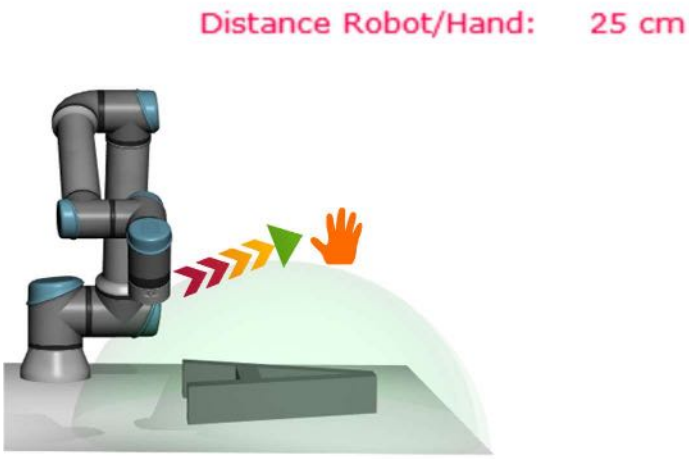

Figure 15: Hand Position Indication

For the polishing tool there are two different states, off or on (see figure 16).

Figure 16: Visualization Polishing Tool

The tool force setting displays the current force value as well as instructions on how to change it (see figure 17).

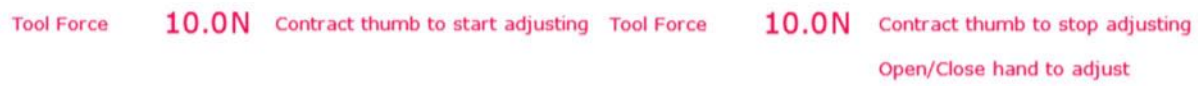

Figure 17: Visualization Tool Force

The tool angle setting displays the current angle values as well as instructions on how to change them (see figure 18).

Figure 18: Visualization Tool Angle

In general every object is surrounded with a translucent green sphere. As soon as the polishing tool gets into close proximity of the object the sphere turns red to signal that high caution is needed, in order to prevent a crash of the robot (see figure 19).

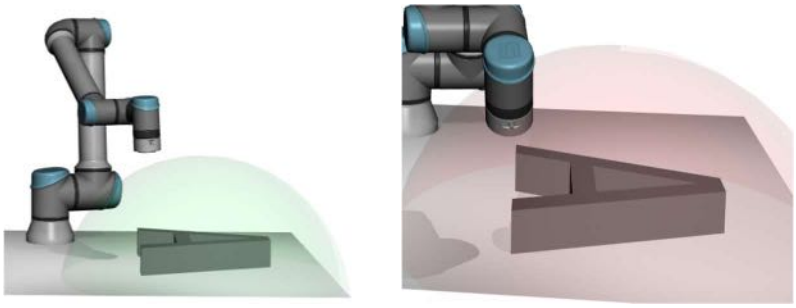

Figure 19: Visualization Precission Zone

Implementation

Hardware

To run the project the user needs a UR10e robot, a LeapMotion device and a powerful Windows computer, that runs all the necessary software, as well as a display that can be mounted on to the robot cell.

Software

The project consists of different programmes, which require different software:

- Touchdesigner 2023
- UltraLeap HandTracking software
- VSCode with Python Extension
- Python V3.10.xx (e.g. with Anaconda) and packages ur-rtde, python-osc and mido

- Oracle VirtualMachine with URSim setup (for robot simulation)

Concept

Touchdesigner is used for processing the LeapMotion data and the robot data, as well as providing the visualization and simulation. With a predefined library CHOP the sensor data provides all the necessary information about the hand movement (see figure 20). By combining different signals, processing them with math or logic functions or applying other steps, the gestures can be recognized by the system.

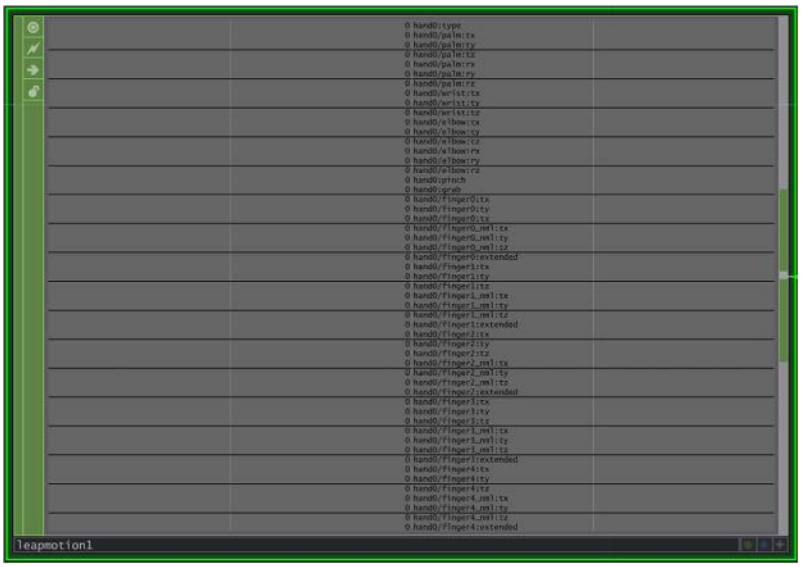

Figure 20: Leap Motion CHOP in Touchdesigner

With a Python script and the OSC and RTDE protocols the communication between Touchdesigner and PolyScope - the software of the UR robots – is established. There are two different main scripts. In RTDE_OSC_read the robot provides information about the current position to Touchdesigner. This information is necessary for the robot simulation. With RTDE_OSC_write Touchdesigner provides the robot software with the calculated positional values, which the robot should head for, as well as different important commands to control it.

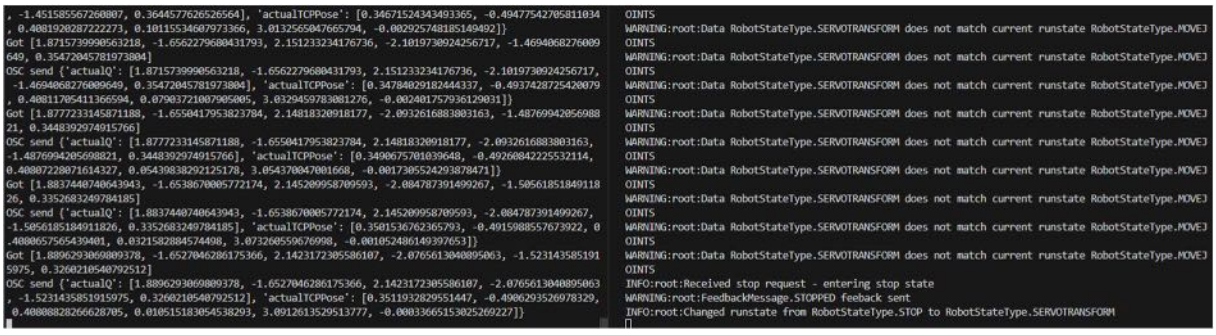

Figure 21: Python Scripts

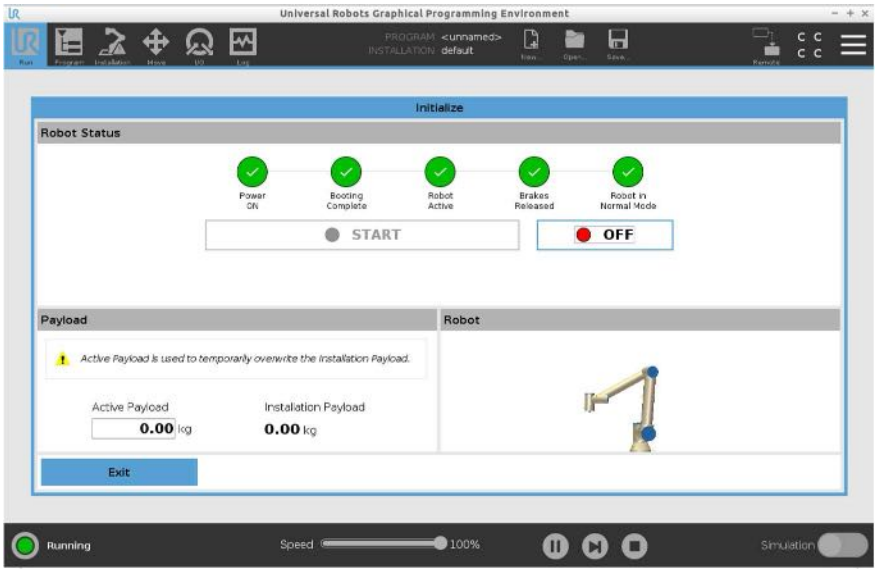

Figure 22: UR PolyScope

User Test

Setup

For a user test five people with different experience in robotics were asked to carry out three different tasks during the session. In task one they had to move from a position A to a position B, outside of the precision zone. The second task was to set the tool force to 10N and the y-angle to 45°. In a third task the participants had to move into the precision zone, turn on the polishing tool, touch the letter and move back to the base position of the robot.

General Feedback on Intuitiveness

Users generally found the gesture controls to be intuitive, especially for simple movement commands. The 'plug and play' aspect was appreciated, indicating a user-friendly approach to operating the robot. However, there were concerns about the intuitiveness of more complex controls, such as maintaining a constant height during the polishing process.

Fatigue

Some users reported a degree of fatigue after using the system for an extended time. This was primarily due to the physical demand of maintaining the position and gestures or the stress of ensuring the robot does not cause any damages.

Challenges in Gesture Tracking

The most challenging aspects reported by users were related to the setup task where they had to set up the tool force and angle. Users also noted that the system's response could sometimes be laggy or 'jumpy', leading to difficulties in performing precise adjustments.

Comparison with Traditional Methods

Users had mixed feelings about whether they would prefer gesture controls over traditional methods. Some found gesture controls to be better if further improved, while others,

especially those with programming backgrounds, still preferred traditional programming methods for their precision and reliability.

System Responsiveness and Reliability

Instances of unexpected robot behaviour were reported, such as delayed movements or uncontrolled feedback values, which sometimes caused anxiety. One user experienced an inverse kinematics failure, indicating a need for further improvements on the system to ensure reliability.

Suggestions for Improvement

The most common suggestion for improvement was to increase the precision and responsiveness of the gesture tracking interface. Users indicated a preference for a more continuous mapping of gestures to actions, rather than having to hold a gesture to increment values.

Control Improvement

While general movements were handled with a lot of confidence, different tasks such as angle adjustments made users less confident. Familiarity with the system seemed to improve confidence levels over time, suggesting that with practice, users might become better at operating the system.

Summary

The user test revealed that while the gesture control system is promising, there are areas that require improvement. Key issues include the need for more precise and responsive control and improved reliability to prevent unexpected robot behaviour.

Conclusion

The use of hand gestures for interacting with a robot was found to be an intuitive concept with potential for further research. The prototype was perceived as interesting to use and possible use case scenarios indicate a demand for similar systems.

Results from a user test show that there are still many improvements necessary to enhance stability, safety and user-friendliness of the system. As a next step the mapping of the sensor data to the robot movement could be reworked to achieve more stable movements as well as a more responsive system.

Overall PbG offers an interesting approach to an intuitive robot interaction and is considered an overall successful implementation of the prototype.