

Chapter 6

RESULTS AND DISCUSSION

6.1 Eye Mechanism

The eye mechanism developed and tested demonstrates successful integration of hardware and software components, resulting in a vision system capable of tracking and detecting faces effectively while providing smooth motion. Utilizing parts from INMOOV's mechanism, such as SG90 servos and an Arduino Uno, alongside Python programming for flexible control via serial communication, the mechanism achieves a lifelike appearance and functionality. This capability to replicate human senses enhances its utility across various applications, including assistive technology, surveillance systems, and human-robot interaction.



Fig 6.1 Eye movement before



Fig 6.2 Eye movement after

Result:

The Eye mechanism system has been successfully implemented, utilizing hardware and software to build the vision system able to easily track and detect faces. We create an actual functional system by integrating INMOOV mechanism components.

Discussion:

- Component Integration is the combination of SG90 servos, Arduino Uno, And Python resulting in a smooth and effective vision system.

- Realistic and Functional Design is the real eye mechanism that increases the assistive technology interaction between humans and robots.
- Wide Application is effectively face-tracking and face-detection making it valuable in assistive technology.

6.2 Hand Mechanism

The hand mechanism demonstrates successful functionality across its five degrees of motion, utilizing potentiometers for precise servo control. A custom-built teach pendant enhances user interaction, offering tactile control for intuitive manipulation. Through strategic servo selection based on torque requirements, including a 60kgcm servo for shoulder movement and lighter servos for subsequent joints, the mechanism achieves efficient load handling. With control options ranging from manual operation via the teach pendant to remote control via Bluetooth serial communication, the hand mechanism offers flexibility and versatility in its operation.



Fig 6.3 Hand mechanism before

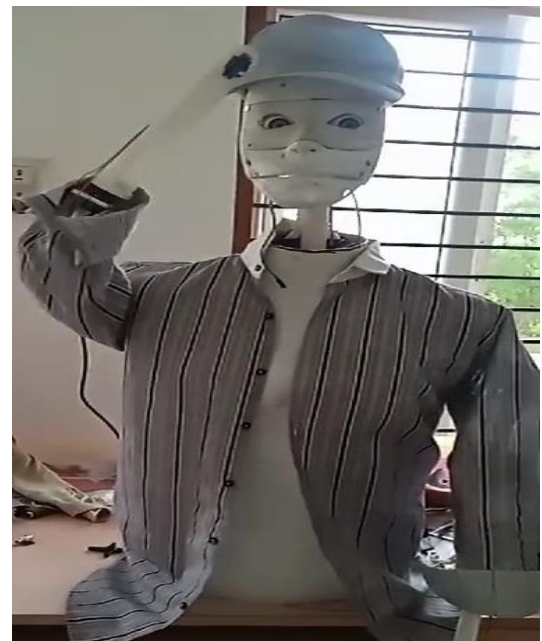


Fig 6.4 Hand mechanism

Result: The hand mechanism is effective and accurate control over the five degrees of motion. Potentiometers ensure accurate servo control and a unique teach pendant provides simple

interaction with servos. Which servos are based on torque requirements, and ability to handle load.

Discussion:

- Effective and Accurate Control is the hand mechanism that accurately controls motion across five degrees to ensure consistency that can be seen in Fig 6.3 above.
- Potentiometers provide precise servo control, improving the mechanism's accuracy and speed.
- A one-of-a-kind teach pendant allows for a simple, connection with the servos, making it easy to use.

6.3 Forward motion

The forward motion system employs two wiper motors operating at a speed of 45rpm, regulated by a 15amp motor driver controlled by an ESP32 microcontroller. The mechanism, situated at the rear end of the frame, is stabilized by two caster wheels at the front. Bluetooth control with speed adjustment facilitates precise maneuvering. With its robust torque capabilities, the wiper motors can reliably bear loads of up to 20-30kg without encountering any issues.



Fig 6.5 Forward motion Before



Fig 6.5 Forward motion After

Result:

The forward motion system uses two wiper motors for a steady 45rpm speed, controlled by a 15amp motor driver and ESP32 microcontroller for smooth operation. Front caster wheels provide stability. Bluetooth control allows precise navigation, and the system handles 20-30kg loads without performance loss.

Discussion:

- Steady speed it has two wiper motors that ensure a constant speed at 45rpm for reliable forward motion.
- For smooth control a 15amp motor driver and ESP32 microcontroller provide smooth operation.
- The front caster wheels add stability and support during the movement.
- When a forward command is given to the robot it takes a single bit of data and accordingly can move for a distance of 10m as shown in Fig 6.5 and Fig 6.6.

6.4 Face Recognition and Tracking

The face recognition system employs Python, OpenCV, and PyFirmata for serial communication with Arduino. Utilizing OpenCV's face detection module, the system detects the presence of a face as shown in Fig 6.7 and Fig 6.8 below also it tracks accordingly. Data containing coordinates for servo control is sent to the Arduino, facilitating precise tracking of the detected face.



Fig 6.7 Face recognition before



Fig 6.8 Face recognition after



Fig 6.9 Face tracking before



Fig 6.10 Face tracking after

Result:

The facial recognition system has been successfully built, with Python and OpenCV for face detection and tracking. It check for the database for the facial feature stored previously if the user is present in the database then it displays the name on the screen as shown in Fig 6.8.

Discussion:

- The facial recognition system is effectively built using Python and OpenCV technologies.
- Python and OpenCV provide robust face detection capabilities for accurate identification.
- The face tracking system reliably tracks faces, ensuring continuous monitoring and recognition.

6.5 Line following

The line-following system utilizes three IR sensors in conjunction with the same controller used for forward motion. These sensors enable the system to detect and follow lines on the ground, facilitating autonomous navigation in predefined paths.

Result:

The line-following system uses three IR(Infrared) sensors and a controller for forward motion, allowing autonomous navigation along the specified path by detecting and following lines on the ground.

Discussion:

- IR (Infrared) sensors detect lines and properly guide the system along the specified path.
- The controller evaluates sensor data to allow self-guided movement.
- The system may respond to different variable navigation features.

6.6 Text-to-speech

The text-to-speech technology translates written information into spoken words making it convenient for a variety of applications. The key components are a text input, a processing engine, and an audio output. The processing engine understands the text, converts it into spoken words, and delivers it through an audio output device. This technology improves accessibility and communication for various applications. Here are some pre-commands given below which robot react or respond to humans.

Q: "How are you?"

A: "I am fine, how about you"

Q: "What can you do?"

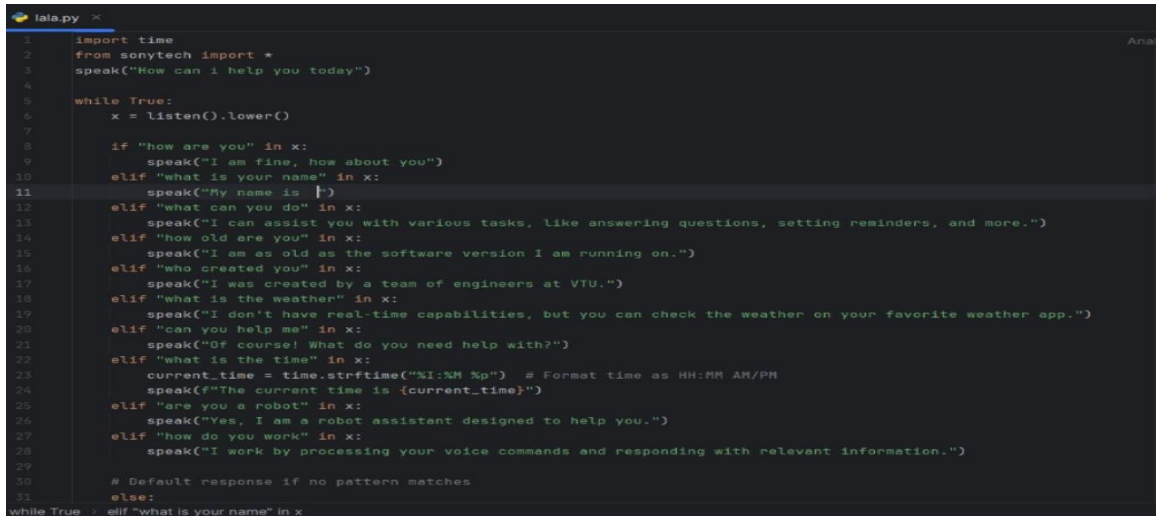
A: "I can assist you with various tasks, like answering questions, setting reminders, and more."

Q: "How old are you"

A: "I am as old as the software version I am running on."

Q: "How do you work?"

A: "I work by processing your voice commands and responding with relevant information."



```
1 import time
2 from sonytech import *
3 speak("How can i help you today")
4
5 while True:
6     x = listen().lower()
7
8     if "how are you" in x:
9         speak("I am fine, how about you")
10    elif "what is your name" in x:
11        speak("My name is lala")
12    elif "what can you do" in x:
13        speak("I can assist you with various tasks, like answering questions, setting reminders, and more.")
14    elif "how old are you" in x:
15        speak("I am as old as the software version I am running on.")
16    elif "who created you" in x:
17        speak("I was created by a team of engineers at VTU.")
18    elif "what is the weather" in x:
19        speak("I don't have real-time capabilities, but you can check the weather on your favorite weather app.")
20    elif "can you help me" in x:
21        speak("Of course! What do you need help with?")
22    elif "what is the time" in x:
23        current_time = time.strftime("%I:%M %p") # Format time as HH:MM AM/PM
24        speak(f"The current time is {current_time}")
25    elif "are you a robot" in x:
26        speak("Yes, I am a robot assistant designed to help you.")
27    elif "how do you work" in x:
28        speak("I work by processing your voice commands and responding with relevant information.")
29
30    # Default response if no pattern matches
31    else:
32        speak("I don't understand that.")
33
34 while True:
35     elif "what is your name" in x
```

Fig 6.11 Text to Speech Code

Result:

In text-to-speech technology converts written words making it a useful variety of applications. This technology serves education, entertainment, and telecommunication by providing simple and easy audio output from text information.

Discussion:

- The text-to-speech depends upon how much pre commands are uploaded to the database.
- It is flexible in programming code pre command which can be changed based on the application.
- It helps the daily routine life of humans by assisting like reminders, weather forecasting, etc.