# PES UNIVERSITY, Bangalore

(Established under Karnataka Act No. 16 of 2013)

**Department of Computer Science & Engineering**

**Mobile and Autonomous Robots (UE22CS343BB7)**

**6th Semester**

## Mini-Project

**Project Title:**

# Voice Controlled Robotic Arm

## Team Details:

**AARUSHI (PES2UG22CS008)**

**ADITYA VINAYAK BHAT(PES2UG22CS037)**

**APEKSHA MAHANTESH MATHAPATI (PES2UG22CS091)**

**DEEKSHA SUKUMAR JAIN (PES2UG22CS164)**

## Professor Name: Dr. Gokul Kannan

## Project Description:

This project implements a voice-controlled robotic arm simulation using the Webots robotics simulator and Python. The system allows users to control a robotic arm's movements and gripper actions through natural voice commands, making human-robot interaction more intuitive and accessible.

**Project Objectives:**

Voice Command Recognition

- Utilizes the SpeechRecognition library to process audio input
- Supports common commands: up, down, left, right stop, position
- Optimized for both Windows and WSL environments with automatic environment detection

Robotic Arm Control

- Multi-joint arm with precise position control;
- Enhanced dual-motor gripper mechanism for better object manipulation
- Predefined positions for common movements
- Real-time position feedback through integrated sensors

Cross-Platform Compatibility

- Works on standard Windows installations
- Special WSL (Windows Subsystem for Linux) integration using PowerShell commands
- Debug mode with text input for testing without microphone

## Methods and Materials:

### 1. System Design

The system uses a client-server architecture where the voice recognition module acts as a client that sends commands to a Webots simulation server. The design supports both Windows native and WSL environments through adaptive audio capture methods. Commands are transmitted via TCP socket connection to the Webots controller which then manipulates the robotic arm.

### 2. Algorithm/Model Development

The system employs a command recognition algorithm that:

- Captures audio input through microphone
- Processes speech using Google's speech recognition API
- Matches recognized text against a predefined set of valid commands
- Converts matched commands into action directives for the robotic arm
- Supports commands like "up", "down", "left", "right", "open", "close" and “home”.

### 3. Implementation Steps

- Audio capture via native microphone or Windows audio tools (WSL mode)
- Speech-to-text conversion using Google's speech recognition service
- Command extraction from recognized text
- JSON command formatting and transmission via TCP socket
- Response handling from the Webots controller

### 4. Hardware Components (if applicable)

- Microphone for voice input
- Computer running Windows or Linux with WSL
- Webots simulation environment
- Virtual or physical robotic arm compatible with Webots

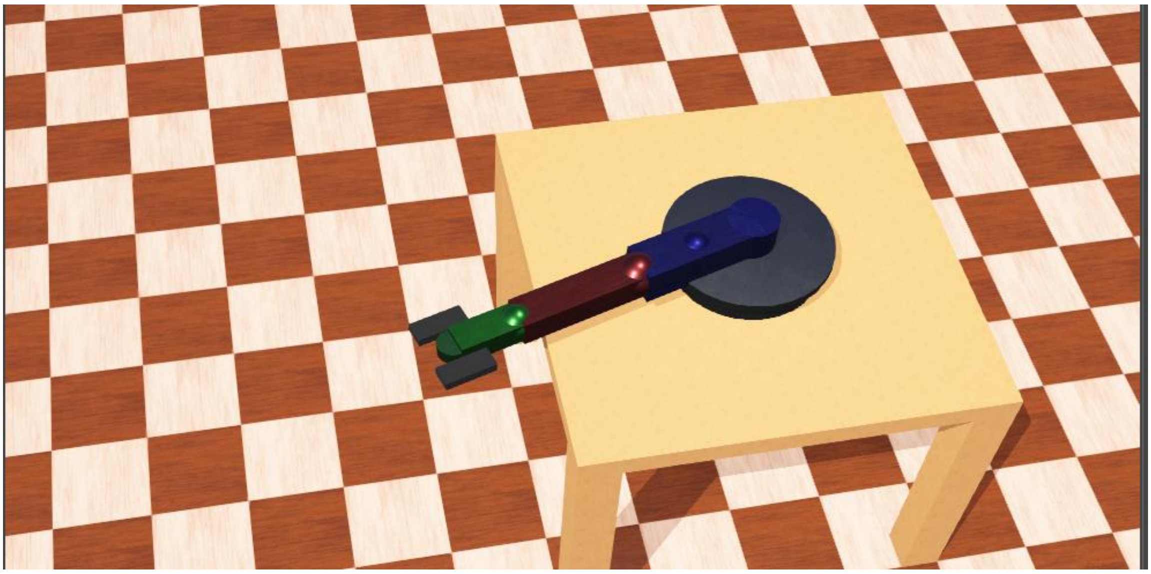
5. **Software Tools**

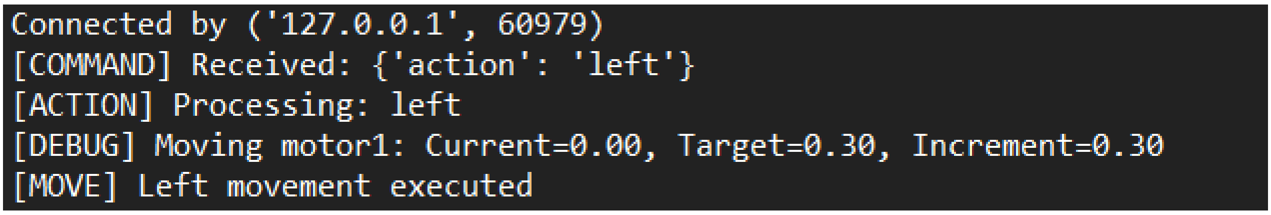
- Python 3 with libraries:
- speech_recognition for voice processing
- socket for network communication
- pyaudio for audio capture (non-WSL environments)
- Webots robot simulation platform
- Windows PowerShell
- JSON for command serialization
- Command-line interface with debug and standard modes

**Project Outcome:**

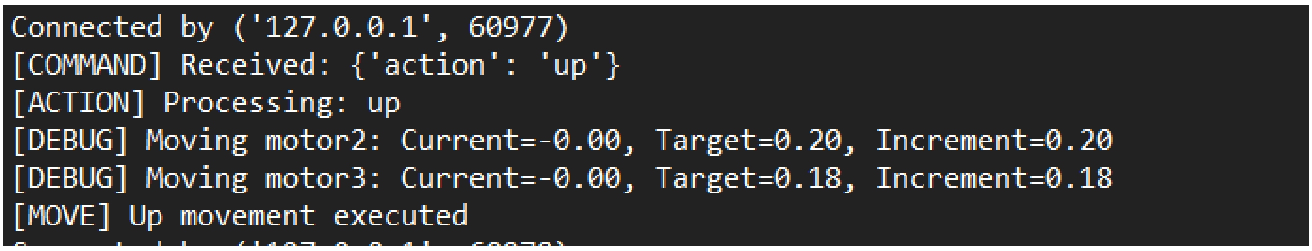
1. **Output results** –

**Server Side**:

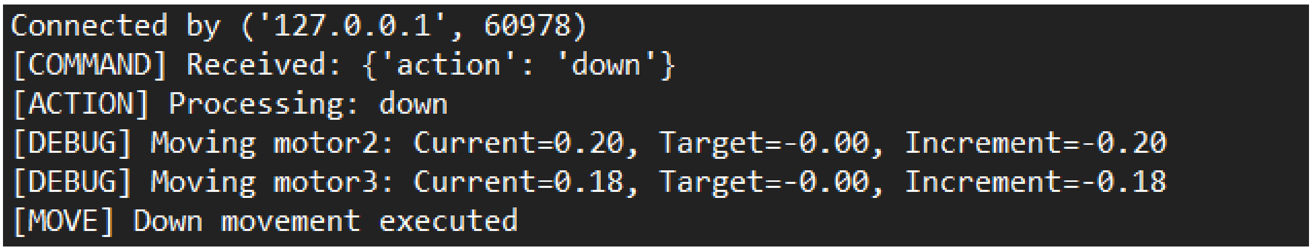




```
Connected by ('127.0.0.1', 60979)
[COMMAND] Received: {'action': 'left'}
[ACTION] Processing: left
[DEBUG] Moving motor1: Current=0.00, Target=0.30, Increment=0.30
[MOVE] Left movement executed
```

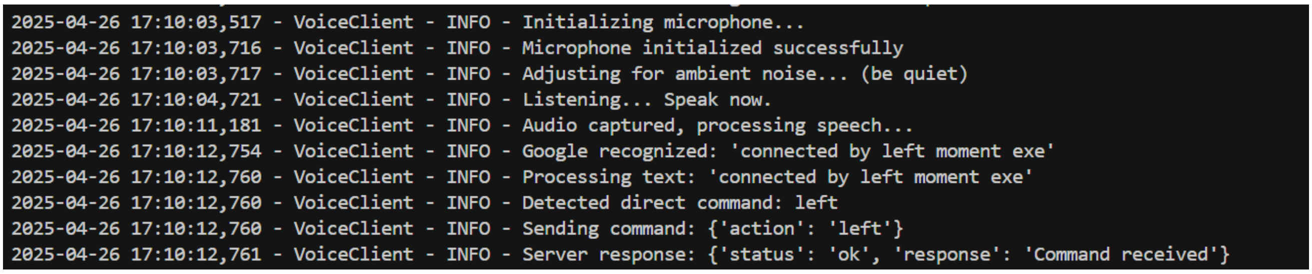


```
Connected by ('127.0.0.1', 60977)
[COMMAND] Received: {'action': 'up'}
[ACTION] Processing: up
[DEBUG] Moving motor2: Current=-0.00, Target=0.20, Increment=0.20
[DEBUG] Moving motor3: Current=-0.00, Target=0.18, Increment=0.18
[MOVE] Up movement executed
```

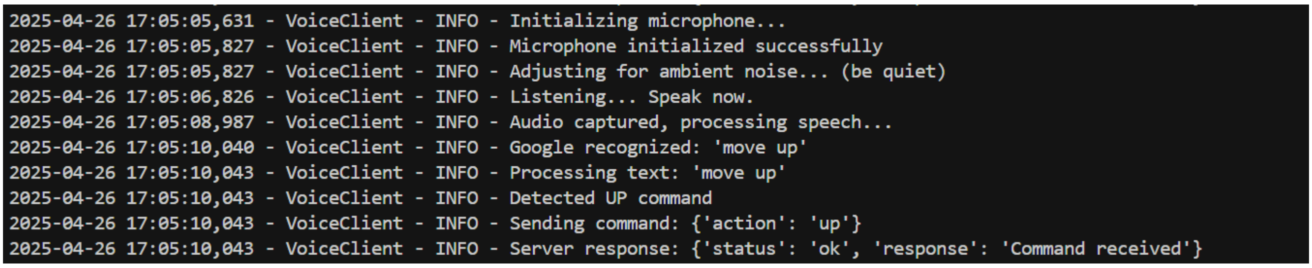


```
Connected by ('127.0.0.1', 60978)
[COMMAND] Received: {'action': 'down'}
[ACTION] Processing: down
[DEBUG] Moving motor2: Current=0.20, Target=-0.00, Increment=-0.20
[DEBUG] Moving motor3: Current=0.18, Target=-0.00, Increment=-0.18
[MOVE] Down movement executed
```

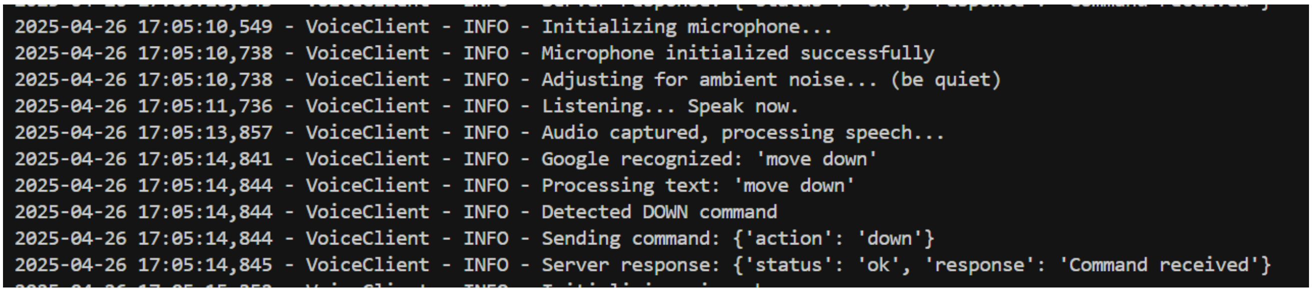
**Client Side:**



```
2025-04-26 17:10:03,517 - VoiceClient - INFO - Initializing microphone...
2025-04-26 17:10:03,716 - VoiceClient - INFO - Microphone initialized successfully
2025-04-26 17:10:03,717 - VoiceClient - INFO - Adjusting for ambient noise... (be quiet)
2025-04-26 17:10:04,721 - VoiceClient - INFO - Listening... Speak now.
2025-04-26 17:10:11,181 - VoiceClient - INFO - Audio captured, processing speech...
2025-04-26 17:10:12,754 - VoiceClient - INFO - Google recognized: 'connected by left moment exe'
2025-04-26 17:10:12,760 - VoiceClient - INFO - Processing text: 'connected by left moment exe'
2025-04-26 17:10:12,760 - VoiceClient - INFO - Detected direct command: left
2025-04-26 17:10:12,760 - VoiceClient - INFO - Sending command: {'action': 'left'}
2025-04-26 17:10:12,761 - VoiceClient - INFO - Server response: {'status': 'ok', 'response': 'Command received'}
```



```
2025-04-26 17:05:05,631 - VoiceClient - INFO - Initializing microphone...
2025-04-26 17:05:05,827 - VoiceClient - INFO - Microphone initialized successfully
2025-04-26 17:05:05,827 - VoiceClient - INFO - Adjusting for ambient noise... (be quiet)
2025-04-26 17:05:06,826 - VoiceClient - INFO - Listening... Speak now.
2025-04-26 17:05:08,987 - VoiceClient - INFO - Audio captured, processing speech...
2025-04-26 17:05:10,040 - VoiceClient - INFO - Google recognized: 'move up'
2025-04-26 17:05:10,043 - VoiceClient - INFO - Processing text: 'move up'
2025-04-26 17:05:10,043 - VoiceClient - INFO - Detected UP command
2025-04-26 17:05:10,043 - VoiceClient - INFO - Sending command: {'action': 'up'}
2025-04-26 17:05:10,043 - VoiceClient - INFO - Server response: {'status': 'ok', 'response': 'Command received'}
```



```
2025-04-26 17:05:10,549 - VoiceClient - INFO - Initializing microphone...
2025-04-26 17:05:10,738 - VoiceClient - INFO - Microphone initialized successfully
2025-04-26 17:05:10,738 - VoiceClient - INFO - Adjusting for ambient noise... (be quiet)
2025-04-26 17:05:11,736 - VoiceClient - INFO - Listening... Speak now.
2025-04-26 17:05:13,857 - VoiceClient - INFO - Audio captured, processing speech...
2025-04-26 17:05:14,841 - VoiceClient - INFO - Google recognized: 'move down'
2025-04-26 17:05:14,844 - VoiceClient - INFO - Processing text: 'move down'
2025-04-26 17:05:14,844 - VoiceClient - INFO - Detected DOWN command
2025-04-26 17:05:14,844 - VoiceClient - INFO - Sending command: {'action': 'down'}
2025-04-26 17:05:14,845 - VoiceClient - INFO - Server response: {'status': 'ok', 'response': 'Command received'}
```

2. **Simulation video link (drive link)-** https://drive.google.com/drive/folders/1O-AQr_Lyc7wU7Vc0apa2eKxlY8c2newq?usp=drive_link

3. **GitHub link (Source code)-** https://github.com/ApekshaMathapati/Voice_controlled_robotic-_arm.git

**References:**

1. Quigley, M., Conley, K., Gerkey, B., et al. (2009). ROS: an open-source Robot Operating System. ICRA Workshop on Open Source Software.
2. Michel, O. (2004). Webots: Professional Mobile Robot Simulation. International Journal of Advanced Robotic Systems, 1(1), 39-42.
3. Google Cloud. (2023). Cloud Speech-to-Text Documentation. https://cloud.google.com/speech-to-text/docs
4. Campeau-Lecours, A., Lamontagne, H., Latour, S., et al. (2019). Kinova Modular Robot Arms for Service Robotics Applications. International Journal of Robotics Research, 38(7), 729-750.
5. PyAudio Documentation. (2023). PyAudio: PortAudio v19 Python Bindings. https://people.csail.mit.edu/hubert/pyaudio/docs/
6. SpeechRecognition Documentation. (2023). SpeechRecognition Python Library. https://pypi.org/project/SpeechRecognition/