Underground Mining Safety Innovation

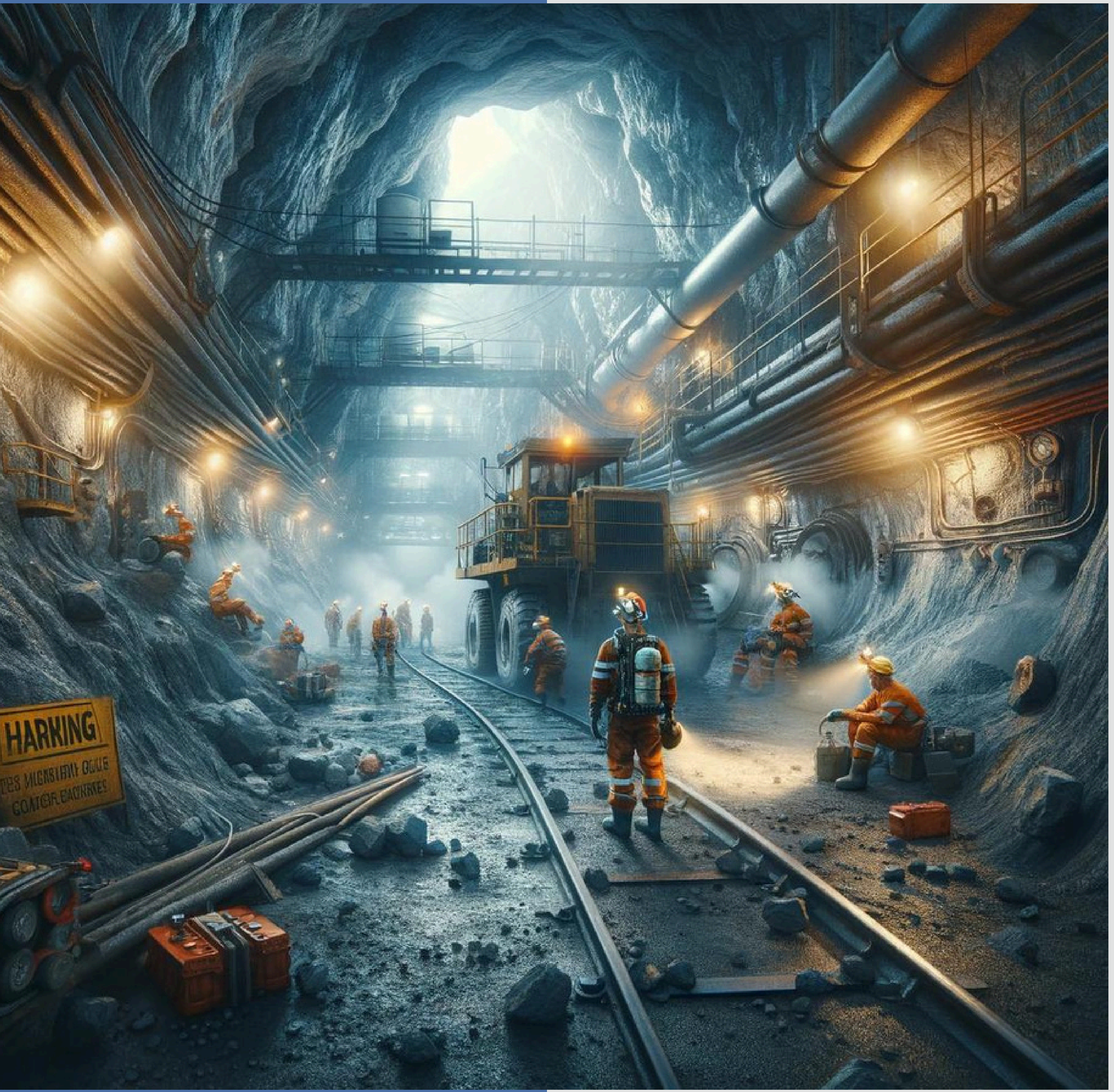
KSL Sanjana - ME21B1015
Sibi M - EC21B1042
Jhansi Rajeswari Athota - EC21B1054

Introduction

In the demanding realm of the mining industry, ensuring the safety of operations poses a multifaceted challenge:

1. **High-Stress Environments**: Mines are inherently high-stress workplaces where errors can have severe consequences. There's a critical need for systems that can withstand extreme conditions and provide reliable performance.
2. **Complex Safety Scenarios:** The dynamic nature of mining operations means that traditional safety protocols may not suffice. Workers face a variety of risks that require more nuanced and responsive safety measures.
3. **Technological Shortcomings:** Conventional safety equipment, like proximity sensors, often provides limited response times, potentially failing to prevent accidents. This highlights the necessity for more advanced warning systems.
4. **Inadequate HMI Design:** Human-Machine Interfaces (HMI) that aren't tailored to the users' needs can impede rather than aid operations, adding complexity and increasing the potential for errors in high-risk situations.

To enhance miner safety and operational efficiency, a shift towards more sophisticated and adaptive safety technologies is imperative. These should not only preempt risks but also harmonize with the miners' work patterns, creating an intuitive and failsafe environment.



Problem Statement

- Mining's high-stress setting demands advanced safety beyond traditional measures.
- Proximity sensors offer limited reaction time, necessitating enhanced safety solutions.
- Existing Human-Machine Interfaces fails to adapt to miners' needs to mitigate risks effectively.
- Traditional safety measures fall short in post accident scenarios
- Minimal error margin underscores the need for evolved safety approaches in the mining industry.

Tentative Timeline

January	February	March	April

Concept Initiation
Initial brainstorming.
Detailed study of possible prototypes.

Component Development
Design of capacitive sensors.
Development of CAD models and electronic components

Assembly and Integration
3D printing and assembly of brace components.
Sensor integration with haptic feedback system.

Testing, Refinement, and Presentation
Testing and refinement of device functionality.
Final adjustments, system integration tests, and machine learning model refinement.
Documentation, final prototype demonstration, and project presentation.

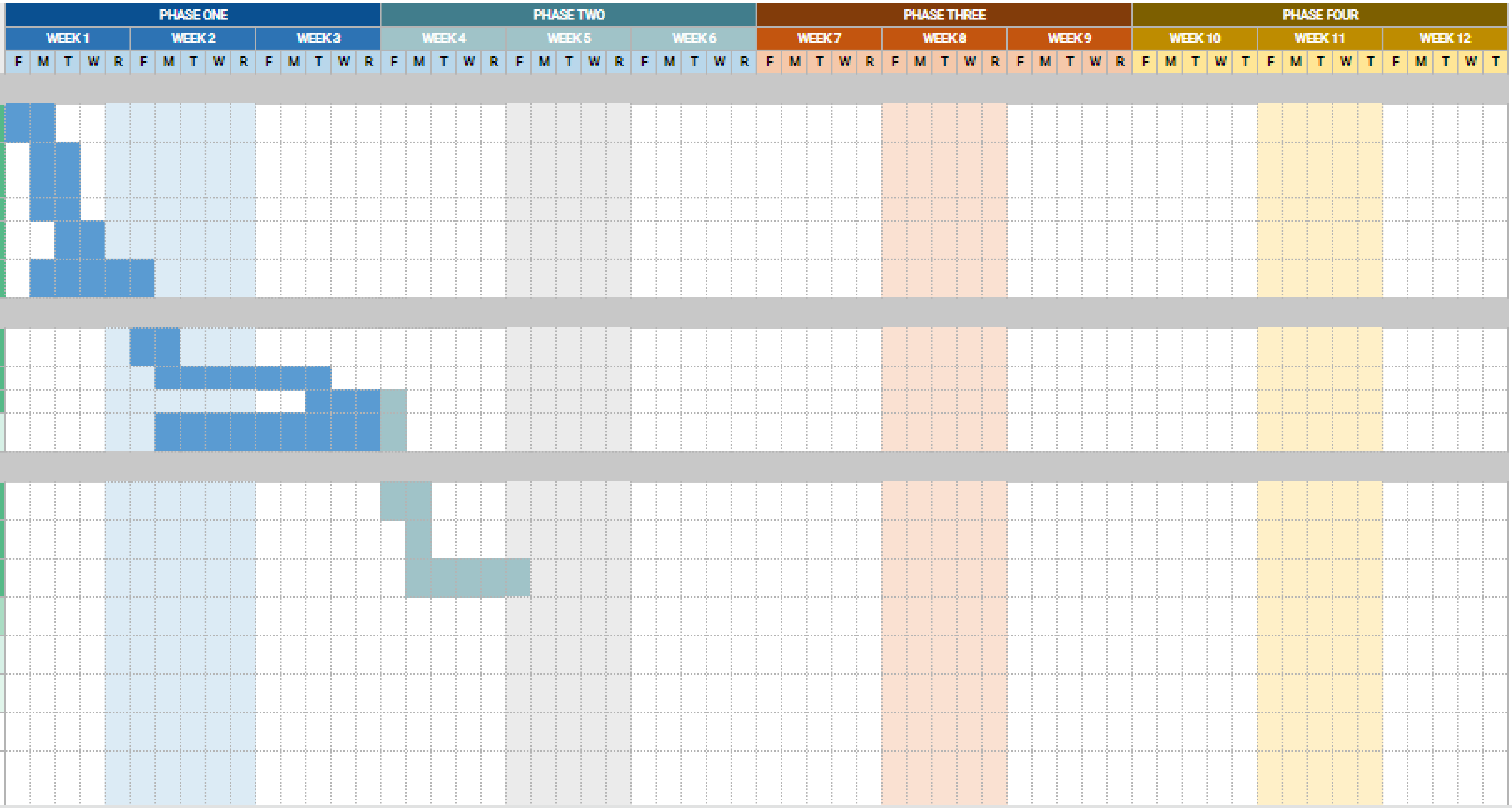
Gantt Chart Document:

https://docs.google.com/spreadsheets/d/1X7XD6QCdY0D95djrMoF0Rg6L8jy70uq7DNddNUpApJY/edit?usp=sharing

GANTT CHART

PROJECT TITLE	Mining Safety Innovation	TEAM NUMBER	F-14
TEAM	KSL Sanjana (ME21B1015), Sibi M(EC21B1042), Jhansi Rajeswari Athota(EC21B1054)	DATE	2/19/24

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Concept Ideation and Finalization**					
1.1	Finalizing domain and concept ideation	Everyone	2/9/24	2/10/24	1	100%
1.2	Identify & finalize problem statement to obtain a minimum viable product	Everyone	2/10/24	2/12/24	2	100%
1.3	Finding solutions	Everyone	2/10/24	2/12/24	2	100%
1.4	Defining main objectives for the solution	Everyone	2/11/24	2/13/24	2	100%
1.5	Extensive Literature Survey for concept validation	Everyone	2/12/24	2/16/24	4	100%
2	**Initial Design and Solution clarity**					
2.1	Competitor Analysis	Jhansi, Sanjana	2/16/24	2/19/24	4	100%
2.2	CAD Designs for solutions	Sanjana, Sibi	2/19/24	2/26/24	3	100%
2.3	Circuit Diagrams	Sibi, Jhansi	2/26/24	3/1/24	5	100%
2.4	Research on Testing and Validation for subsystems incorporated	Everyone	2/19/24	3/1/24	12	22%
3	**Product Finalization, BoM and Procurement**					
3.1	Finalizing on the solution, CAD design and circuits	Everyone	3/1/24	3/4/24	3	100%
3.2	Research and shortlist potential sensors based on desired functions	Sibi, Jhansi	3/2/24	3/4/24	2	100%
3.3	Generate BoM and identify suitable vendors	Sibi, Jhansi	3/3/24	3/9/24	5	100%
3.3.1	Contact vendors for sensor specifications and quotes	Jhansi	3/4/24	3/15/24	11	50%
3.3.2	Finalize sensor selection and Procuring Items	Sibi, Jhansi	3/12/24	3/13/24	1	22%
3.4	Starting the initial phase of building the prototype	Everyone	3/14/24	3/16/24	2	16%
3.5	Modeling and Simulating the hardware	Sanjana	3/4/24	3/16/24	12	0%
3.6	Software Development Kickoff (Sensor Data Processing and mapping states)	Sibi, Jhansi	3/10/24	3/16/24	6	0%



4	**Initial Prototyping Phase**					
4.1	Hardware (Circuit, PCB, Mechanical design) Development	Everyone	3/16/24	3/20/24	4	0%
4.2	Hardware Interfacing	Everyone	3/17/24	3/21/24	4	0%
4.3	GUI and UX Development	Everyone	3/18/24	3/21/24	3	0%
4.4	Final Testing & V1 prototype completion	Everyone	3/21/24	3/22/24	1	0%
5	**Prototyping Phase**					
5.1	Prototype completion	Everyone	3/22/24	3/26/24	4	0%
5.2	1st level demo of assembled product	Everyone	3/27/24	3/28/24	1	0%
5.3	Preparing for Ehipassiko	Everyone	3/22/24	3/28/24	6	0%
6	**Product FInalization**					
6.1	Full product demo	Everyone	4/4/24	4/5/24	1	0%
6.2	Patent feasibility analysis	Everyone	3/28/24	4/1/24	3	0%
6.3	Submission of pending bills	Sanjana	4/1/24	4/4/24	3	0%
7	**Final Report and Patenting**					
7.1	Final report and PPT preparation	Sanjana, Jhansi	4/4/24	4/19/24	15	0%
7.2	Patent writing completion of the selected products	Sibi	4/5/24	4/19/24	14	0%
8	**The End**					
8.1	Final review and report submission	Everyone	4/19/24	4/26/24	7	0%

Roles and Responsibilities

- KSL Sanjana (ME21B1015): Led the design and development of the haptic glove, focusing on mechanical integration and durability testing. Further, is also the overall lead and the project manager.
- Sibi M (EC21B1042): Managed the HMI development and integration, worked on the glove's responsiveness and control accuracy. Also oversees all Technical aspects of the product
- Jhansi Rajeswari Athota (EC21B1054): Spearheaded the electronic components selection and circuit design, ensuring optimal performance and compatibility. Is also responsible for communicating with external parties like vendors and thorough documentation.

WEEK 1–2 (FEB 9, 13, 16): CONCEPT IDEATION AND FINALIZATION

Main Goals:

- Finalizing domain and concept ideation (Mining Safety Technology)
- Identify & finalize problem statement to obtain a minimum viable product
- Finding solutions (Haptic gloves/braces)
- Defining main objectives for the solution
- Extensive Literature Survey for concept validation

Subtasks

Problem Identification

- Analyze causes of accidents in mining.
- Assess the cost and scale of mining accidents.
- Evaluate commercial gains from addressing these problems.
- Narrow down a specific problem to focus on as the project's driving force.

Solution Analysis

- Research existing solutions for mining safety.
- Understand the barriers to the implementation of these solutions.
- Decide on the scope of the solution the team can develop.

Concept Finalization

- Determine the problems/features customers need solutions for.
- Identify the target market based on commodity, mine size, and whether it's surface or underground mining.
- Compile the number of current installations, if applicable.

Technology Adoption Challenges

Discuss the major hurdles in adopting new technology within the industry.

Underlying Technology

- Detail the underlying technologies being used, noting if they are new or adapted from other industries.
- Interoperability and Open Data
- Confirm if solutions support competitor interoperability and open data.

Safety and Research

- Clarify which specific safety issues the technology addresses.
- Identify any research gaps.
- Outline current R&D tasks being undertaken.

Literature Survey on Mining Accidents and Safety Technologies

Causes of Mining Accidents

- Fall of roof/sides/persons/objects in underground mines.
- Accidents involving dumpers, trucks, tankers, and other vehicles.
- Rope haulage, explosives, and machinery-related incidents.
- Gas and dust explosions, often due to the ignition of accumulated dust and gas.
- Lack of trained on-site rescue workers and the destruction of miner identification systems.

Regulatory Framework

The Coal Mines Regulations, 2017; the Metalliferous Mines Regulations, 1961; and the Oil Mines Regulation, 2017, under the Mines Act, 1952, mandate provisions for personal protective equipment (PPE), including protective footwear, glove, helmets, self-rescuers, and other PPEs, to be provided free of cost by mine owners, agents, and managers.

Economic Impact and Scale

The financial implications of mining accidents are significant, with costs including exploration, price of technical reports, and direct accident-related losses.

Recorded incidents have resulted in material losses and human casualties, with substantial compensation and insurance claims.

On a global scale, nearly 9950 lives have been lost in US and India coal mines from 1990 to 2019, with the economic burden reaching about 500,000 crores per year.

Literature Survey on Mining Accidents and Safety Technologies

Safety Interventions and Technologies

- Implementation of safety provisions is monitored through inspections by authorities like DGMS.
- Advanced safety technologies, including telematics, predictive analytics, wearable devices, and autonomous operations, have been identified as key areas for innovation to enhance miner safety and operational efficiency.
- Integration and interoperability among safety systems, cost-effective solutions, improved data management, and regulatory compliance are crucial for advancing mining safety.
- Guidance for operability in post-accident situations.

Future Directions

- Enhancing human-machine interface issues, including human performance and machine control, to prevent accidents.
- Leveraging integration across diverse systems for comprehensive risk management.
- Investing in predictive analytics, machine learning, wearable technology, and enhanced collision avoidance systems.
- Expanding autonomous and remote-controlled operations to reduce human exposure to hazardous conditions.

Conclusion

The literature underscores the critical need for ongoing research, development, and implementation of innovative safety technologies and practices in the mining industry to mitigate risks, reduce the frequency of accidents, and safeguard the welfare of miners.

Literature Survey on Mining Accidents and Safety Technologies

- http://www.ismenvis.nic.in/Database/Mining_Accidents_in_India_24483.aspxhttps:/sansad.in/getFile/loksabhaquestions/annex/1711/AU1619.pdf?
- source=pqals#:~:text=The%20possible%20cause%20of%20the,machinery%20and%20gas%2C%20dust%20etc.
- https://www.cdc.gov/niosh/mining/UserFiles/works/pdfs/IC7493.pdf%20
- https://www.sciencedirect.com/science/article/pii/S0925753516303769%20
- https://www.wboy.com/only-on-wboy-com/the-deadliest-mine-disaster-in-u-s-history-was-in-west-virginia/%20https://www.mdpi.com/2313-576X/6/1/3
- https://www.ndtv.com/india-news/uttarkashi-tunnel-rescue-arnold-dix-uttarakhand-tunnel-news-softly-softly-how-tunnelling-expert-arnold-dix-rescued-41-workers-4616893
- https://www.ndtv.com/world-news/12-dead-13-injured-in-coal-mine-accident-in-northeast-china-report-4714623%20https://www.sciencedirect.com/science/article/pii/S1878522015001071%20
- https://www.indiastat.com/Home/DataSearch?Keyword=Mines%20Accidents%20
- https://www.cdc.gov/niosh/mining/researchprogram/strategicplan/haultruckroadmap2020.html

WEEK 3-4(FEB 23, MAR 1): INITIAL DESIGN AND SOLUTION CLARITY

Main Goals:

- Competitor Analysis
- CAD Designs for solutions
- Circuit Diagrams
- Research on Testing and Validation for subsystems incorporated

Subtasks

Choose which solution to proceed with!!

Testing and Validation
Document testing and validation procedures performed on the overall solution and subsystems.

Technical Specifications
- Specify the type of haptics required.
- Discuss issues with current existing haptic gloves.

Circuit Design & CAD Modeling
- Create detailed circuit diagrams for the haptic device's electronics.
- Design 3D CAD models for all device components, ensuring proper fit and function within the glove and braces structures.

Differentiation and Signage
- Define how to differentiate between heavy machinery in a mining context.
- Determine what signs need to be incorporated into the design for safety and identification.

Overcoming Industry Adoption Challenges
- Document and plan for the challenges in adopting new technology, potentially using CAD models to demonstrate the solution's integration within existing systems.

Interoperability and Open Data Compliance
- Model interfaces and connection points that support interoperability and data sharing standards, ensuring the CAD design facilitates these aspects.

Comprehensive CAD Design Review
- Perform an exhaustive review of all CAD designs, cross-checking against requirements for functionality, safety, and interoperability.

Proposed Solution

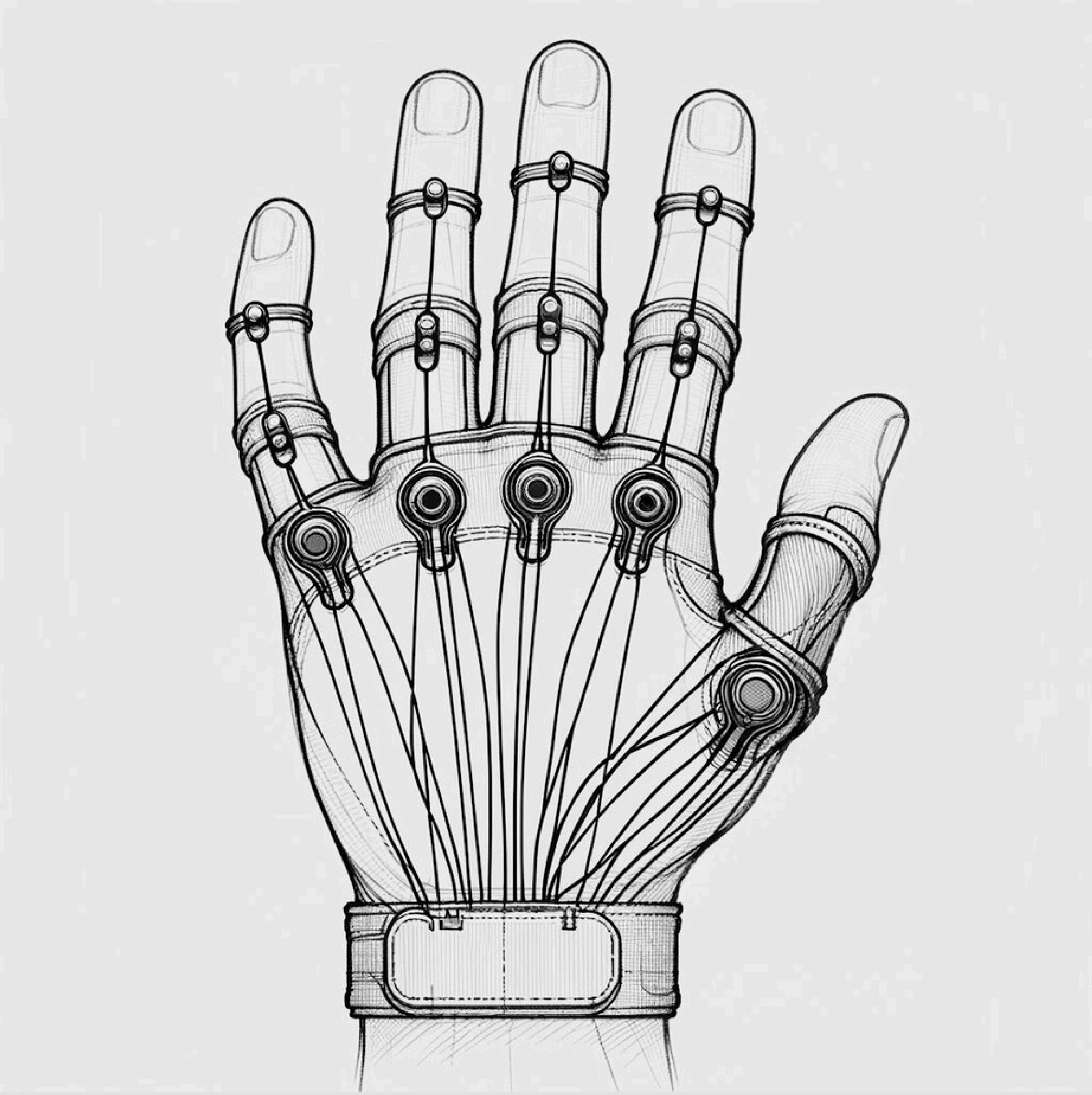
Problem

Conventional haptic devices are often complex, with numerous sensors that can be exposed to hazardous conditions, making maintenance challenging and costly.

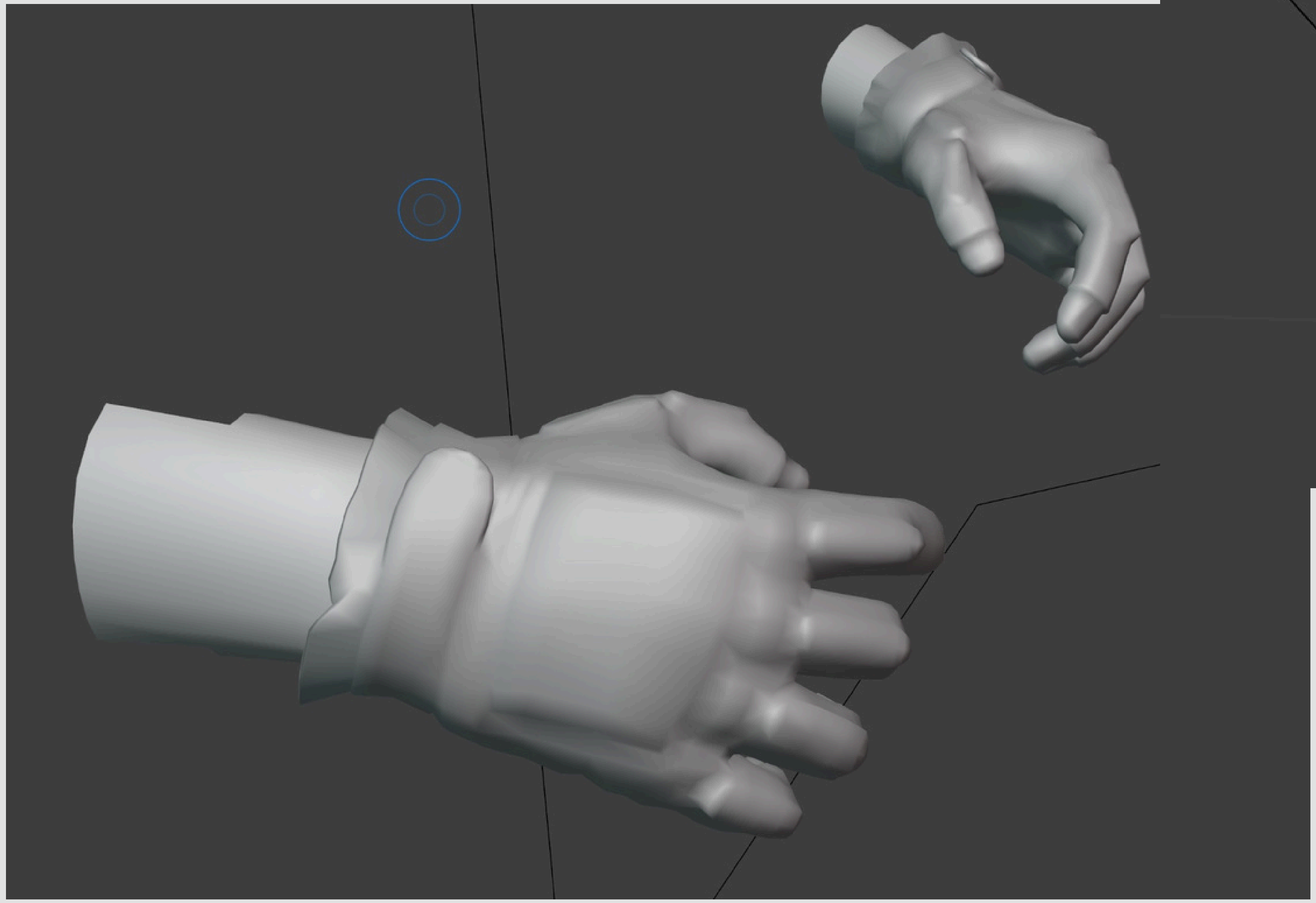
Solution: Unconventional Haptic Glove

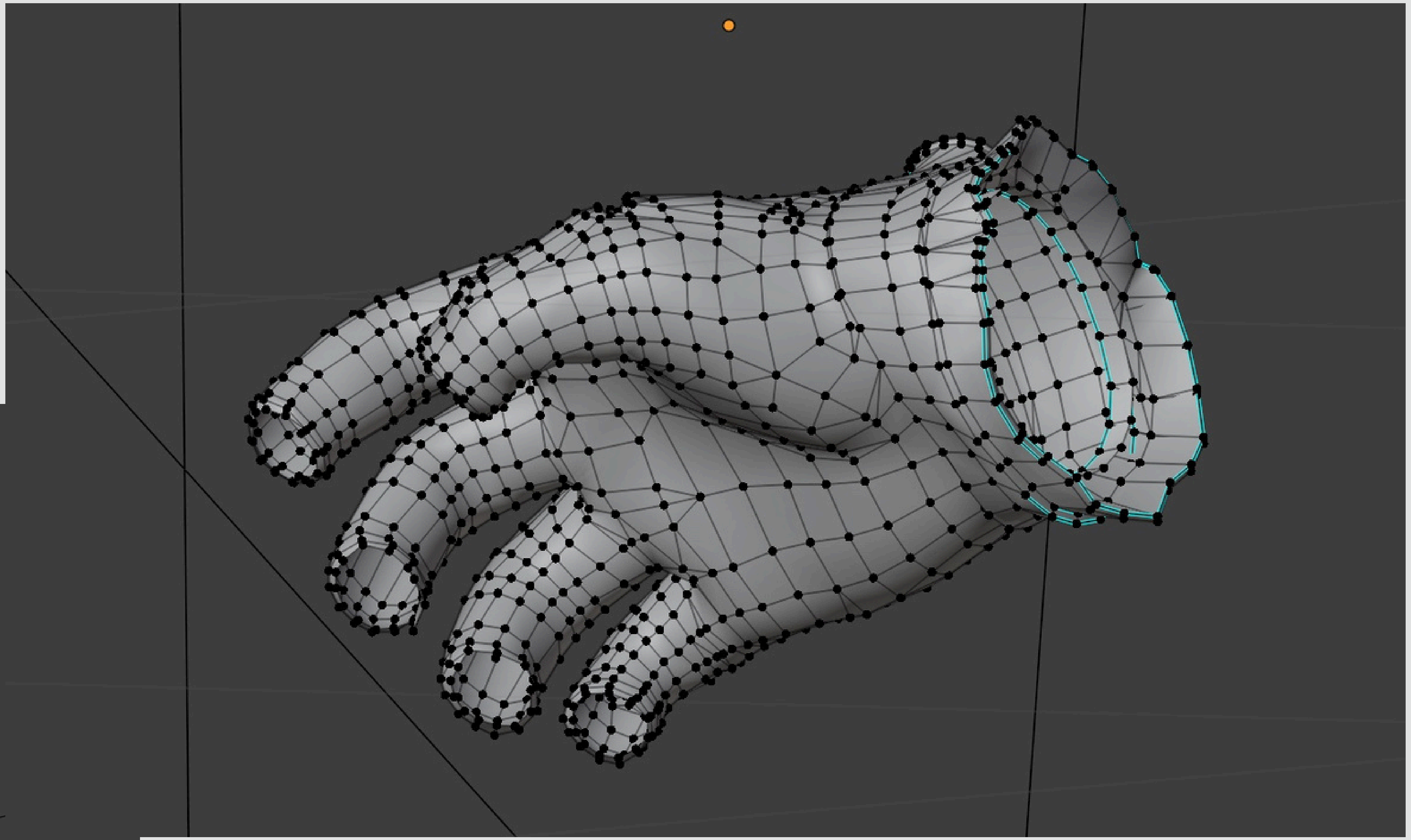
- **Minimalist Design**: A glove-type haptic device with a minimal number of exposed sensors, reducing vulnerability and maintenance.
- **Centralized Sensing:** All sensors are concentrated in one location for easier maintenance and protection from potential damage.
- **Tension-Based Sensing:** Each finger is equipped with a tension-based sensor connected to a string, which joins at a central point.
- **Machine Learning Integration:** Utilizes Reinforcement Learning (RL) or Machine Learning (ML) to interpret tension patterns for each finger and hand placement.
- **HMI Output Control:** The sensory input is converted to modulate the Human-Machine Interface (HMI) output, which will be determined in subsequent design stages.

Rough Sketch



Model





Proposed Solution

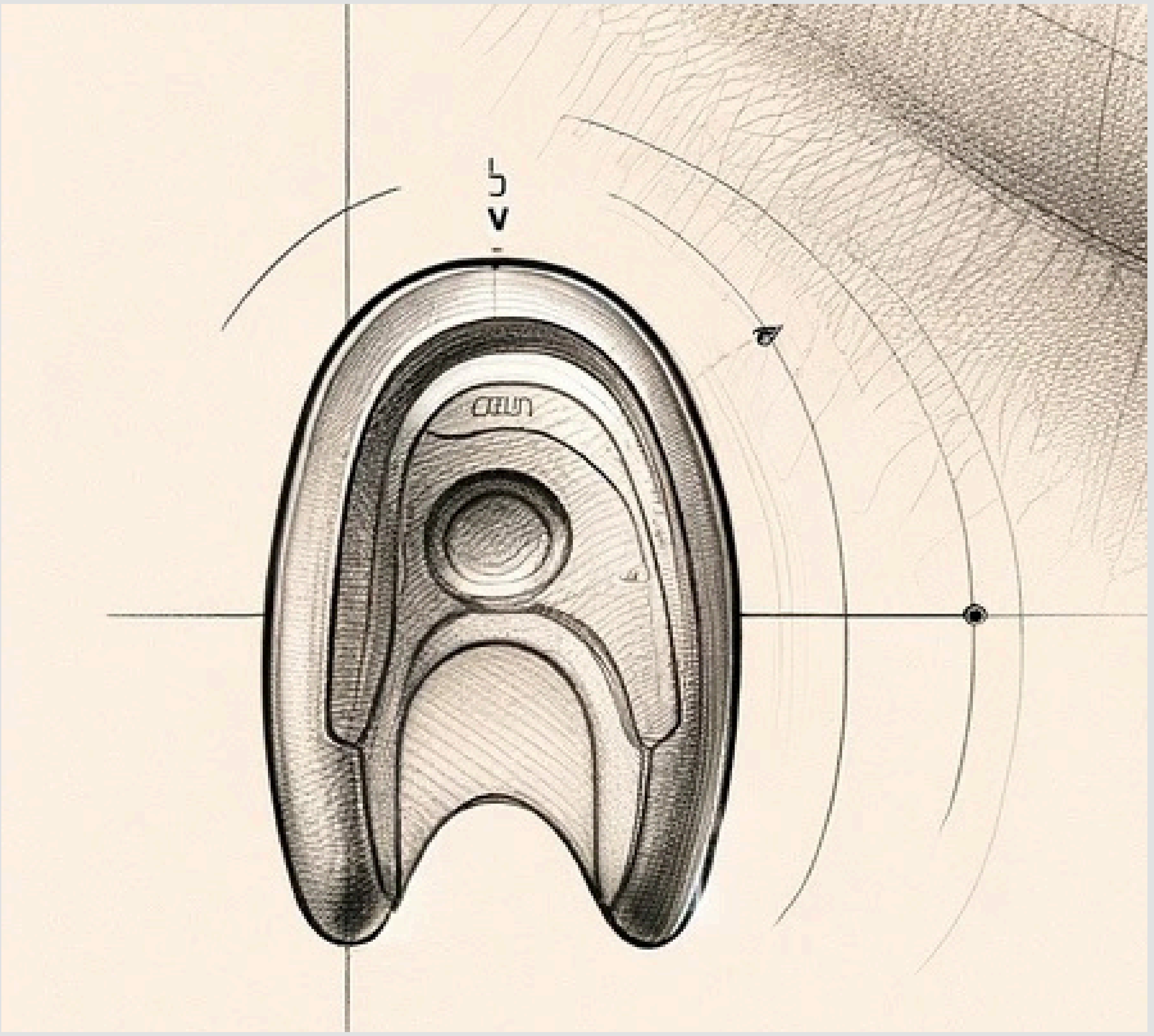
Alternative Solution: Inner Braces-Inspired Device

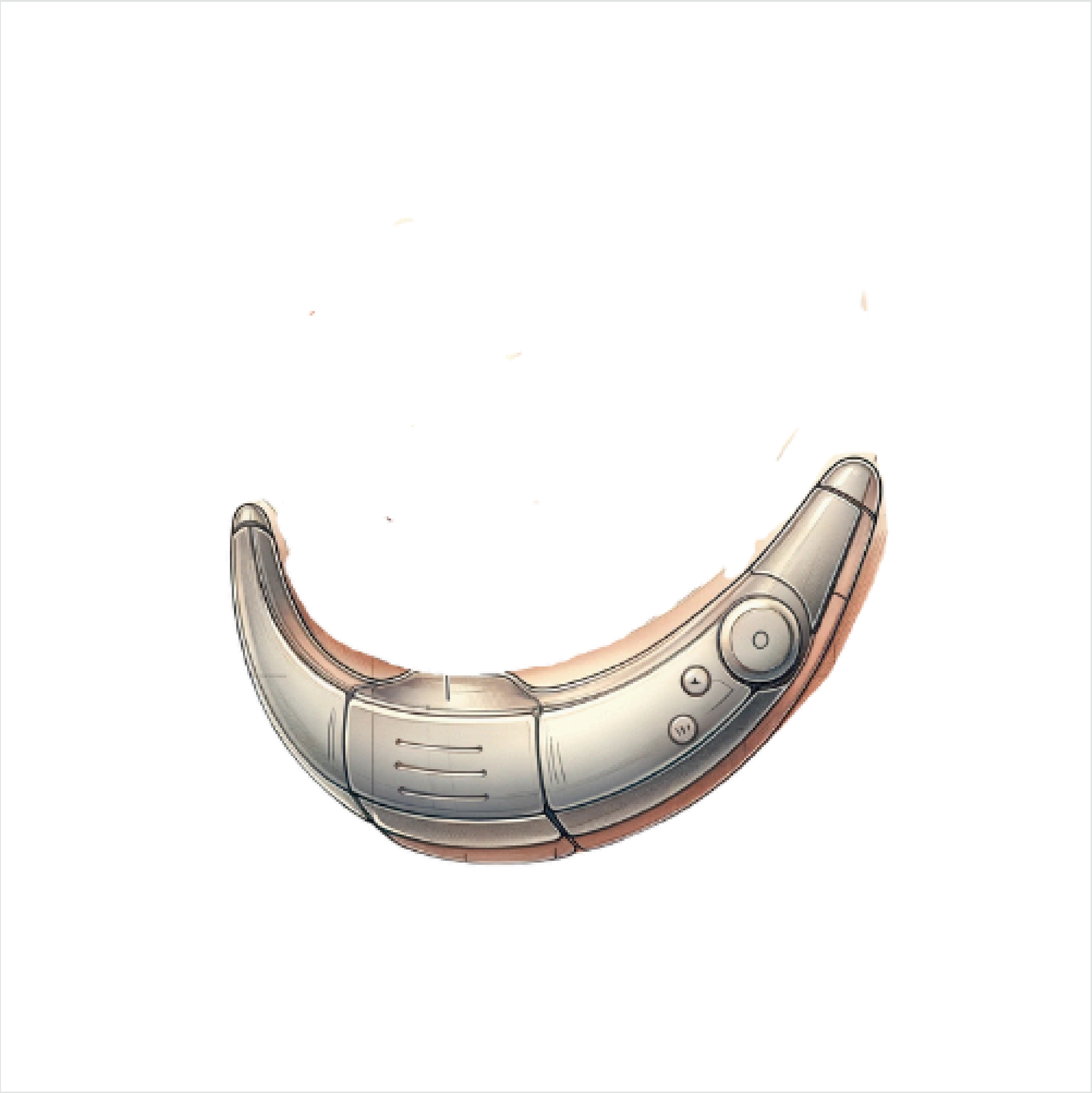
- **Protection Through Concealment:** Inspired by dental inner braces, the device is positioned in a less exposed area—inside the mouth.
- **Tongue-Based Operation:** Operates primarily on tongue movements, leveraging the mouth's natural protection for the device.
- **Capacitive Sensors:** Employs capacitive sensors to detect and interpret the user's tongue movements for controlling the HMI.

Advantages:

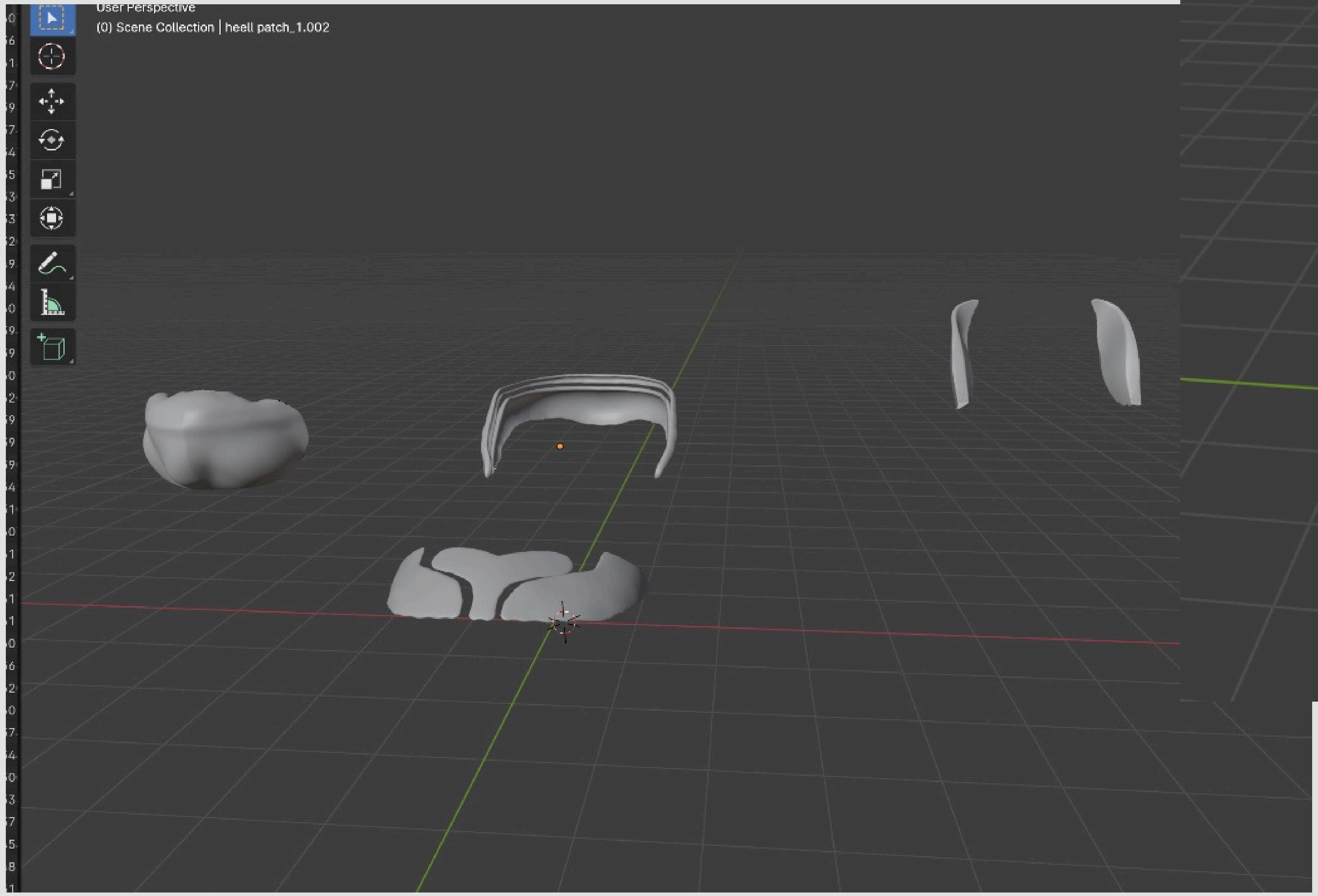
1. Reduced exposure of sensors to environmental hazards.
2. Simplified maintenance due to the centralized location of sensors.
3. The innovative design allows for natural, intuitive control through body movements.
4. Enhanced durability and safety with reduced operational costs.

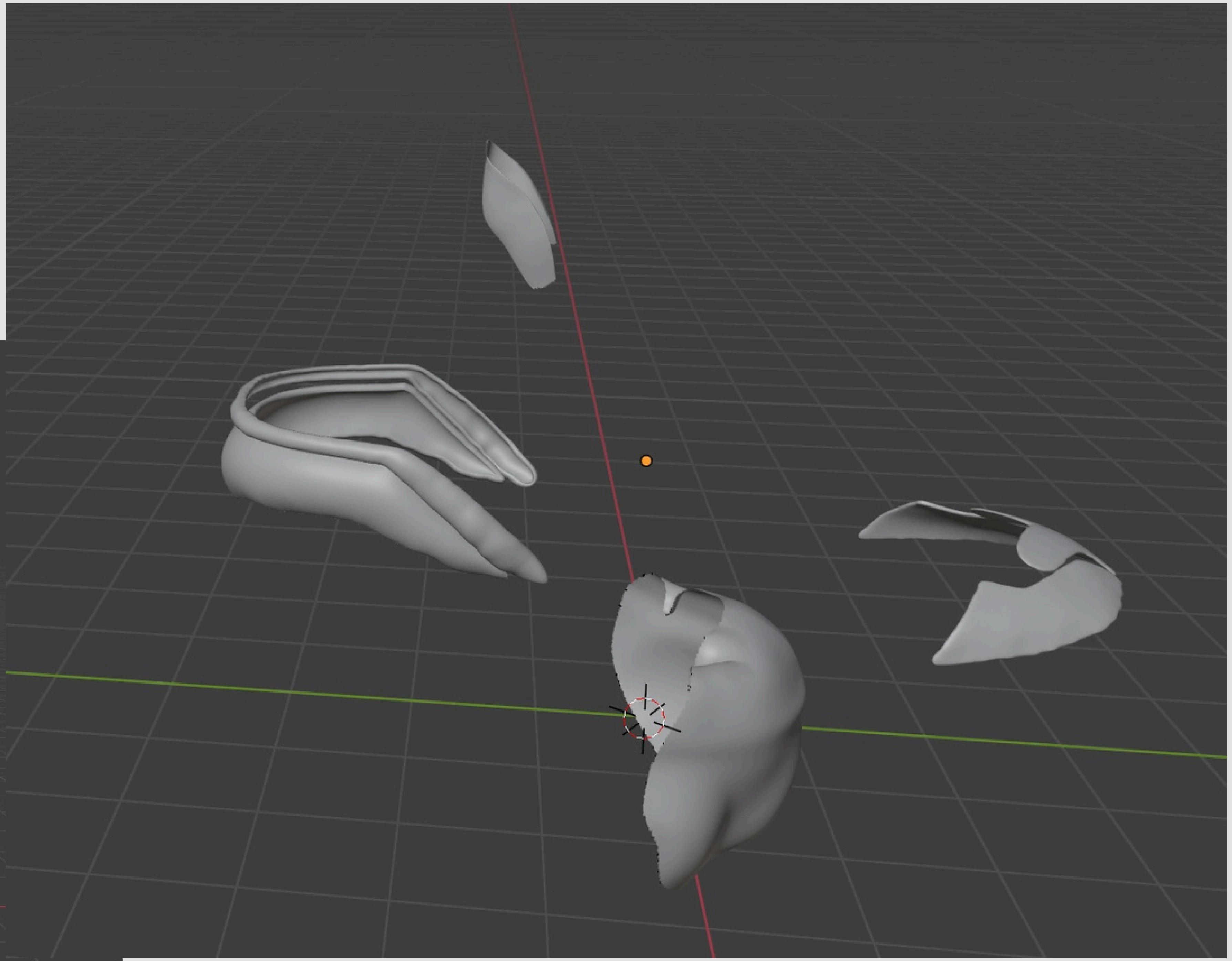
Rough Sketch





Model





WEEK 5–6 (MAR 8, 14, 16): PRODUCT FINALIZATION, INITIAL BOM

Main Goals:

- Finalizing on the solution, CAD design and circuits
- Deciding on exact components of product
- Generate BoM and identify suitable vendors
- Procuring Items
- Starting the initial phase of building the prototype

Subtasks

Solution Finalization
Finalize the design solution for the haptic device, incorporating feedback and research findings.

Rough Sketch with Component Placement
Draw a rough sketch detailing the exact placement of components within the device, aiding in visualization and assembly guidance.

CAD Design and Circuit Development

- Design detailed 3D CAD models for all physical components of the product.
- Develop and finalize the electronic circuit diagrams for the haptic system.

Circuit Simulation
Perform simulations of the electronic circuit to verify functionality and identify any potential issues before physical assembly.

Component Selection
Decide on the exact components needed for the haptic device, including electronic parts (sensors, actuators, microcontrollers) and mechanical parts (glove materials, structural elements).

Bill of Materials (BoM) Generation

- Generate a comprehensive BoM based on the finalized CAD designs and selected components.
- Identify suitable vendors and suppliers for each item listed in the BoM.

Prototype Assembly Initiation
Begin the initial phase of assembling the prototype, starting with the integration of electronic components followed by mechanical assembly.

Mid-Review Check

Conduct a mid-review check to assess the project's progress against the timeline and make necessary adjustments to stay on schedule.

Final Product Concept

Unlike the proposed inner braces concept, which presented feasibility issues, the haptic glove concept focuses on practical application and ease of integration into existing mining operations.

Key Advantages of the Haptic Glove:

1. **Minimalist Design:** By minimizing the number of exposed sensors, the glove significantly reduces the risk of damage and the associated maintenance costs, offering a durable solution for harsh mining environments.
2. **Centralized Sensing in Safe Zones:** Concentrating all sensors in the back palm or wrist area—a region less exposed to movement and potential harm—enhances the device's resilience and reliability. This strategic placement ensures optimal protection and longevity of critical components.
3. **Tension-Based Sensing with Machine Learning:** The glove features tension-based sensors for each finger, linked to a central processing unit. This design, combined with advanced machine learning algorithms, allows for precise interpretation of hand movements and tension patterns, enabling accurate control over the HMI.
4. **Simplified HMI Control:** By converting sensory inputs into streamlined HMI outputs, the glove facilitates intuitive interaction with machinery, improving safety and efficiency in mining operations without the complexity and vulnerability of traditional devices.

Opting for the haptic glove over the braces approach was driven by a commitment to practicality, ease of use, and enhanced safety. By focusing on a solution that integrates seamlessly into miners' daily operations while offering superior protection and intuitive control, we are setting a new standard for haptic devices in the mining industry. This innovation not only addresses current safety challenges but also paves the way for future advancements in mining technology

Product Uniqueness

The uniqueness of the unconventional haptic glove, as developed from our literature review and subsequent discussions, centers on several key innovations that distinguish it from existing solutions in the field:

Integrated Tension-Based Sensing: Unlike traditional haptic devices that rely on a wide array of complex sensors, our glove utilizes tension-based sensors connected by strings to each finger, offering a unique approach to capturing hand movements and gestures with high precision.

Centralized Sensor Placement: By strategically locating all sensors in the back palm or wrist area—parts of the hand less prone to damage—we significantly enhance the durability and reliability of the device, setting it apart from others that scatter sensors across more vulnerable locations.

Machine Learning for Enhanced Interpretation: Our glove leverages machine learning algorithms, a feature not commonly integrated into haptic devices. This allows for sophisticated interpretation of tension patterns and hand placements, enabling nuanced control over machinery through the Human-Machine Interface (HMI).

Minimalist and Durable Design: The glove's design philosophy prioritizes minimalism and durability, catering specifically to the harsh conditions of mining environments. This focus on reducing exposed components to minimize maintenance and increase longevity is a distinctive advantage.

Focused on Mining Safety and Efficiency: Drawing from the literature review, our solution is tailored to address the specific challenges and risks associated with mining operations. It's designed not just as a haptic device but as a comprehensive safety solution, aiming to reduce accidents through better control and interaction with heavy machinery.

Adaptable and Scalable: The haptic glove is conceived with adaptability in mind, capable of integration into existing systems and scalable to accommodate future technological advancements or changes in mining operations.

Our haptic glove emerges as a pioneering solution within the realm of mining safety equipment, distinguished by its intelligent sensor integration, focus on durability, and advanced data interpretation capabilities. These attributes underscore our commitment to innovation, safety, and efficiency in challenging industrial environments.

Key challenges (both prototyping and in manufacturing) and How we aim to solve them?

1. Complexity in Design:
- Simplify design by minimizing the number of components.
- Use modular design for easy replacement and upgrades.

2. Sensor Durability:
- Select robust materials suitable for harsh mining conditions.
- Implement redundancy for critical sensors.

3. Integration with Existing Systems:
- Develop using standard communication protocols for compatibility.
- Engage with equipment manufacturers early for integration support.

4. User Acceptance:
- Involve end-users in the design process for user-friendly interfaces.
- Provide comprehensive training and support.

5. Cost Constraints:
- Focus on cost-effective materials and components.
- Optimize design to reduce manufacturing expenses.

6. Supply Chain Delays:
- Diversify supplier base.
- Maintain inventory of critical components.

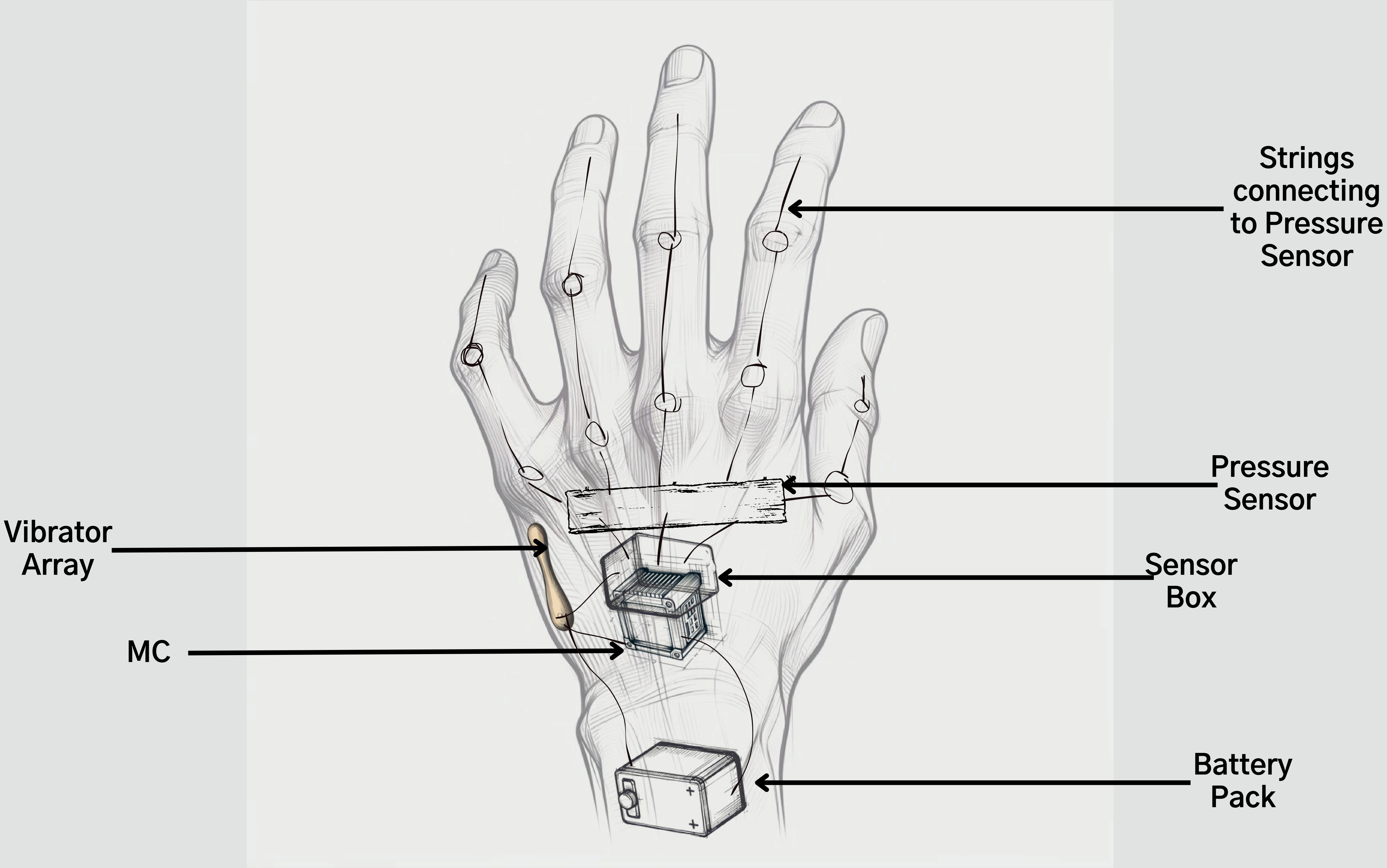
7. Technological Limitations:
- Stay updated on emerging technologies to find advanced solutions.
- Partner with tech companies for innovative components.

8. Regulatory Compliance:
- Consult with regulatory bodies throughout development.
- Ensure thorough testing to meet safety standards.

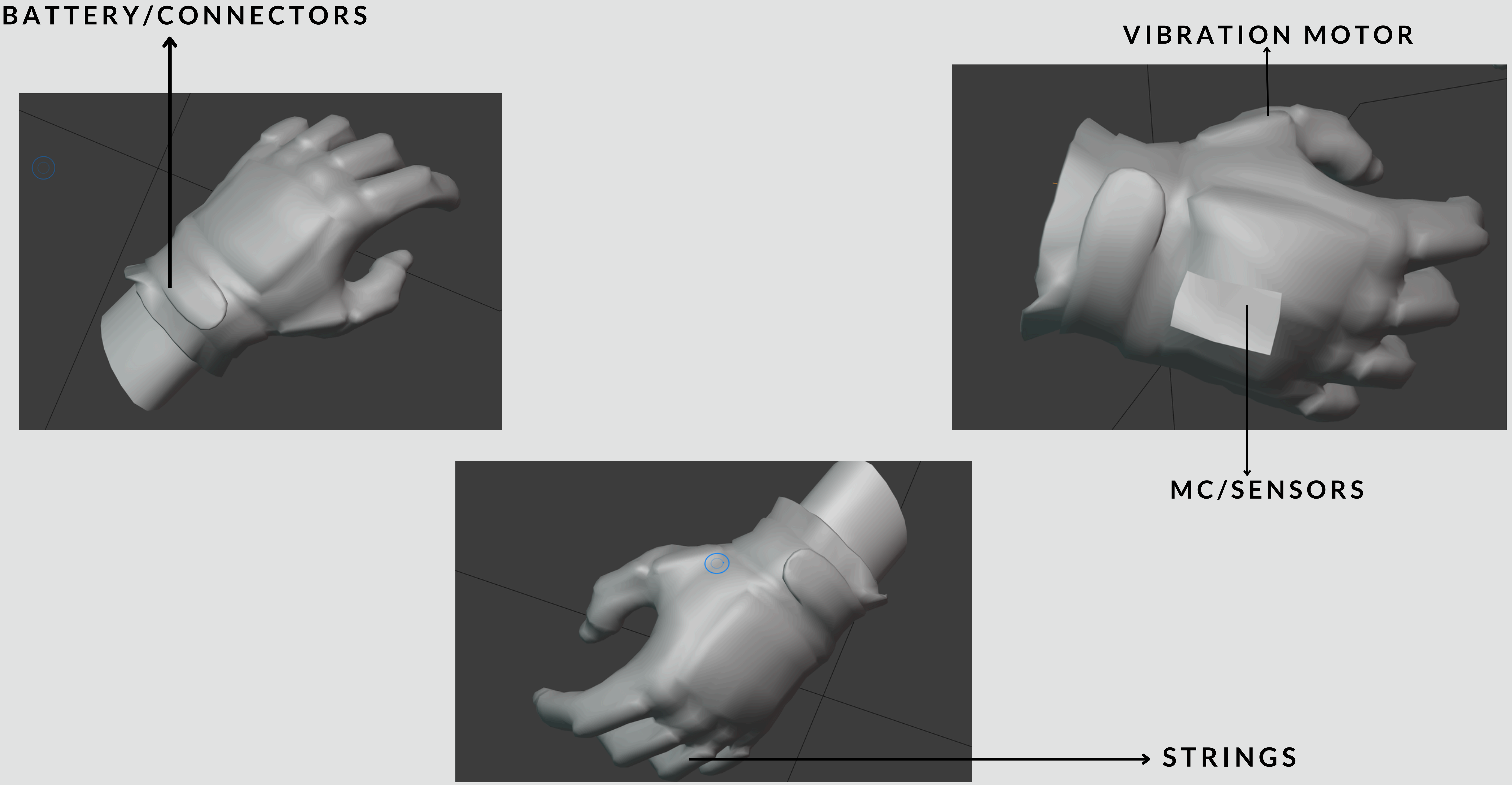
9. Data Privacy and Security:
- Incorporate encryption and secure data protocols.
- Regularly update security measures to protect user data.

10. Maintaining Project Timelines:
- Implement agile project management.
- Allow for buffer periods to accommodate unforeseen delays.

Rough Sketch with exact placement of components

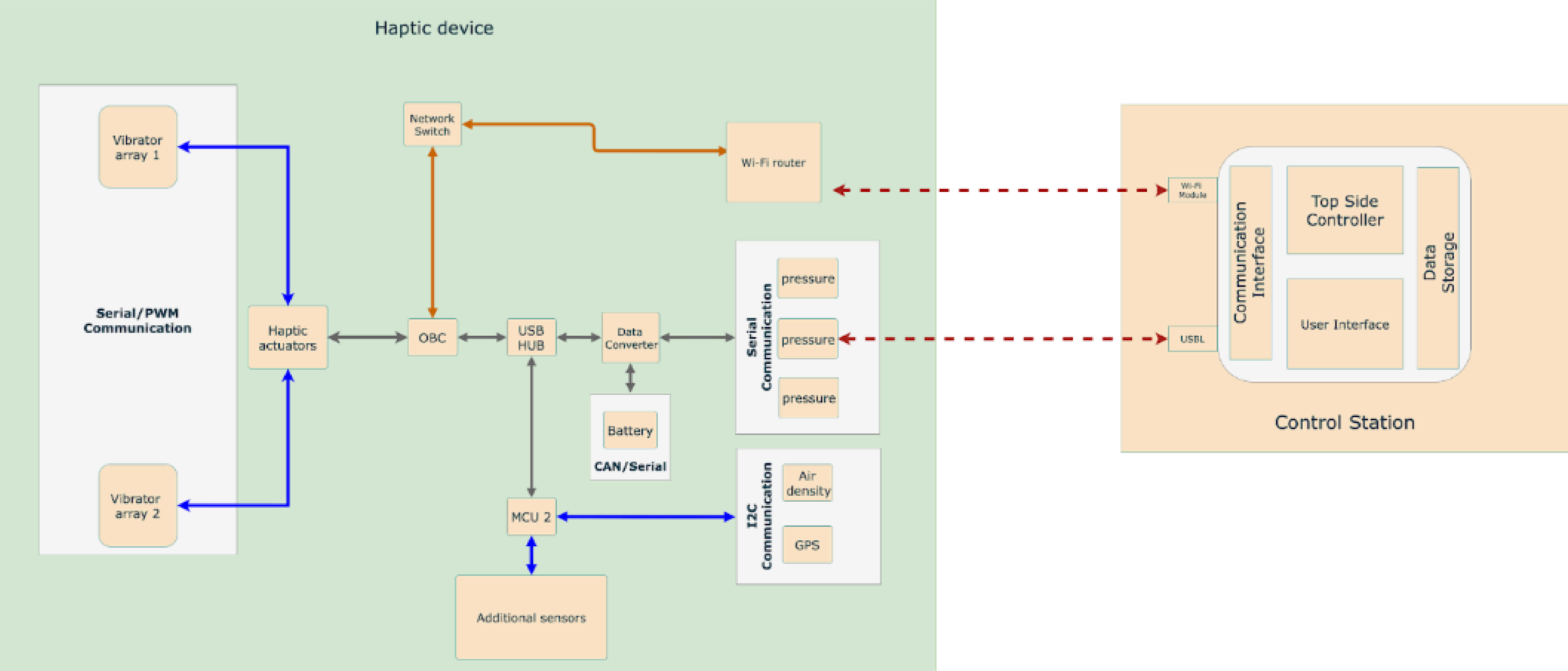


Final CAD Model



Final Circuit Diagram

Due to limitations in KiCad's resources for simulation, we will pivot to developing a predictive framework, directly mapping sensor inputs to code for real-time analysis and functionality testing.



Framework

Hand gestures can be classified as static or dynamic.

- Static hand gesture recognition considers the shape characteristics of the hand at a point in time, and dynamic hand gestures focus on a series of hand movements over a time interval.
- Dynamic hand gestures can involve both hand gesture classification and finger angle estimation. Hand gesture classification can decode human intentions as discrete commands for interacting with and controlling physical hardware. Finger angle estimation can be utilized to achieve the continuous operation of physical devices. For example, varying finger angles could be mapped to operate dynamic realistic tasks, such as game joysticks in virtual reality.

Overview

The architecture is designed to predict finger joint states based on tension sensor data. It encompasses various methods and considerations to ensure accurate prediction of finger movements, enabling seamless interaction with physical devices.

Key Points that are considered Consider:

Sensor Placement and Configuration:

- Optimize sensor placement on the hand to capture finger joint movements effectively.
- Configure sensors to provide reliable tension readings for different finger joint states.

Feature Extraction:

- Extract relevant features from tension sensor data to represent finger joint states accurately.
- Include features such as tension levels, rate of change of tension, and temporal patterns in tension data.

Framework

Model Training Methods:

- Training based on visual input:
- Utilize visual data alongside tension sensor readings to train the prediction model, providing additional context for accurate joint state prediction.

Training based on fixed states:

Use labeled datasets mapping tension sensor readings to specific finger joint states for discrete state prediction tasks.

Dynamic State Prediction:

- Employ dynamic hand gesture recognition techniques to predict finger joint states over time intervals.
- Analyze sequences of tension sensor readings to predict continuous finger joint movements accurately.

Boundary Conditions and Calibration:

- Define boundary conditions for finger joint movements to ensure realistic predictions.
- Perform calibration to account for variations in sensor readings due to factors such as hand size and sensor sensitivity.

Mapping to Real-World Applications:

- Develop mapping functions to translate predicted finger joint states into actions for controlling physical devices.
- Design intuitive mappings, such as mapping finger angles to control game joysticks in virtual reality environments.

Initial BoM and List of Sources

Item	Cost	Count
Teensy 4.0	1977	1
5v operated vibration motors	65	2
Force sensor	600	4
Tension retaining string	189	1
Glove with pocket	–	1

Click on the item to redirect to the vendor website link

WEEK 7(MAR 22): INTEGRATION, FINAL SENSOR SELECTION & BOM, INITIAL PROTOTYPING PHASE

Main Goals:

- Component Integration and Design
- Connectivity and Final Sensor Selection
- Development of a Training Aid

Subtasks

Component Integration and Design

- **Evaluate Potentiometer Torque:** Research the required torque for comfortable potentiometer use.
- **Select and Attach Torsion Springs:** Find low-torque torsion springs and design their attachment mechanism.
- **Design String-Ring Mechanism**: Develop a method for attaching strings to finger rings, ensuring movement accuracy without hindrance.
- **Glove Integration Plan:** Outline integration of all components into the glove, focusing on comfort and durability.

Connectivity and Sensor Selection

- **Wireless Communication Technology Decision:** Choose between Bluetooth and WiFi based on performance criteria.
- **Finalize Sensor Selection:** Pick sensors based on accuracy, efficiency, and robustness, planning their integration into the glove.

Final Bill of Materials (BoM) Generation

Generate a comprehensive BoM based on the finalized CAD designs and selected components with updated sensors and components

Development of a Training Aid

- **OpenCV Training Aid Development:** Build a software tool using OpenCV for calibrating the glove, focusing on gesture analysis and sensitivity adjustment.

Procurement

Initiate the procurement process for all necessary components and materials, coordinating with identified vendors and suppliers.

Main Simulation

Start procuring!

Final Sensors and Components

Proposal for changing sensor stack:

Our current sensor stack faces several critical issues that impact both performance and viability:

1. **EM Interference Susceptibility:** The sensors are prone to electromagnetic interference, compromising the reliability of data.
2. **Exceeding Budget Constraints:** The final cost of the sensor stack exceeds our budget cap of INR 3000, making it financially unfeasible.
3. **Complex Mechanical Assembly:** The assembly process for the current sensors is overly complicated, increasing production time and potential for errors.
4. **High Variance in Noise:** There's a significant variance in white noise, leading to data inconsistency and unreliable outputs.

Alternate choice:

To address these challenges, we propose switching to rotary potentiometers. This alternative offers several advantages:

- **EM Interference:** Rotary potentiometers are less susceptible to electromagnetic interference, ensuring more stable and reliable readings.
- **Cost-Effective:** The cost of integrating rotary potentiometers into our design significantly falls within our budget of INR 3000, offering a more economically viable solution.
- **Simplified Assembly:** The mechanical assembly of rotary potentiometers is straightforward, reducing production complexity and the likelihood of assembly errors.
- **Reduced Noise Variance:** Potentiometers inherently exhibit lower variance in signal noise, providing cleaner and more consistent data.

Integration

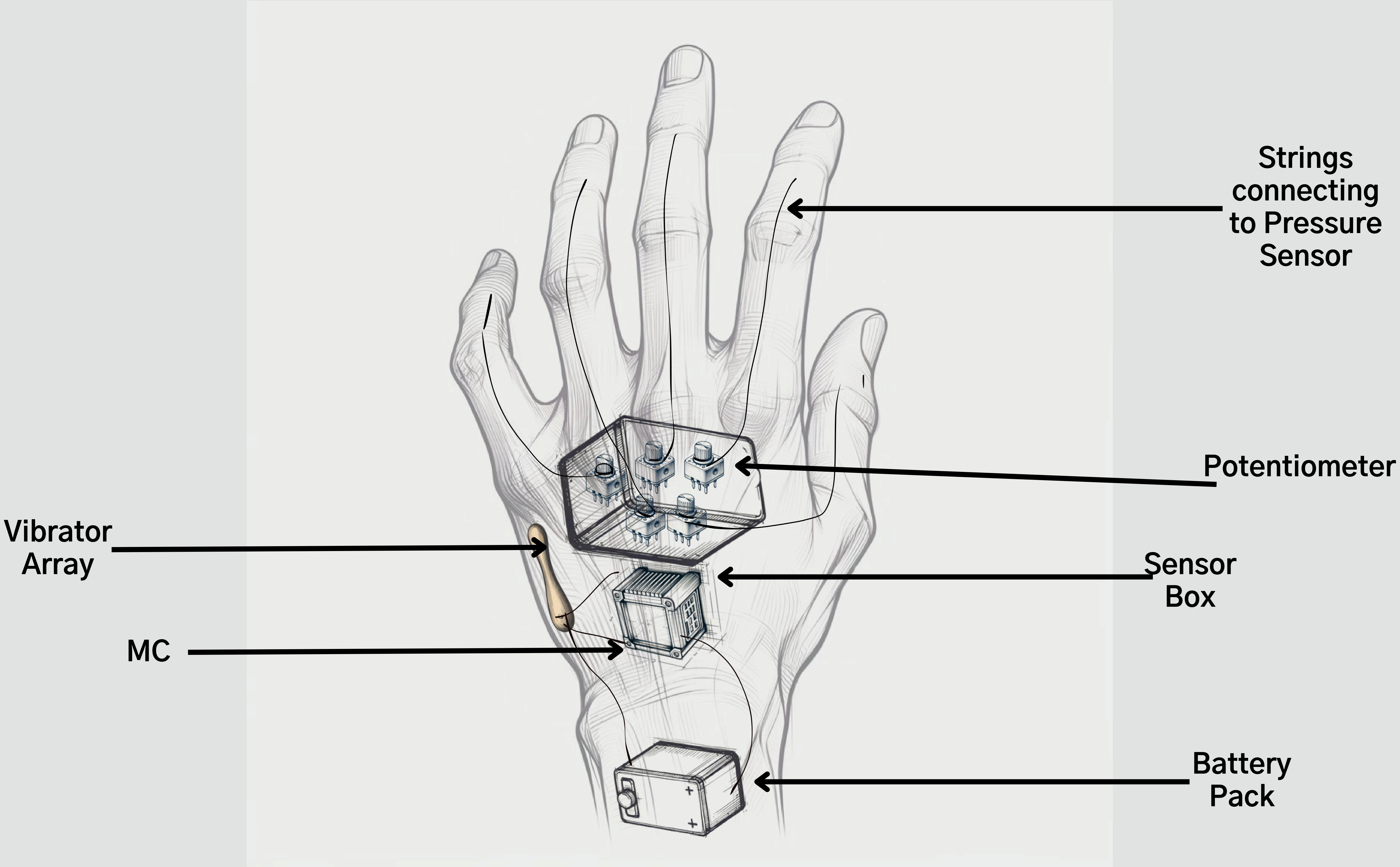
How many potentiometers and where?
3-5 rotary potentiometers in the glove are achievable with design focusing on the back of the palm/wrist area.

How much torque does it require to turn?
Typically, potentiometers used in such applications require very low torque, often less than 10 mNm (milliNewton-meters).

How are we attaching the torsion spring?

- **Spring-Potentiometer Integration:** Attach one end of a torsion spring to the potentiometer's shaft for direct movement capture.
- **String Mechanism to Finger Rings:** Connect finger rings to torsion springs with strings, translating finger movements into potentiometer rotation.
- **Spring Housing:** Design compact housing on the glove to securely hold torsion springs and potentiometers in place, aligned with each finger's movement.
- **Adjustment Mechanism:** Incorporate an adjustment feature for string tension and spring torque setting, allowing calibration for accuracy across users.
- **Comfort and Mobility:** Select springs that offer sufficient force for neutral position return without hindering finger mobility, ensuring the glove remains flexible and comfortable.

Integration of components and mechanism



OpenCV Training Aid Development

```python
import cv2
import time
import numpy as np
import os

protoFile = os.path.relpath("/Users/camelot/projects/PAT_mining/training_aid/MV_HandKeyPointDetector/hand/pose_deploy.prototxt")

# protoFile = "hand/pose_deploy.prototxt.txt"
weightsFile = "hand/pose_iter_102000.caffemodel"
nPoints = 22
POSE_PAIRS = [ [0,1],[1,2],[2,3],[3,4],[0,5],[5,6],[6,7],[7,8],[0,9],[9,10],[10,11],[11,12],[0,13],[13,14],[14,15],[15,16],[0,17],[17,18],[18,19],[1

threshold = 0.2


input_source = 0
cap = cv2.VideoCapture(input_source)
hasFrame, frame = cap.read()

frameWidth = frame.shape[1]
frameHeight = frame.shape[0]

aspect_ratio = frameWidth/frameHeight

inHeight = 368
inWidth = int(((aspect_ratio*inHeight)*8)//8)

vid_writer = cv2.VideoWriter('output.avi',cv2.VideoWriter_fourcc('M','J','P','G'), 15, (frame.shape[1],frame.shape[0]))

net = cv2.dnn.readNetFromCaffe(protoFile, weightsFile)
k = 0
while 1:
    k+=1
    t = time.time()
    hasFrame, frame = cap.read()
    frameCopy = np.copy(frame)
```

```
    inpBlob = cv2.dnn.blobFromImage(frame, 1.0 / 255, (inWidth, inHeight),
                              (0, 0, 0), swapRB=False, crop=False)

    net.setInput(inpBlob)

    output = net.forward()

    print("forward = {}".format(time.time() - t))

    # Empty list to store the detected keypoints
    points = []

    for i in range(nPoints):
        # confidence map of corresponding body's part.
        probMap = output[0, i, :, :]
        probMap = cv2.resize(probMap, (frameWidth, frameHeight))

        # Find global maxima of the probMap.
        minVal, prob, minLoc, point = cv2.minMaxLoc(probMap)

        if prob > threshold :
            cv2.circle(frameCopy, (int(point[0]), int(point[1])), 6, (0, 255, 255), thickness=-1, lineType=cv2.FILLED)
            cv2.putText(frameCopy, "{}".format(i), (int(point[0]), int(point[1])), cv2.FONT_HERSHEY_SIMPLEX, .8, (0, 0, 255), 2, lineType=cv2.LINE_A

            # Add the point to the list if the probability is greater than the threshold
            points.append((int(point[0]), int(point[1])))
        else :
            points.append(None)

    # Draw Skeleton
    for pair in POSE_PAIRS:
        partA = pair[0]
        partB = pair[1]

        if points[partA] and points[partB]:
            cv2.line(frame, points[partA], points[partB], (0, 255, 255), 2, lineType=cv2.LINE_AA)
```

```python
        if prob > threshold :
            cv2.circle(frameCopy, (int(point[0]), int(point[1])), 6, (0, 255, 255), thickness=-1, lineType=cv2.FILLED)
            cv2.putText(frameCopy, "{}".format(i), (int(point[0]), int(point[1])), cv2.FONT_HERSHEY_SIMPLEX, .8, (0, 0, 255), 2, lineType=cv2.LINE_A

            # Add the point to the list if the probability is greater than the threshold
            points.append((int(point[0]), int(point[1])))
        else :
            points.append(None)

    # Draw Skeleton
    for pair in POSE_PAIRS:
        partA = pair[0]
        partB = pair[1]

        if points[partA] and points[partB]:
            cv2.line(frame, points[partA], points[partB], (0, 255, 255), 2, lineType=cv2.LINE_AA)
            cv2.circle(frame, points[partA], 5, (0, 0, 255), thickness=-1, lineType=cv2.FILLED)
            cv2.circle(frame, points[partB], 5, (0, 0, 255), thickness=-1, lineType=cv2.FILLED)

    print("Time Taken for frame = {}".format(time.time() - t))

    # cv2.putText(frame, "time taken = {:.2f} sec".format(time.time() - t), (50, 50), cv2.FONT_HERSHEY_COMPLEX, .8, (255, 50, 0), 2, lineType=cv2.LI
    # cv2.putText(frame, "Hand Pose using OpenCV", (50, 50), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 50, 0), 2, lineType=cv2.LINE_AA)
    cv2.imshow('Output-Skeleton', frame)
    # cv2.imwrite("video_output/{:03d}.jpg".format(k), frame)
    key = cv2.waitKey(1)
    if key == 27:
        break

    print("total = {}".format(time.time() - t))

    vid_writer.write(frame)

vid_writer.release()
```

projects   Sign in

- projects
  - PAT_mining
    - training_aid
      - MV_HandKeyPointDetector
        - data
        - hand
          - pose_deploy.prototx
          - pose_iter_102000.ca
        - .gitignore
        - __init__.py
        - CMakeLists.txt
        - getModels.sh
        - HandKeypointDetector.p
        - handPoseImage.py
        - handPoseVideo.py
        - output.avi
      - output.avi
      - ver1.py

ver1.py | HandKeypointDetector.py | handPoseImage.py | handPoseVideo.py | getModels.sh

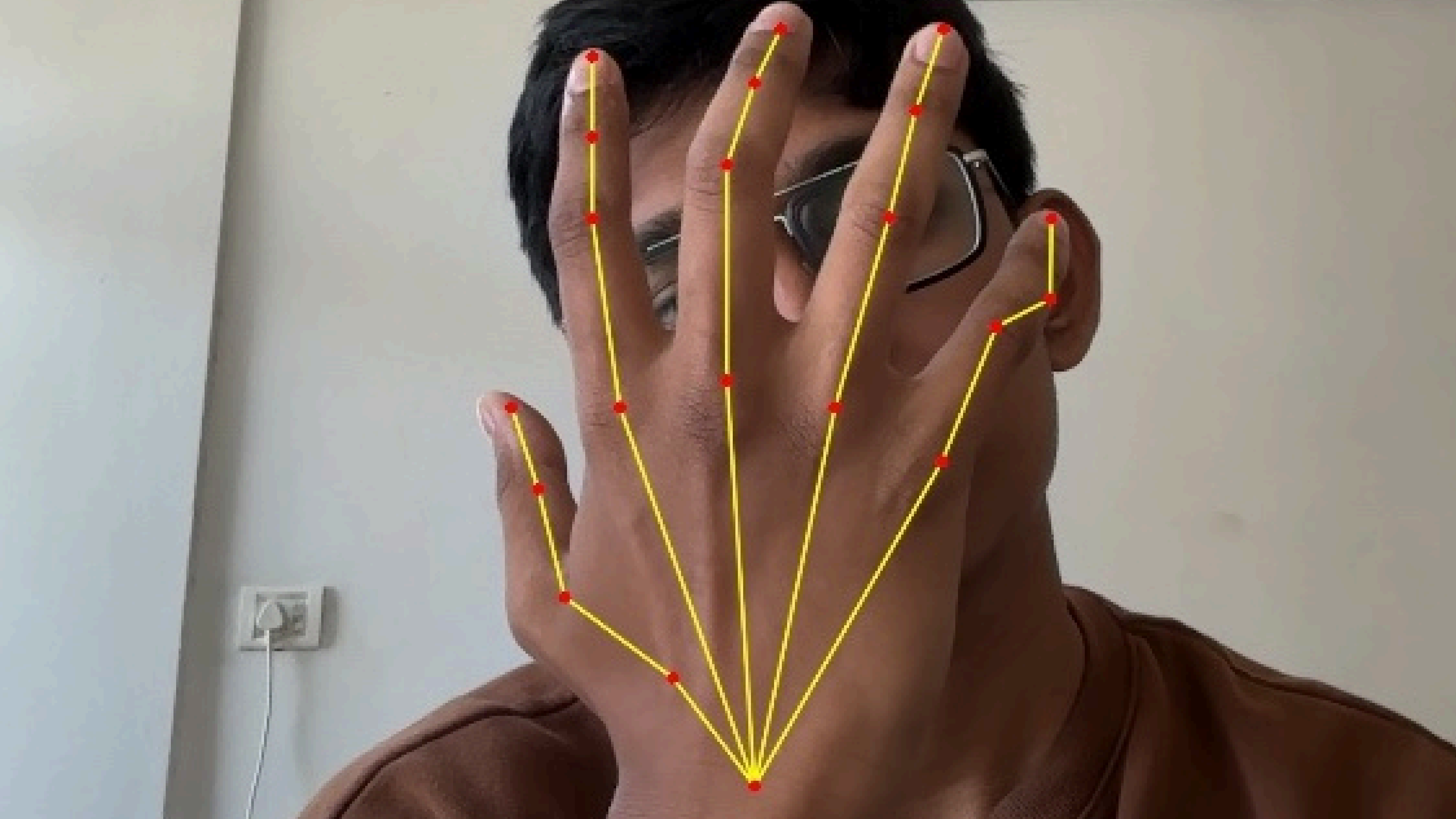
PAT_mining/training_aid/MV_HandKeyPointDetector/HandKeypointDetector.py

```python
                # confidence map of corresponding body's part.
                probMap = output[0, i, :, :]
                probMap = cv2.resize(probMap, (frameWidth, frameHeight))

                # Find global maxima of the probMap.
                minVal, prob, minLoc, point = cv2.minMaxLoc(probMap)

                if prob > threshold :
                    cv2.circle(frame, (int(point[0]), int(point[1])), 2, (0, 0, int(255*prob)), thickness=-1, lineType=cv2.FILLED)
                    cv2.putText(frame, "{}".format(self.rearrange_finger_indices[i]), (int(point[0]), int(point[1])), cv2.FONT_HERSHEY_SIMPLEX, 0.1,

                    # Add the point to the list if the probability is greater than the threshold
                    points.append(np.array([int(point[0]), int(point[1]),prob]))
                    # points_probs.append(prob)
                else :
                    # points_probs.append(0)
```

projects — zsh | MV_HandKeyPointDetector — zsh

```
forward = 0.4654858112335205
Time Taken for frame = 0.47999095916748047
total = 0.49742698669433594
forward = 0.486875057220459
Time Taken for frame = 0.49921393394470215
total = 0.5170688629150391
forward = 0.49483418464660645
Time Taken for frame = 0.5088560581207275
total = 0.5276370048522949
forward = 0.4907708168029785
Time Taken for frame = 0.5044159889221191
total = 0.5232861042022705
forward = 0.472156286239624
Time Taken for frame = 0.4867682456970215
total = 0.5044920444488525
^CTraceback (most recent call last):
  File "/Users/camelot/projects/PAT_mining/training_aid/MV_HandKeyPointDetector/handPoseVideo.py", line 46, in <module>
    output = net.forward()
KeyboardInterrupt

camelot@Sibis-MacBook-Pro MV_HandKeyPointDetector %
```

10 2 Click to restart and update Zed   1:1 Python

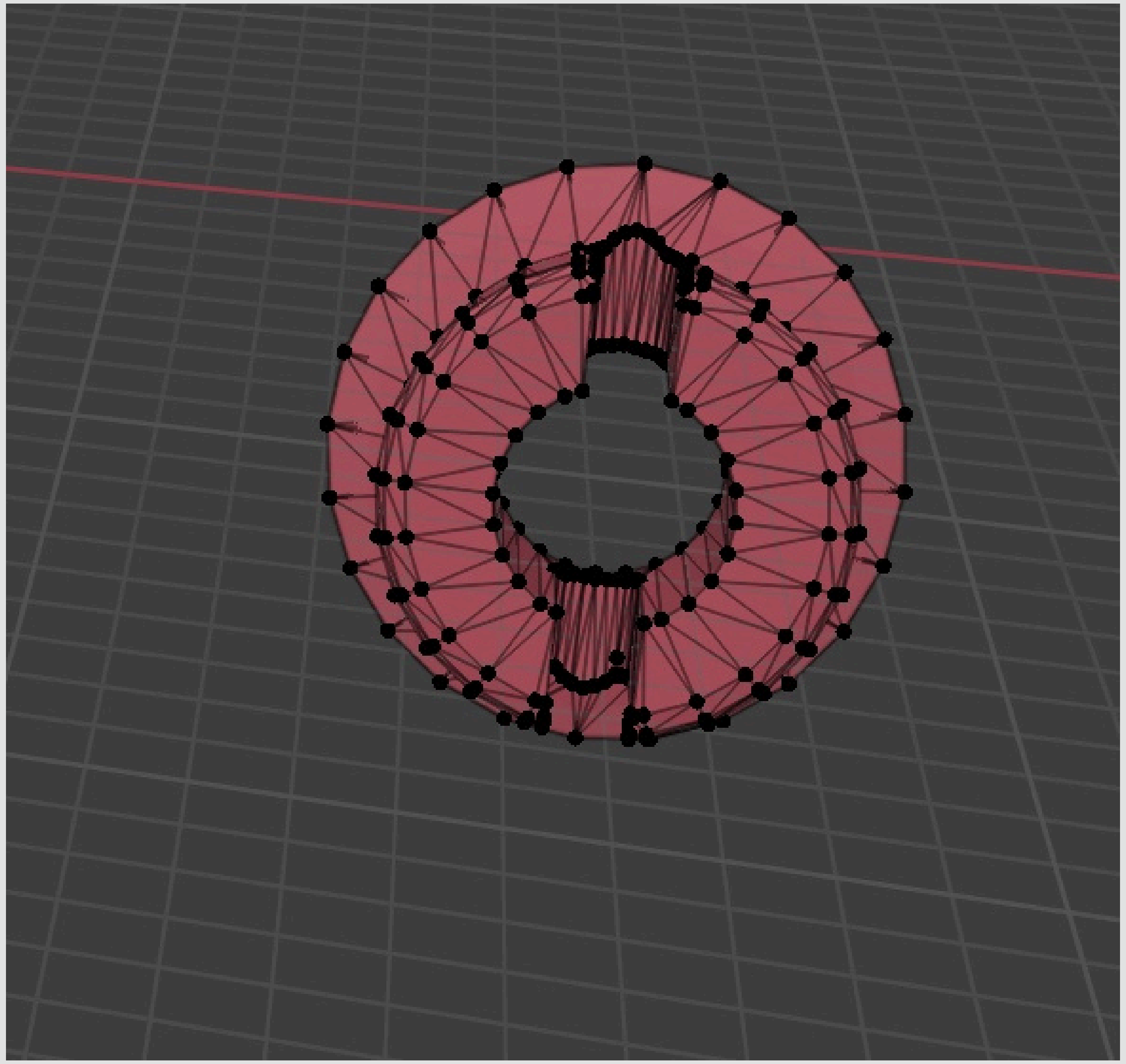


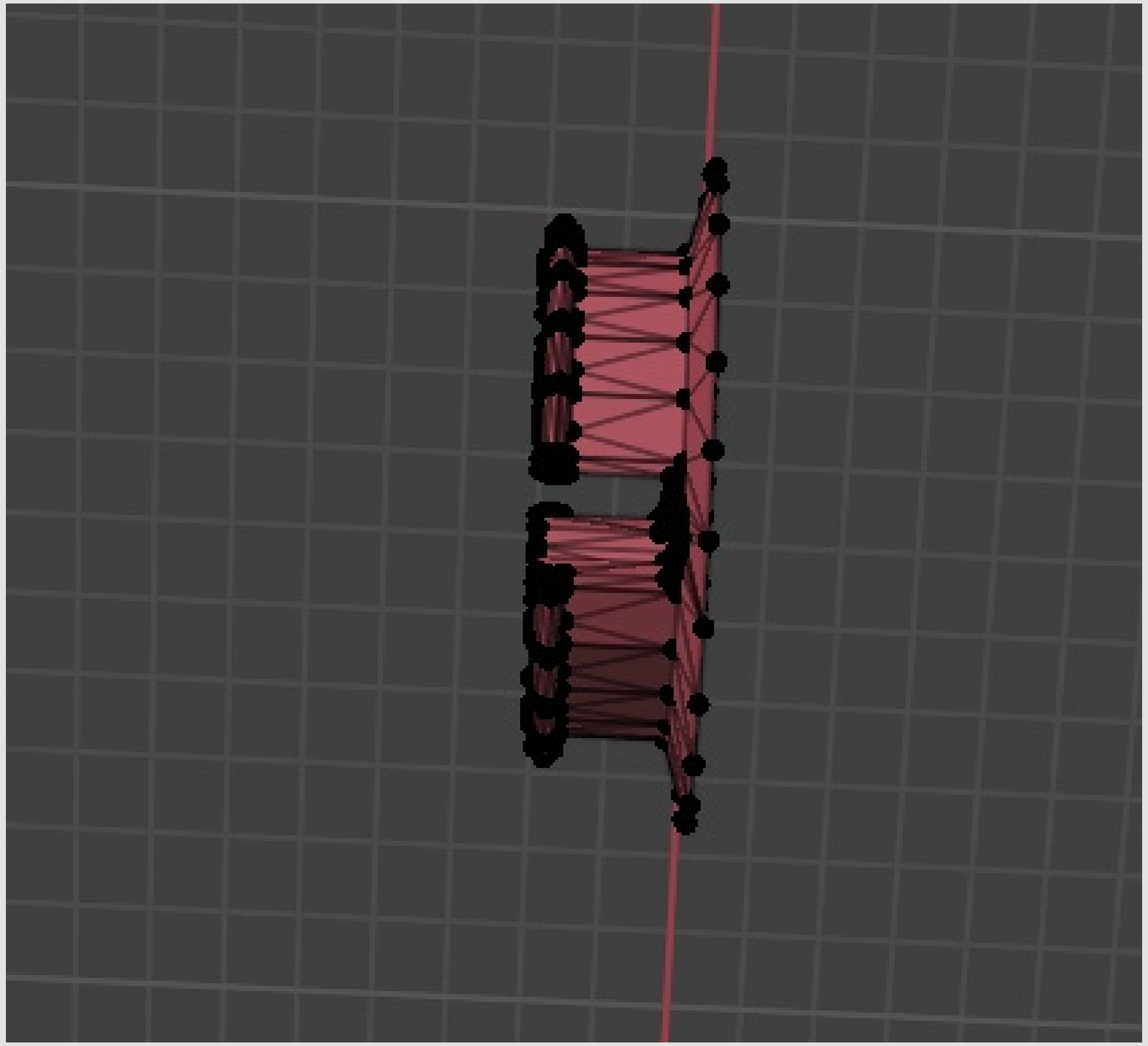
Final BoM and List of Sources

Item	Total Cost	Count
Nylon Wire	269	1
Selector Spring Loaded Potentiometer– QJ16	250	5
Glove	500	1
PLA	759	1
ESP 32	569	1

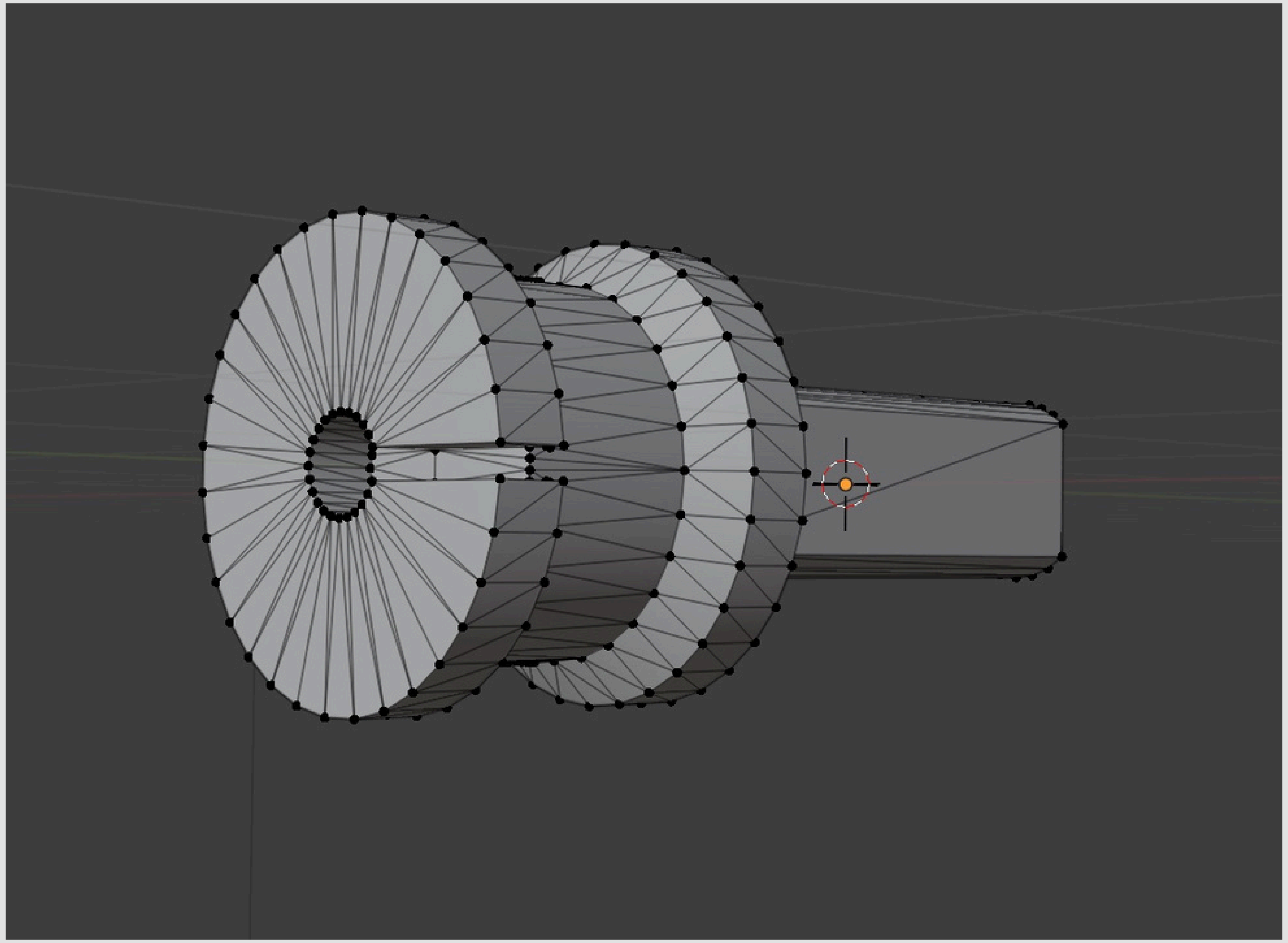
Click on the item to redirect to the vendor website link

Iterations of pulley

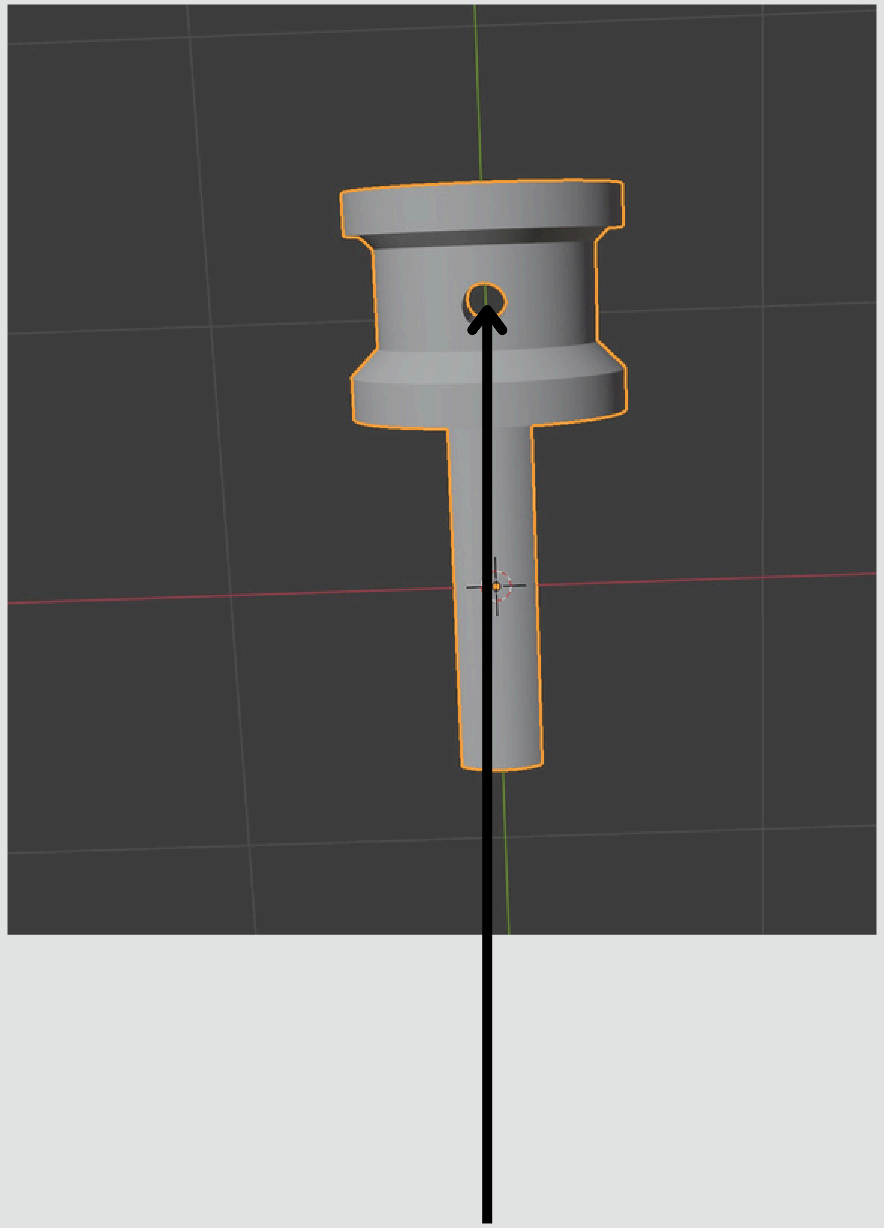




Potentiometer to strong link

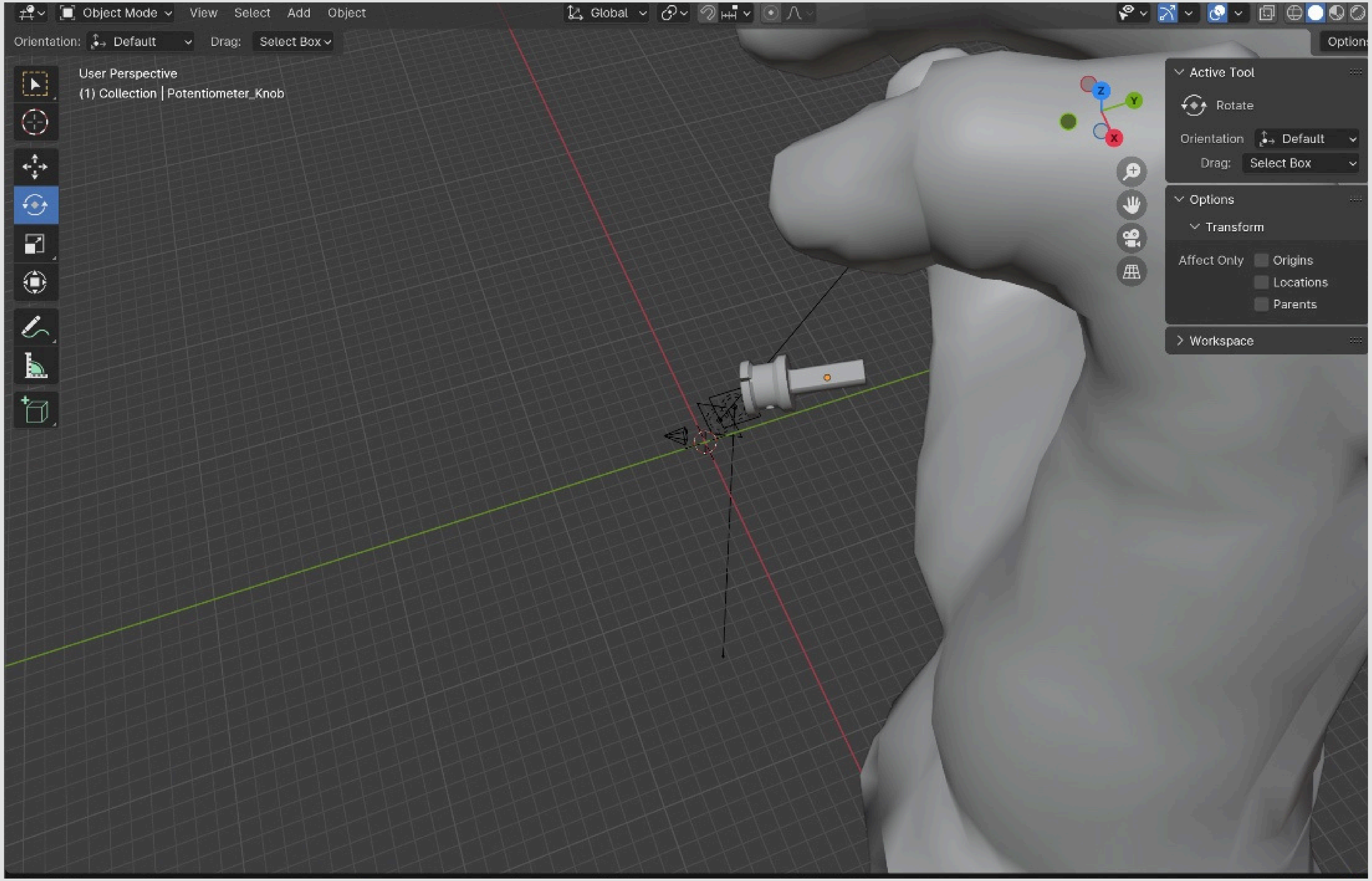


The slit is given as a option to customize in case of not using a spring loaded potentiometer(and aesthetics)



This is where the nylon string goes

Size comparison with hand

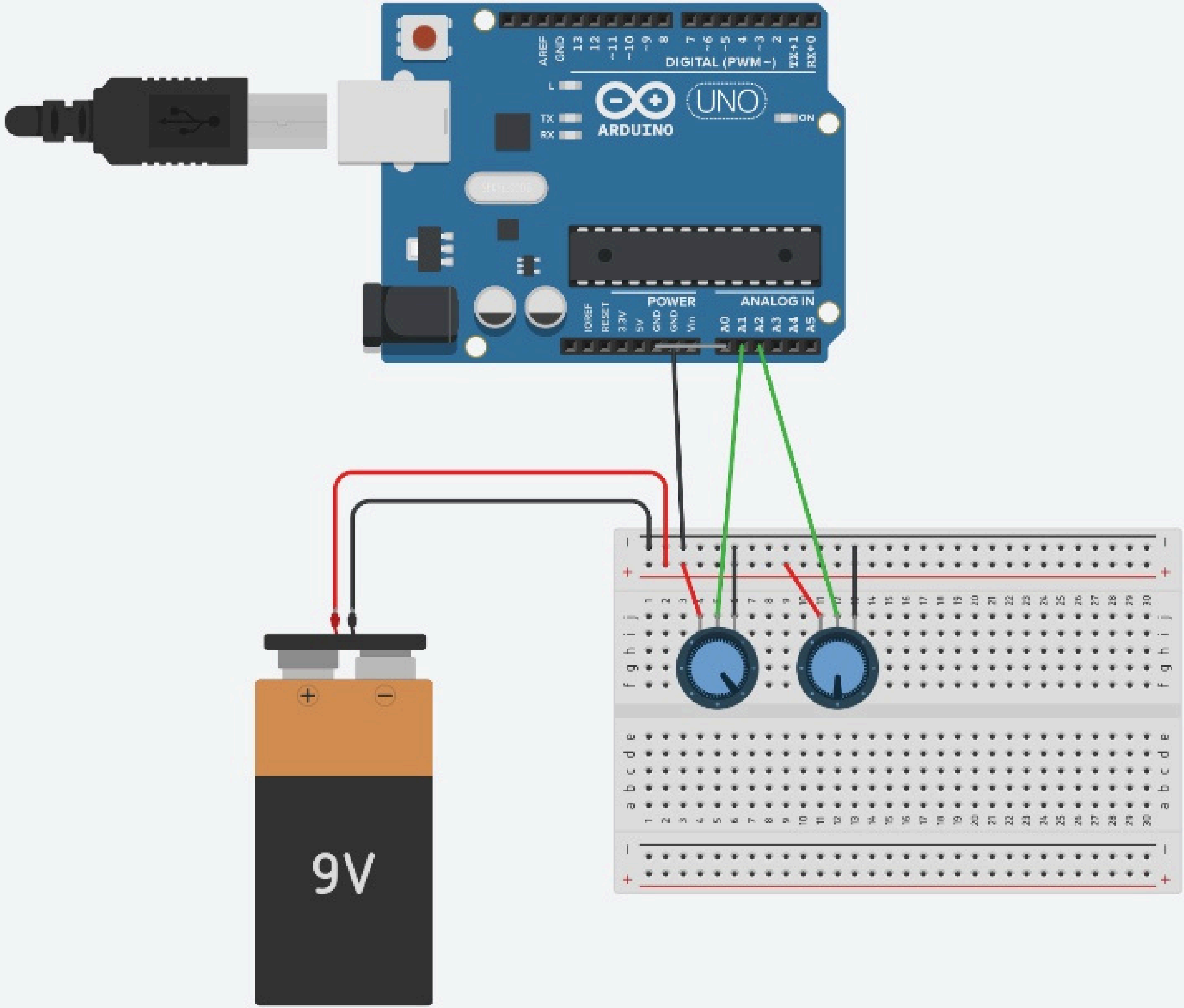


WEEK 8 (MAR 29): SIMULATION

Main Goals:

Working Simulation of the product

Simulation




```
//I found a cleaner solution for reading multiple analogs by
//implementing a "Reset" mechanism. This involves discharging the
//ADC capacitor by taking an analog read of a third pin directly
//connected to ground. This method ensures clean output when
//reading multiple analog inputs with only one ADC.


int sensorValue;
 int sensorValue2;
 float tsens;
 float tset;
 int resetADC;

 void setup()
 {
     Serial.begin(9600);
     pinMode(A0,INPUT);
     pinMode(A1,INPUT);
     pinMode(A2,INPUT);
 }

 void loop()
 {
     resetADC = analogRead(A0);
     sensorValue = analogRead(A2);
     tsens =  map(sensorValue, 0, 1023, 0, 500);

     resetADC = analogRead(A0);
     sensorValue2 = analogRead(A1);
     tset =  map(sensorValue2, 0, 1023, 0, 70);

     Serial.println(tsens);
     Serial.println(tset);
  }
```

Serial Monitor

```
45.00
450.00
45.00
450.00
45.00
450.0
```

WEEK 9 (APR 2): MID SEM REVIEW

Main Goals:

- Generating the final BoM
- Drafting a report
- Working on the final presentation
- Checking status with the gantt chart

WEEK 10 (APR 5,12): PROCURING & PROTOTYPING

Main Goals:

- Contacting vendors & negotiating
- Changing component types based on talks with vendors
- More Coding

```
Gesture-recognition
  model
    keypoint_classifier
      keypoint.csv
      keypoint_classifier.
      keypoint_classifier.
      keypoint_classifier.
      keypoint_classifier_
    point_history_classifie
      point_history.csv
      point_history_classi
      point_history_classi
      point_history_classi
      point_history_classi
    __init__.py
  utils
    __init__.py
    cvfpscalc.py
  .gitignore
```

projects
PAT_mining
Gesture-recognition
model
keypoint_classifier
keypoint.csv
keypoint_classifier.
keypoint_classifier.
keypoint_classifier.
keypoint_classifier_
point_history_classifie
point_history.csv
point_history_classi
point_history_classi
point_history_classi
point_history_classi
__init__.py
utils
__init__.py
cvfpscalc.py
.gitignore
app.py
keypoint_classification.i
keypoint_classification_E
point_history_classificat
README.md
training_aid

point_history_classifier.py | point_history.csv | keypoint_classification_EN.ipynb | keypoint_classifier.py | point_history_classification.ipynb | getMod

PAT_mining/Gesture-recognition/model/keypoint_classifier/keypoint_classifier.py › class KeyPointClassifier › def __init__

```python
#!/usr/bin/env python3
import numpy as np
import tensorflow as tf


class KeyPointClassifier(object):
    def __init__(
        self,
        model_path='model/keypoint_classifier/keypoint_classifier.tflite',
        num_threads=1,
    ):
        self.interpreter = tf.lite.Interpreter(model_path=model_path,
                                               num_threads=num_threads)

        self.interpreter.allocate_tensors()
        self.input_details = self.interpreter.get_input_details()
        self.output_details = self.interpreter.get_output_details()

    def __call__(
        self,
        landmark_list,
    ):
        input_details_tensor_index = self.input_details[0]['index']
        self.interpreter.set_tensor(
            input_details_tensor_index,
            np.array([landmark_list], dtype=np.float32))
        self.interpreter.invoke()

        output_details_tensor_index = self.output_details[0]['index']

        result = self.interpreter.get_tensor(output_details_tensor_index)

        result_index = np.argmax(np.squeeze(result))

        return result_index
```

projects — zsh | PAT_mining — zsh

projects
PAT_mining
Gesture-recognition
model
keypoint_classifier
keypoint.csv
keypoint_classifier.
keypoint_classifier.
keypoint_classifier.
keypoint_classifier_
point_history_classifi
point_history.csv
point_history_classi
point_history_classi
point_history_classi
point_history_classi
__init__.py
utils
__init__.py
cvfpscalc.py
.gitignore
app.py
keypoint_classification.i
keypoint_classification_E
point_history_classificat
README.md
training_aid

point_history_classifier.py | point_history.csv | keypoint_classification_EN.ipynb | keypoint_classifier.py | point_history_classification.ipynb | getMod

PAT_mining/Gesture-recognition/model/point_history_classifier/point_history_classifier.py › class PointHistoryClassifier › def __call__

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import numpy as np
import tensorflow as tf


class PointHistoryClassifier(object):
    def __init__(
        self,
        model_path='model/point_history_classifier/point_history_classifier.tflite',
        score_th=0.5,
        invalid_value=0,
        self.interpreter.allocate_tensors()
        self.input_details = self.interpreter.get_input_details()
        self.output_details = self.interpreter.get_output_details()

        self.score_th = score_th
        self.invalid_value = invalid_value

    def __call__(
        self,
        point_history,
    ):
        input_details_tensor_index = self.input_details[0]['index']
        self.interpreter.set_tensor(
            input_details_tensor_index,
            np.array([point_history], dtype=np.float32))
        self.interpreter.invoke()

        output_details_tensor_index = self.output_details[0]['index']

        result = self.interpreter.get_tensor(output_details_tensor_index)

        result_index = np.argmax(np.squeeze(result))

        if np.squeeze(result)[result_index] < self.score_th:
            result_index = self.invalid_value

        return result_index
```

projects — zsh | PAT_mining — zsh