Voca

Revolutionizing Mobility with Voice-Activated, Non-invasive Prosthetics

Anish Suvarna, Aarav Gupta, Srikar Kode
(5/21/2025 - Systems Lab, Dr. Yilmaz)

Motivation

22.3 Million People Globally Suffer from Limb Loss

Problem

Patient Complications

Invasive EMG sensors cause skin discomfort, fatigue, and mobility issues

High-Cost

Current upper-extremity prosthetics are unaffordable, costing $15,000-80,000

Stigma

Patients face self-esteem issues and lack an emotional bond to their prosthetic limb

Existing Solutions

Taska

Covvi

Ottobock

Ossur

Competitive Analysis

	Taska	Covvi Nexus	Ottobock	Ossur	Voca
Cost	✗	✗	✗	✗	✓
Haptic Feedback	✓	✗	✗	✗	✓
Comfort	✗	✗	✗	✓	✓
Functionality	✓	✓	✓	✓	✓
Durability	✓	✓	✗	✓	✓
Voice Activation	✗	✗	✗	✗	✓

Our Solution: Voca

Reimagining Prosthetics with AI, Voice & Compassion”

- Voice-activated, 3D-printed, fully customizable prosthetic arm
- Under $1500 budget
- Machine learning–driven grasping
- Natural language processing for intuitive control
- Carbon fiber for aesthetics and reduced stigma
- Emotional feedback through voice + haptics

[redacted]

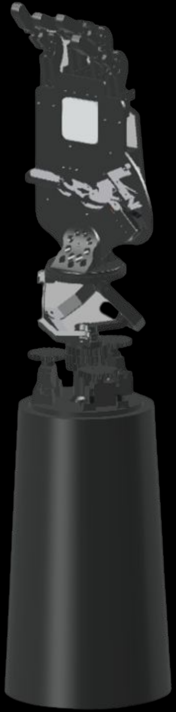
Voca : A Voice-Activated, Multi-Articulating, and Non-Invasive Prosthetic Hand

AI-Based Autonomous Grasping

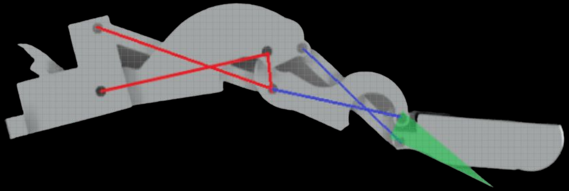
Natural Language Processing, Voice Activation

Cora

ONBOARD VIRTUAL ASSISTANT

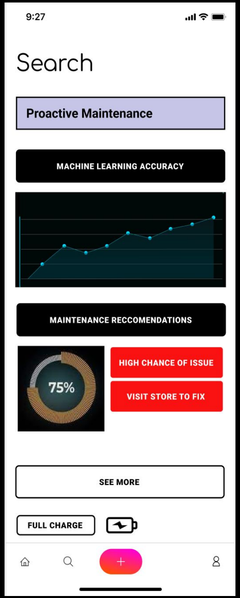


Dynamic Safety and Control



Haptic and Audio Feedback

Mobile App



9:27

Search

Proactive Maintenance

MACHINE LEARNING ACCURACY

MAINTENANCE RECCOMENDATIONS

75%

HIGH CHANCE OF ISSUE

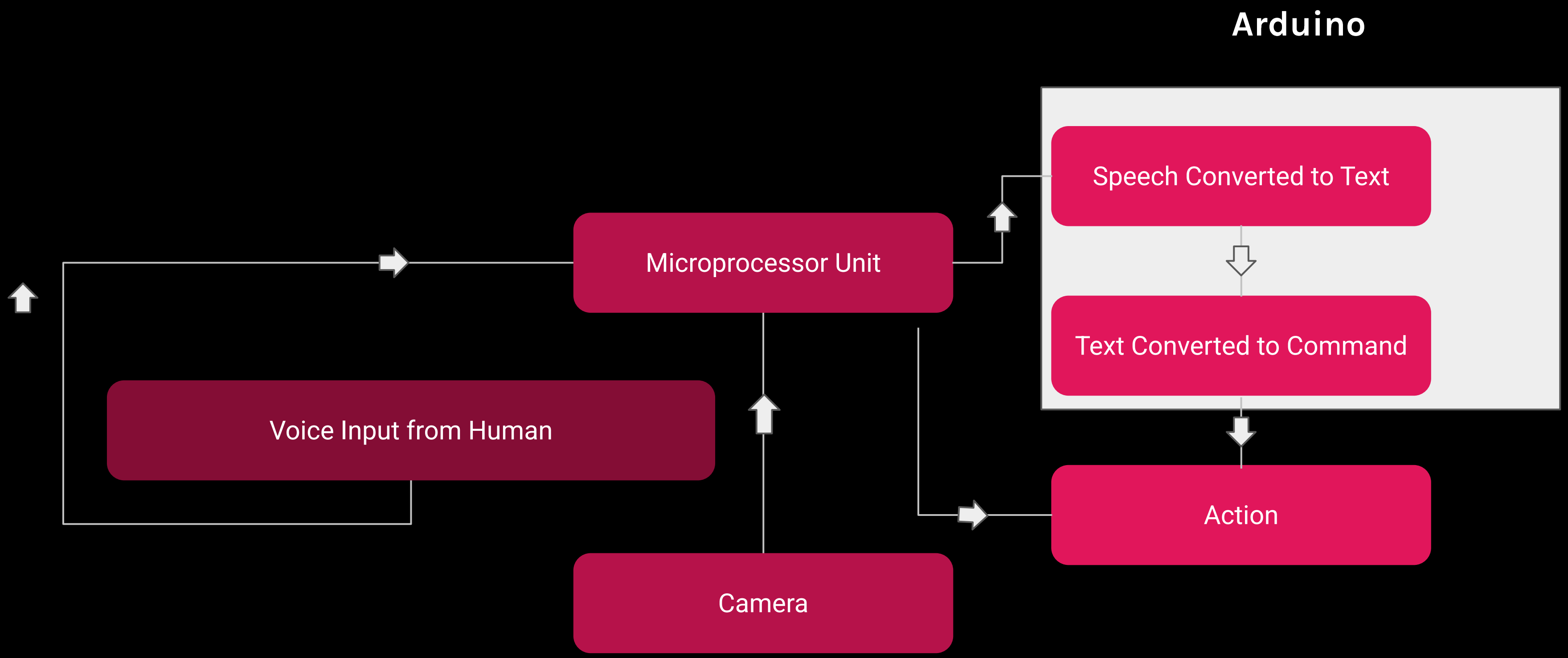
VISIT STORE TO FIX

SEE MORE

FULL CHARGE

Methods

High-level Overview of the System



Arduino

Speech Converted to Text

Text Converted to Command

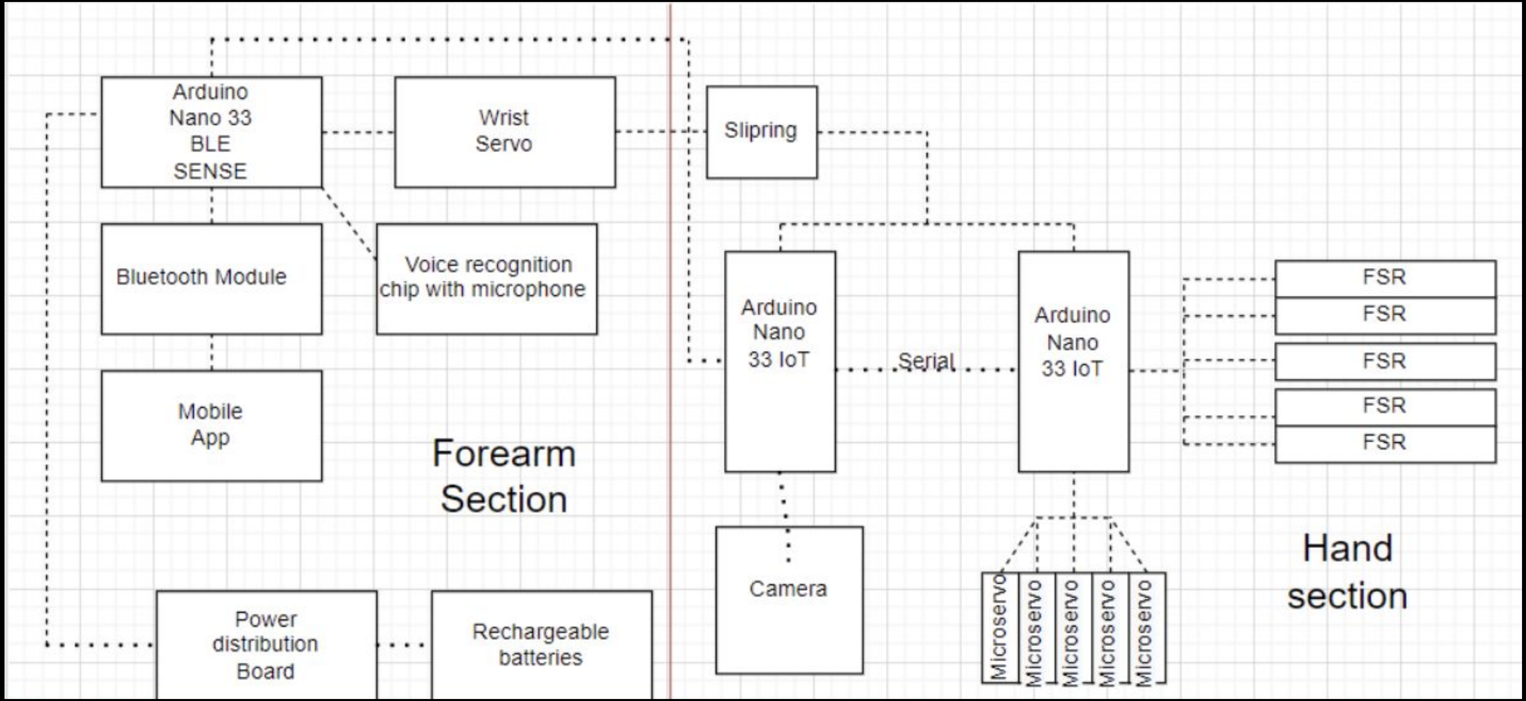
Microprocessor Unit

Voice Input from Human

Camera

Action

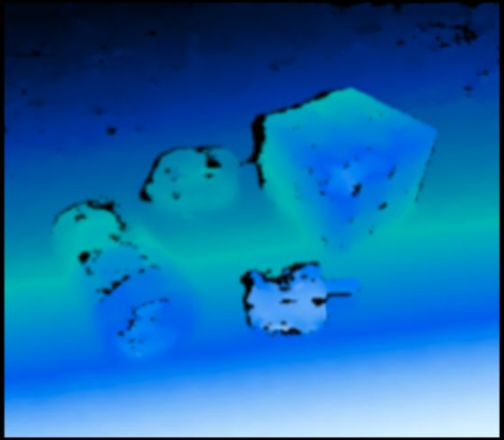
Control Systems Diagram



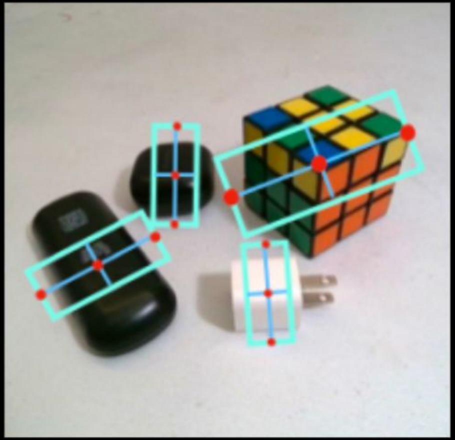
AI Based Grasping Workflow

3D Image Data Collection

Target Object Localization

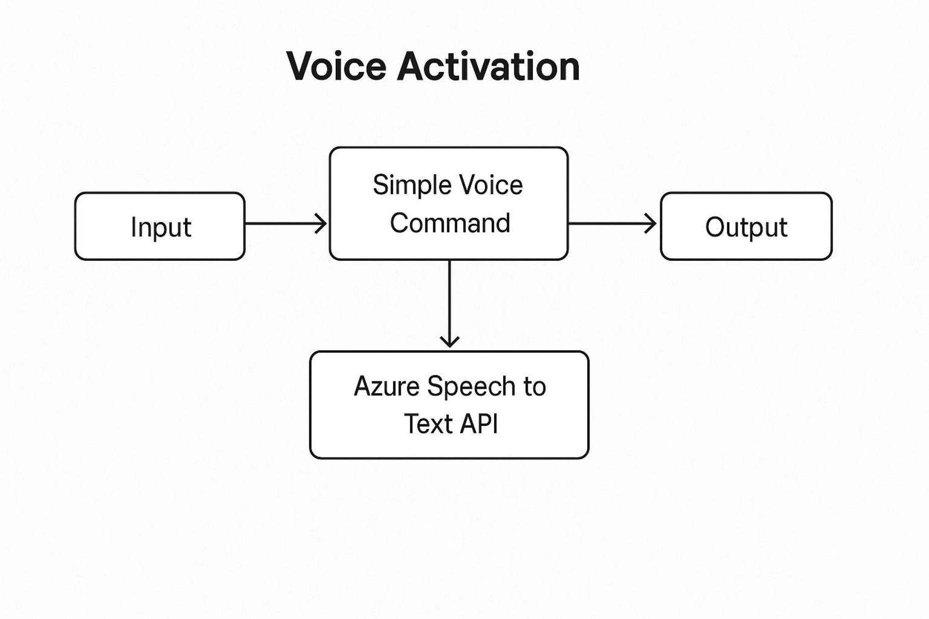


Grasping Pose + Prediction

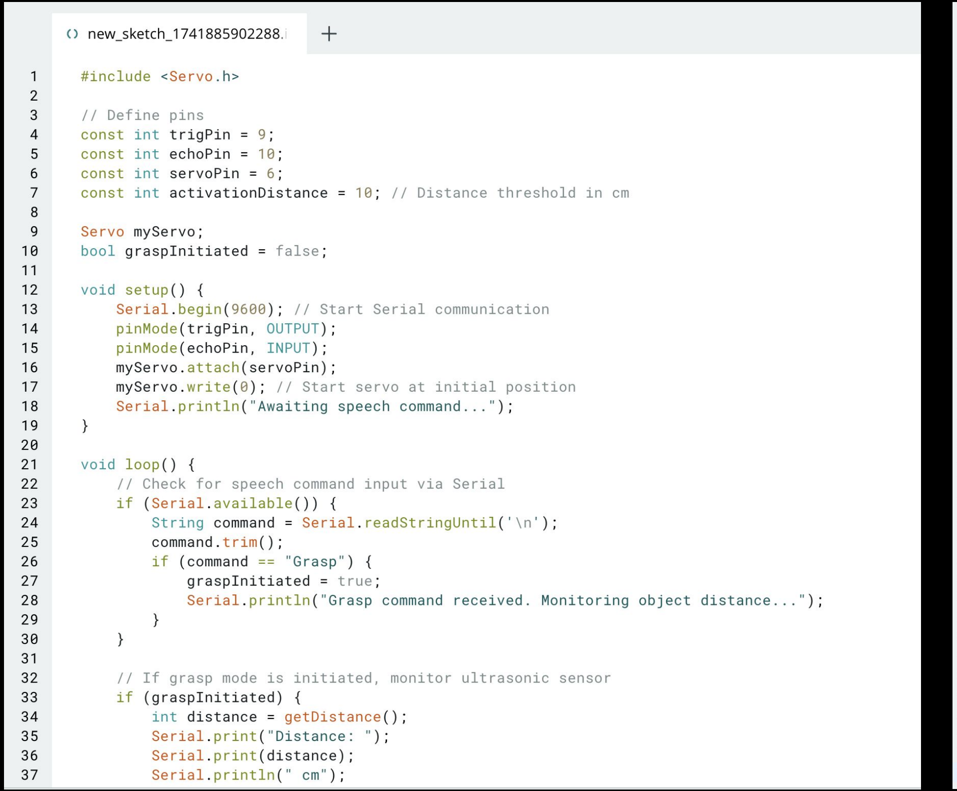


Generative Residual Convolutional Neural Network

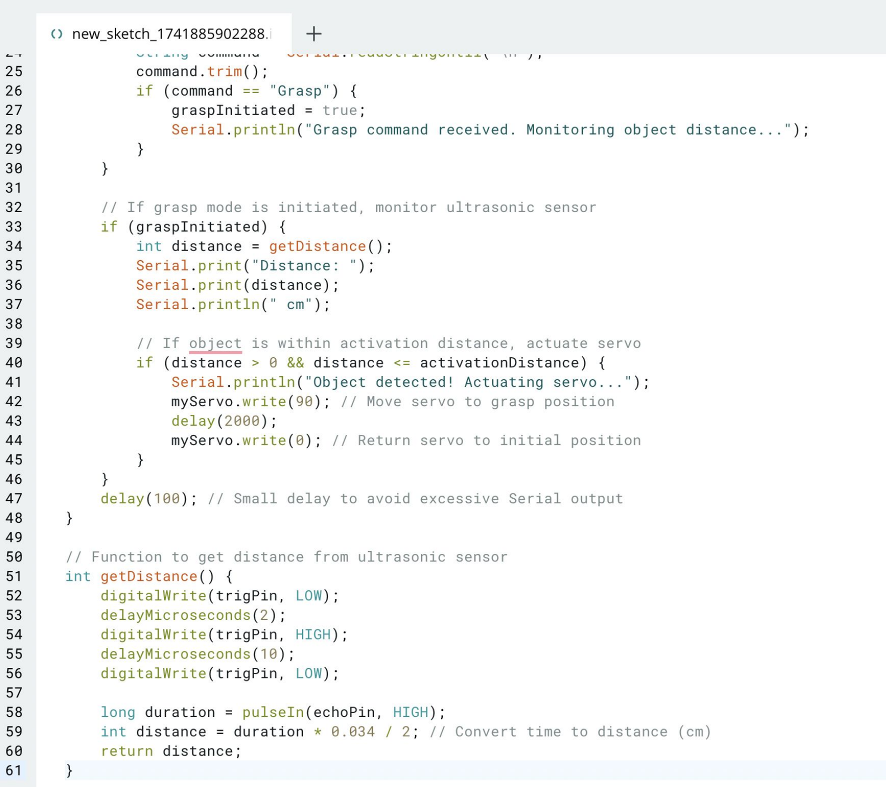
Voice Activation Workflow



Command-based Servo Actuation Code



```
#include <Servo.h>

// Define pins
const int trigPin = 9;
const int echoPin = 10;
const int servoPin = 6;
const int activationDistance = 10; // Distance threshold in cm

Servo myServo;
bool graspInitiated = false;

void setup() {
    Serial.begin(9600); // Start Serial communication
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    myServo.attach(servoPin);
    myServo.write(0); // Start servo at initial position
    Serial.println("Awaiting speech command...");
}

void loop() {
    // Check for speech command input via Serial
    if (Serial.available()) {
        String command = Serial.readStringUntil('\n');
        command.trim();
        if (command == "Grasp") {
            graspInitiated = true;
            Serial.println("Grasp command received. Monitoring object distance...");
        }
    }

    // If grasp mode is initiated, monitor ultrasonic sensor
    if (graspInitiated) {
        int distance = getDistance();
        Serial.print("Distance: ");
        Serial.print(distance);
        Serial.println(" cm");
```

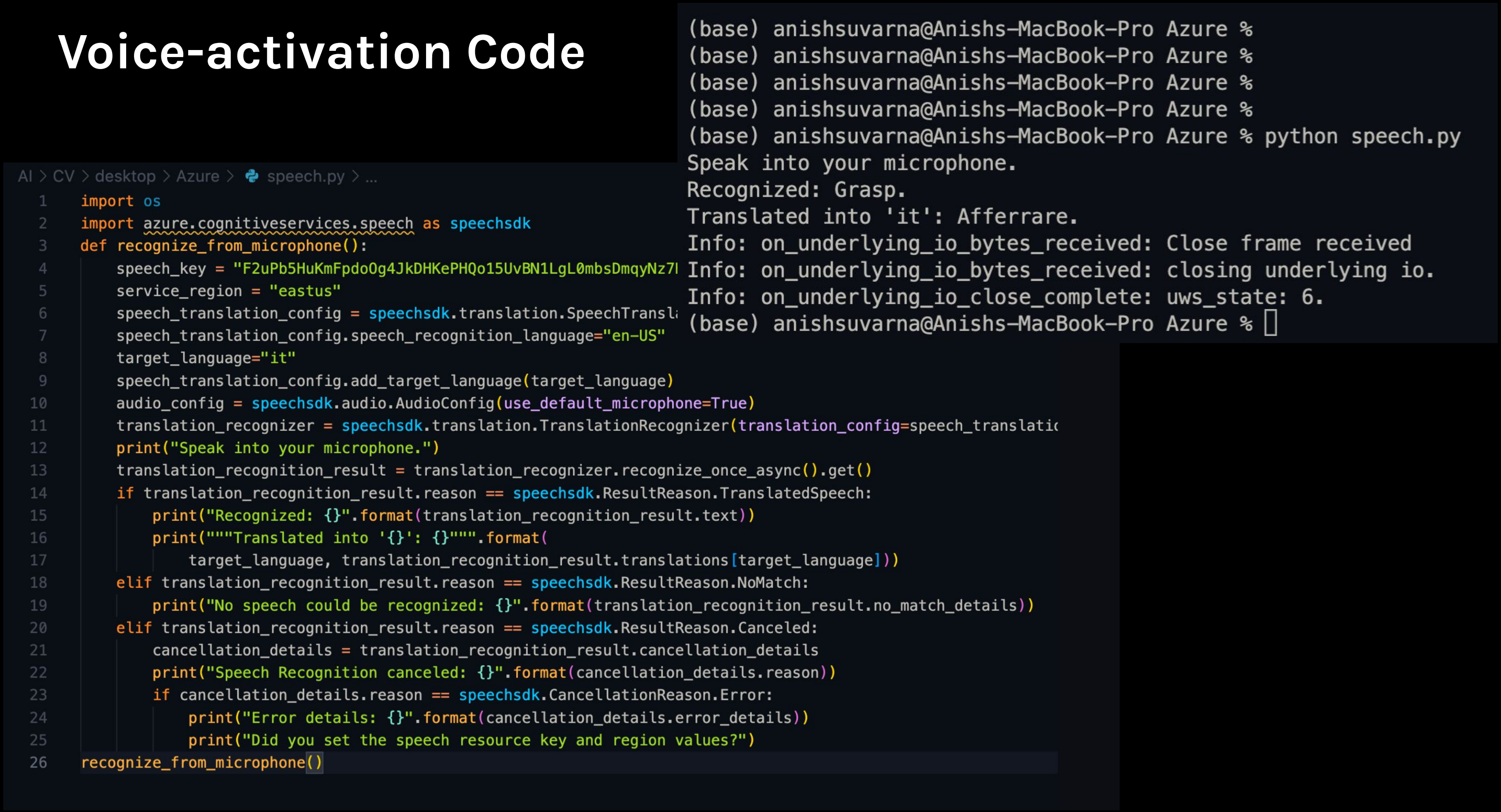


```
        command.trim();
        if (command == "Grasp") {
            graspInitiated = true;
            Serial.println("Grasp command received. Monitoring object distance...");
        }
    }

    // If grasp mode is initiated, monitor ultrasonic sensor
    if (graspInitiated) {
        int distance = getDistance();
        Serial.print("Distance: ");
        Serial.print(distance);
        Serial.println(" cm");
```

```

        // If object is within activation distance, actuate servo
        if (distance > 0 && distance <= activationDistance) {
            Serial.println("Object detected! Actuating servo...");
            myServo.write(90); // Move servo to grasp position
            delay(2000);
            myServo.write(0); // Return servo to initial position
        }
    }
    delay(100); // Small delay to avoid excessive Serial output
}

// Function to get distance from ultrasonic sensor
int getDistance() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH);
    int distance = duration * 0.034 / 2; // Convert time to distance (cm)
    return distance;
}
```

Results

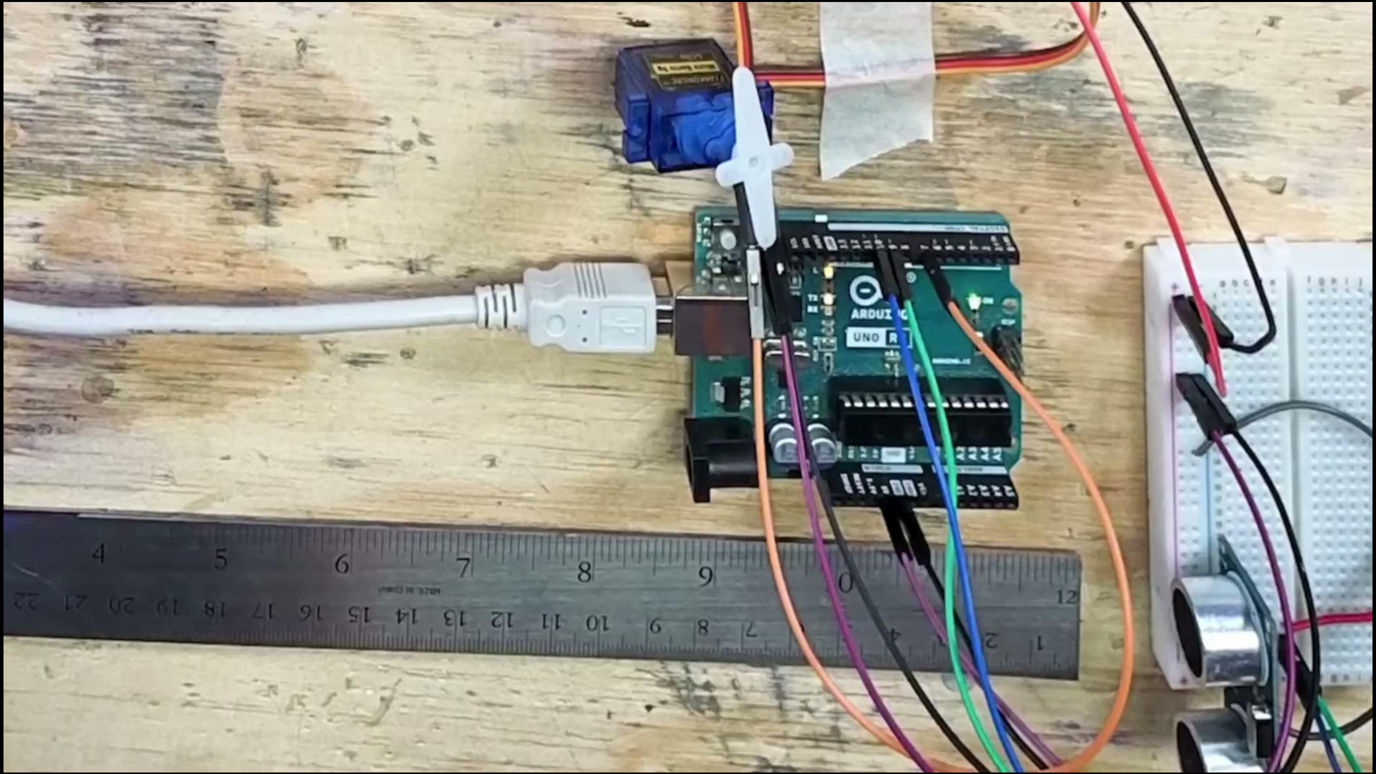
Voice-activation Code



```
(base) anishsuvarna@Anishs-MacBook-Pro Azure %
(base) anishsuvarna@Anishs-MacBook-Pro Azure %
(base) anishsuvarna@Anishs-MacBook-Pro Azure %
(base) anishsuvarna@Anishs-MacBook-Pro Azure %
(base) anishsuvarna@Anishs-MacBook-Pro Azure % python speech.py
Speak into your microphone.
Recognized: Grasp.
Translated into 'it': Afferrare.
Info: on_underlying_io_bytes_received: Close frame received
Info: on_underlying_io_bytes_received: closing underlying io.
Info: on_underlying_io_close_complete: uws_state: 6.
(base) anishsuvarna@Anishs-MacBook-Pro Azure %
```

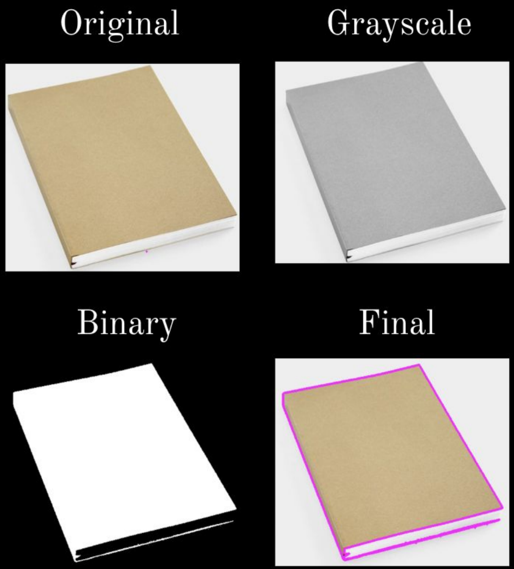
```python
import os
import azure.cognitiveservices.speech as speechsdk
def recognize_from_microphone():
    speech_key = "F2uPb5HuKmFpdoOg4JkDHKePHQo15UvBN1LgL0mbsDmqyNz7
    service_region = "eastus"
    speech_translation_config = speechsdk.translation.SpeechTransl
    speech_translation_config.speech_recognition_language="en-US"
    target_language="it"
    speech_translation_config.add_target_language(target_language)
    audio_config = speechsdk.audio.AudioConfig(use_default_microphone=True)
    translation_recognizer = speechsdk.translation.TranslationRecognizer(translation_config=speech_translatic
    print("Speak into your microphone.")
    translation_recognition_result = translation_recognizer.recognize_once_async().get()
    if translation_recognition_result.reason == speechsdk.ResultReason.TranslatedSpeech:
        print("Recognized: {}".format(translation_recognition_result.text))
        print("""Translated into '{}': {}""".format(
            target_language, translation_recognition_result.translations[target_language]))
    elif translation_recognition_result.reason == speechsdk.ResultReason.NoMatch:
        print("No speech could be recognized: {}".format(translation_recognition_result.no_match_details))
    elif translation_recognition_result.reason == speechsdk.ResultReason.Canceled:
        cancellation_details = translation_recognition_result.cancellation_details
        print("Speech Recognition canceled: {}".format(cancellation_details.reason))
        if cancellation_details.reason == speechsdk.CancellationReason.Error:
            print("Error details: {}".format(cancellation_details.error_details))
            print("Did you set the speech resource key and region values?")
recognize_from_microphone()
```

Voice-activation Demonstration

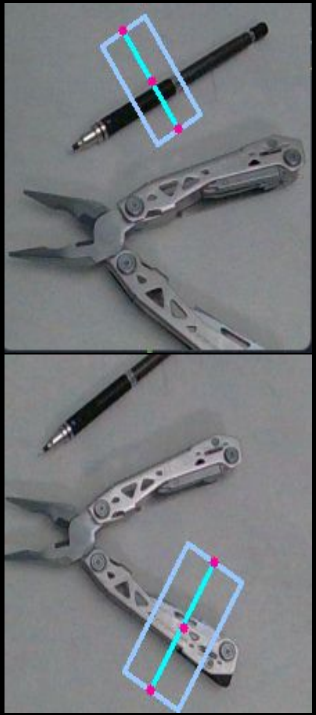


AI-based Grasping Output

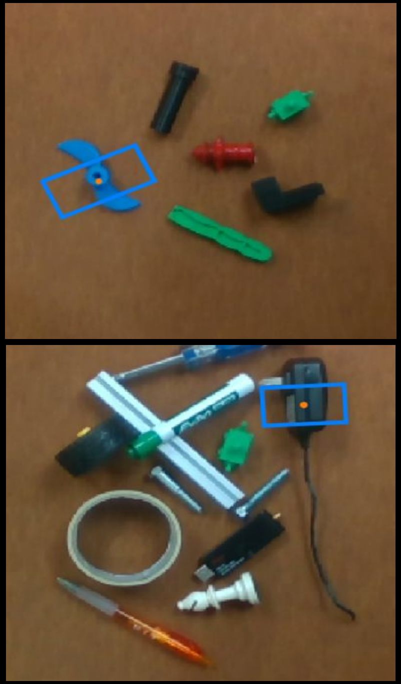
Object Localization



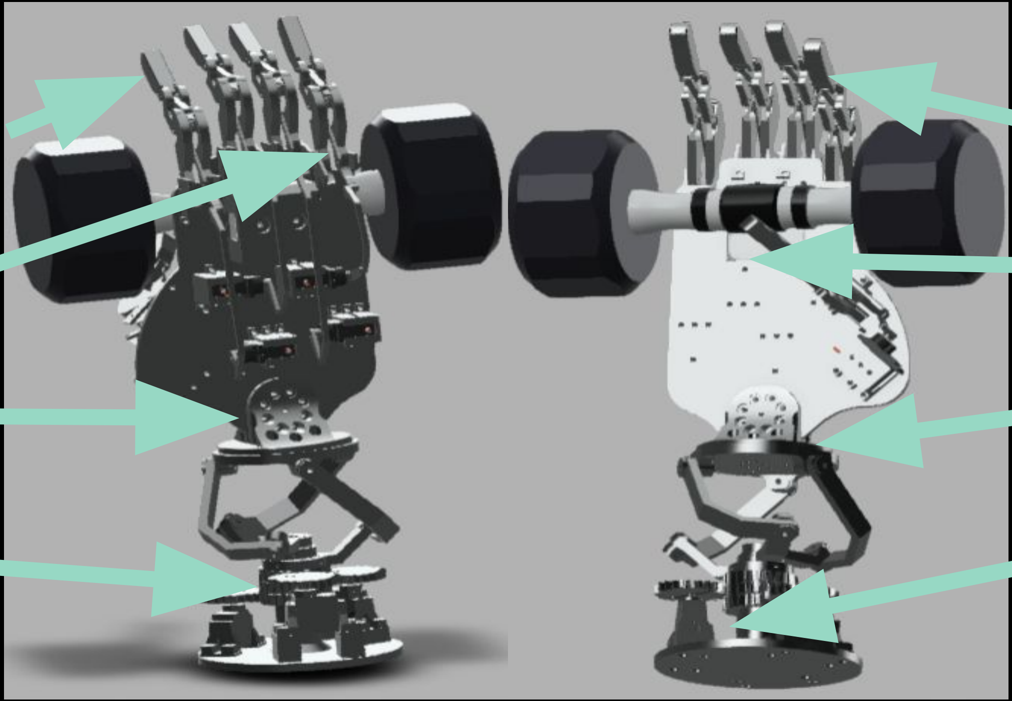
6D Grasping Pose



Grasping in Clutter



CAD Design, Multi-stage Prototype



Multi-Articulation Replicating Human Hand

4-Bar-Linkage System for Mechanical Optimization

PID Control-Based Stability Control for Fingers/Wrist

Spherical Parallel Manipulator for Wrist

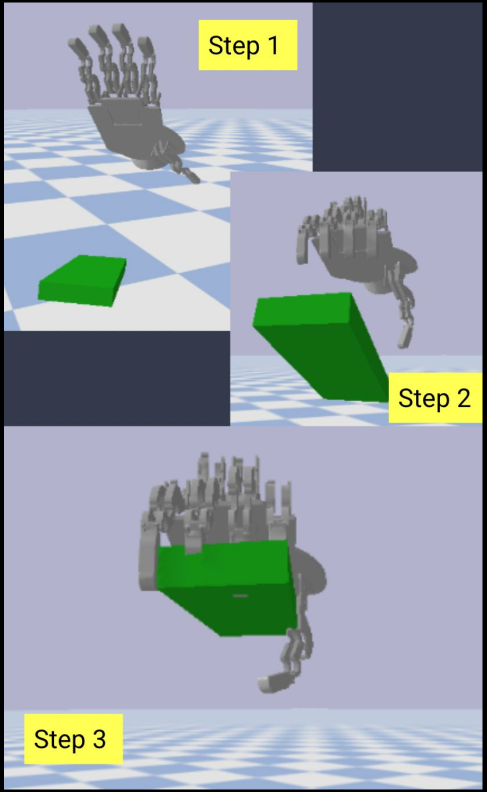
Inverse Kinematics for Finger Motion Planning

Depth Camera for Autonomous Grasping

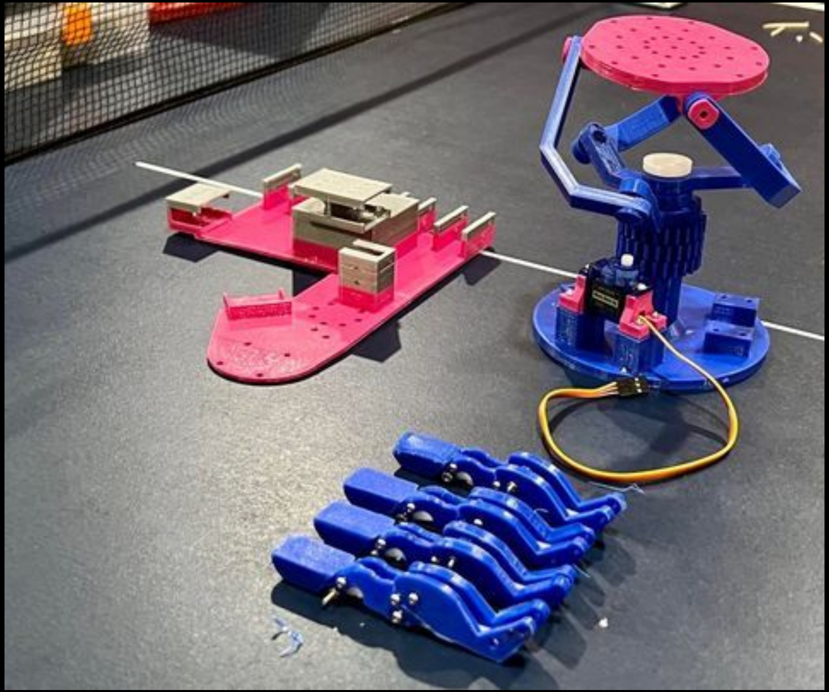
Microphone for Onboard, NLP-Based Voice Assistant

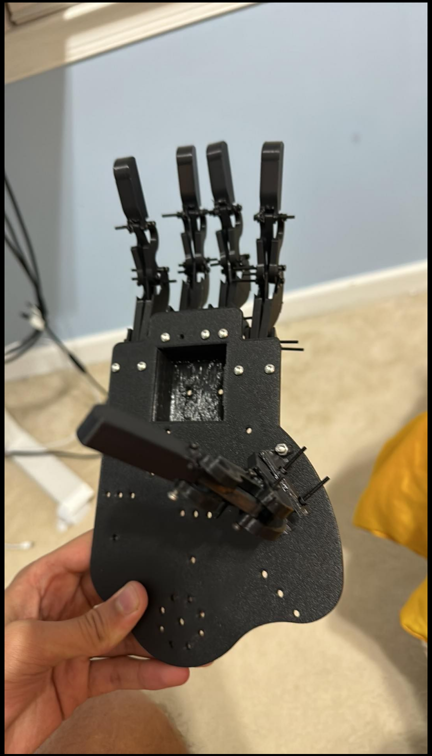
Tactile Sensors for Haptic and Audio Feedback

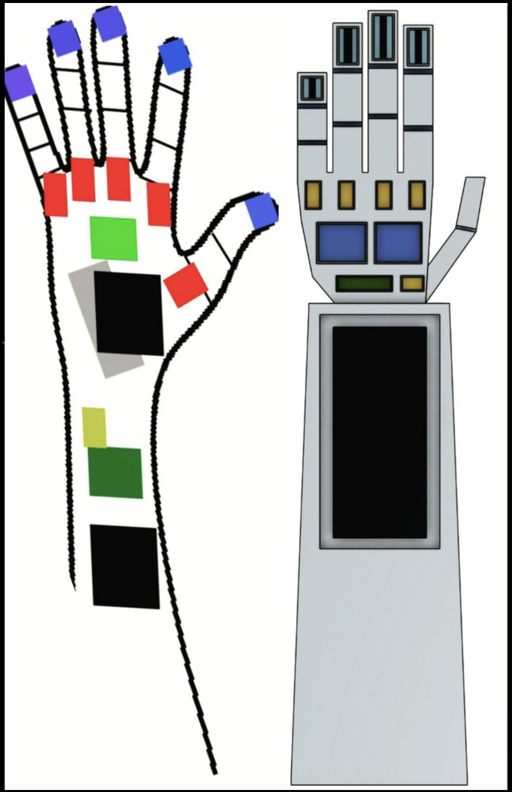
Inverse Kinematics Simulation Demonstration



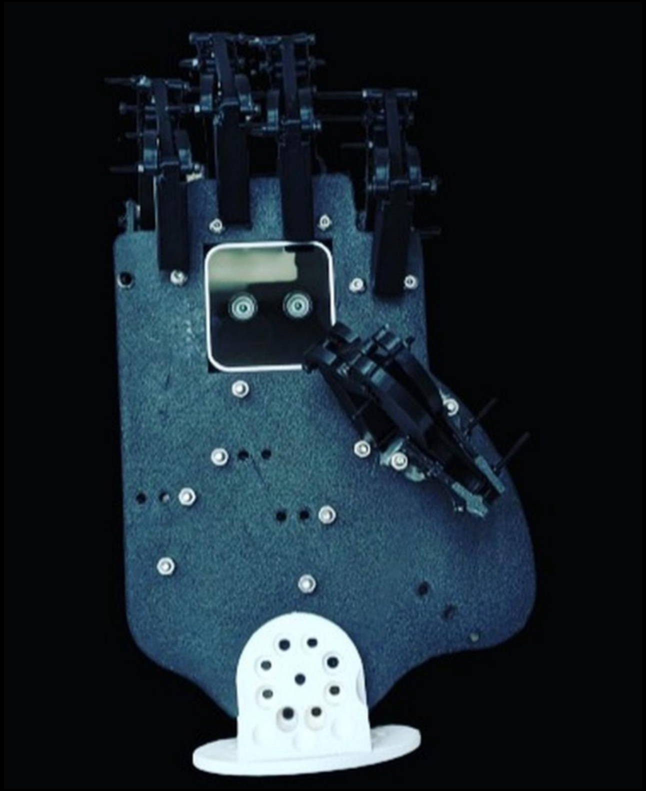
Manufacturing Progression



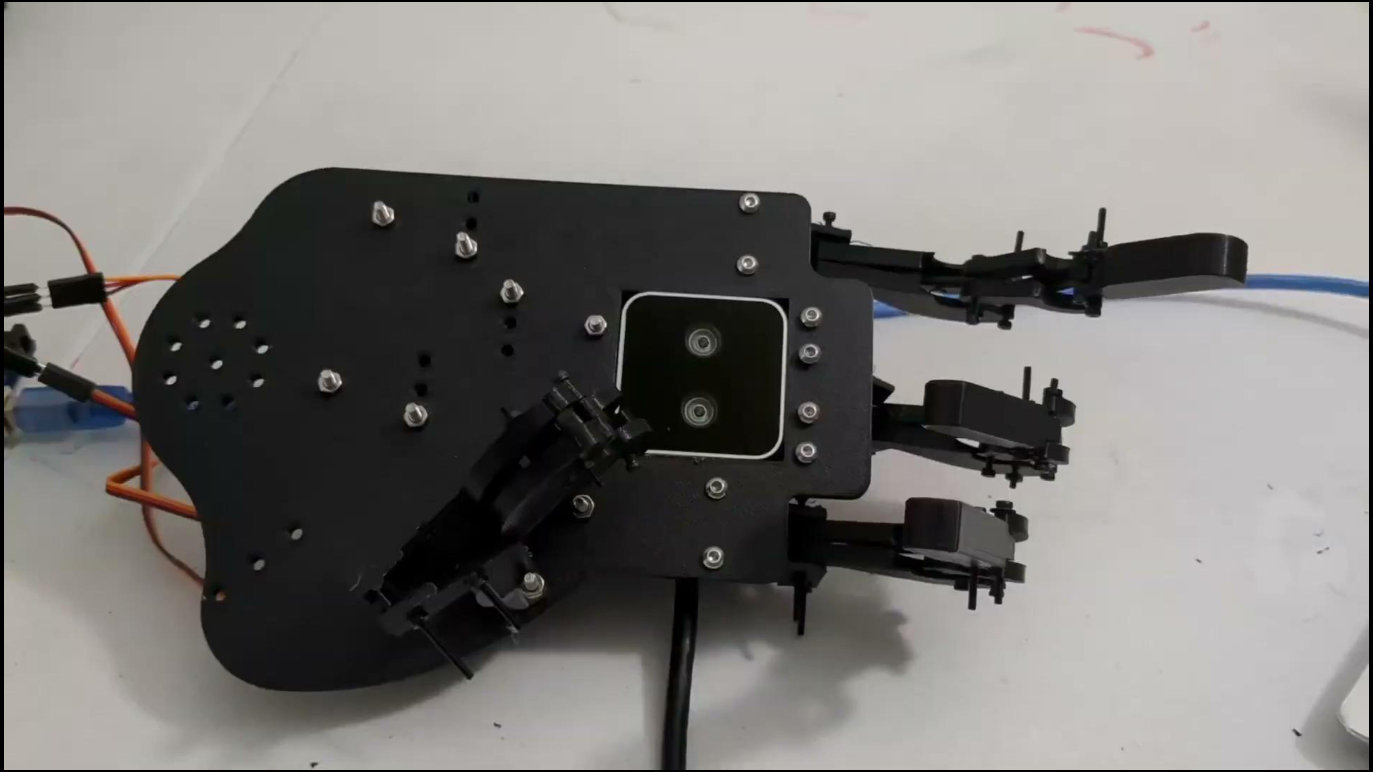




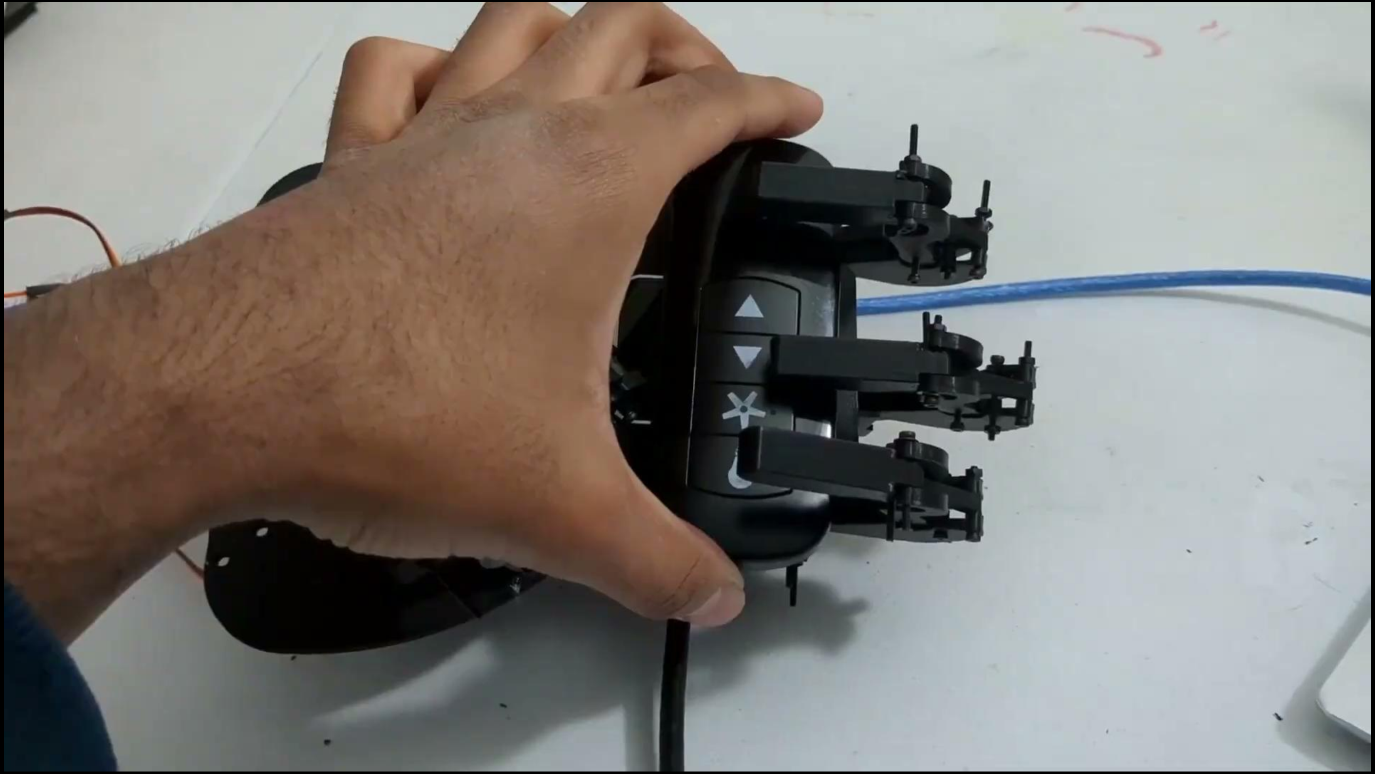
Final Hand



Final Grasp Demonstration



Final Release Demonstration



Limitations

- No brain interface (non-invasive)
- Material doesn’t fully mimic human skin
- Computational power is constrained by current chip size

Conclusion

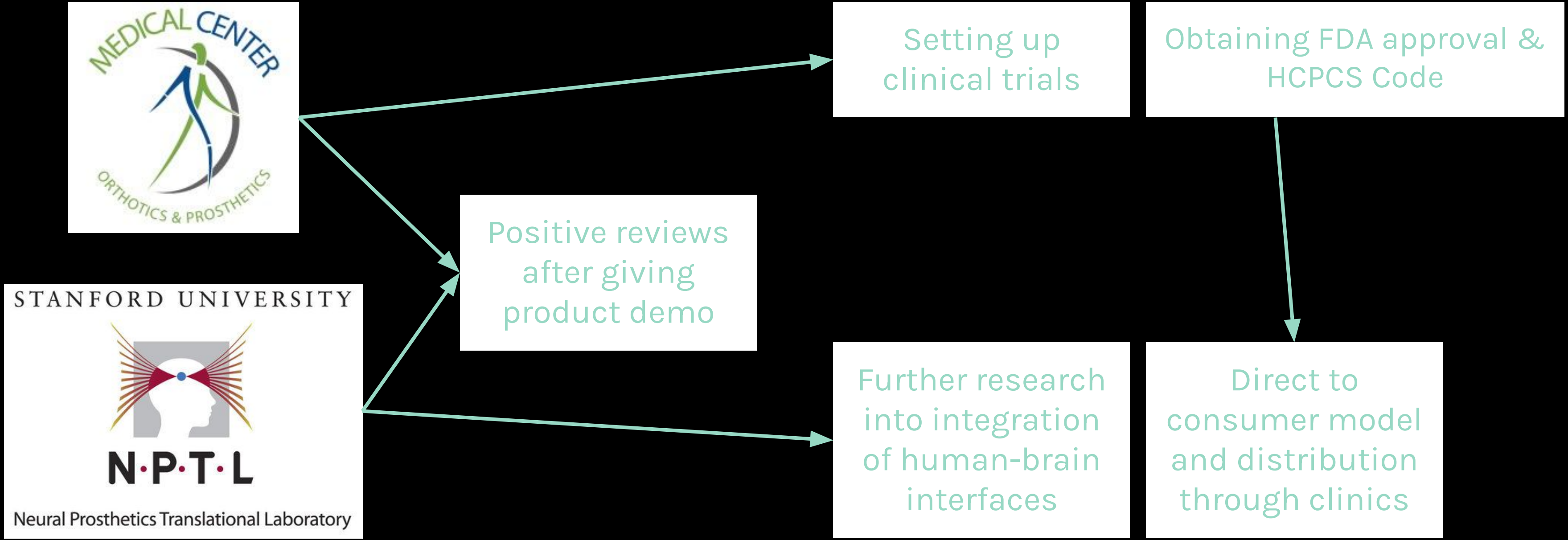
"Rebuilding Dignity, One Voice Command at a Time"

Voca enables users to perform basic daily tasks using just their voice

Fuses engineering precision with emotional empathy

Future-proof platform for next-gen prosthetics

Future Directions



Positive reviews after giving product demo

Setting up clinical trials

Obtaining FDA approval & HCPCS Code

Further research into integration of human-brain interfaces

Direct to consumer model and distribution through clinics

Thank you!

Q & A