Deakin University

Robotics Application Development

OnTrack Submission

Final Project Report HD

Submitted By:
Agrim Gautam
s221071682
2023/06/12 06:26

Tutor:
Akan Cosgun

June 12, 2023

SIT310 T1/2023 Final Project: Voice Control Commander

Agrim Gautam
S221071682
Deakin University, Australia

I. Executive Summary

This project aimed to develop a voice-controlled robot to perform Advance movements and communicate with the user. The goals were to implement a speech recognition system, enable the robot to respond to voice commands, and provide feedback to the user through speech or sound. The technical approach involved using the Vosk voice detection model [1] for offline speech recognition. The model output was saved in a text file and monitored for modifications. The text file was then processed to identify predefined commands, such as ”go,” ”stop,” ”turn right,” and ”turn left.” The system also supported more complex commands, allowing the user to specify movement distances and angles.

To convert words into numbers for specific commands, the word2num [2] Python library was used. This enabled the robot to accurately interpret commands like ”go forward 100” or ”rotate right 45 degrees.” The robot's response would be customised accordingly if values were provided with the command. Otherwise, default values were used.

The project utilised the Tello library for communicating with the drone, sending commands, and receiving replies. Additional logic was added to store and retrieve the received messages, allowing the robot to provide the user with information such as battery level and flight time. The pyttsx3 [3] library was used for text-to-speech conversion, enabling the robot to speak the replies back to the user.

The project's key findings were the successful implementation of voice recognition, accurate interpretation of commands, and effective communication between the robot and the user. The system demonstrated the ability to receive voice commands, execute movements based on those commands, and provide relevant information to the user.

II. Introduction

The project aimed to address the challenge of controlling a robot through voice commands and establishing a communication channel between the robot and the user. Voice control offers a convenient and intuitive way for users to interact with robots, and it can be particularly useful in scenarios where hands-free operation is required.

The objectives of the project were to develop a robust speech recognition system, implement basic movement commands (e.g., go, stop, turn), enable the robot to respond with speech or sound, and incorporate more advanced commands with specific values for distance and angles.

The scope of the project focused on the implementation of voice recognition and basic movement commands. The system was designed to work offline using the Vosk voice detection model. The Tello library was utilized for controlling the drone and receiving information from it.

III. Technical Approach

A. System Overview

The system architecture consisted of multiple components, including the Vosk voice detection model, a text file for storing the speech recognition output, a separate Python file for monitoring and processing the text file, and the Tello library for drone control and communication.

The flowchart of the decision mechanism involved monitoring the text file for modifications, checking the new text for predefined commands, and extracting additional details from the commands if provided. The word2num Python library was used to convert words into numbers for specific commands. The Tello library facilitated the communication with the drone, enabling command execution and message retrieval.

The hardware used in the project included a drone (Tello), a microphone for capturing voice commands, and speakers for audio output. The software components consisted of the Vosk voice detection model, the Tello library, the word2num library for command interpretation, and the pyttsx3 library for text-to-speech conversion.

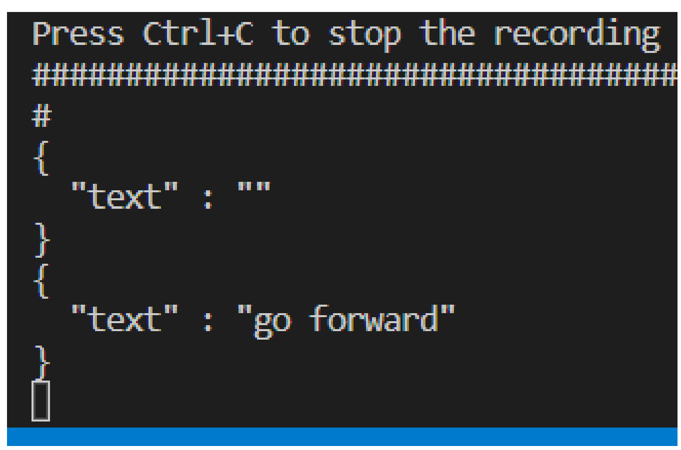

Fig. 1. Text recognised by model.

Deakin University, Australia

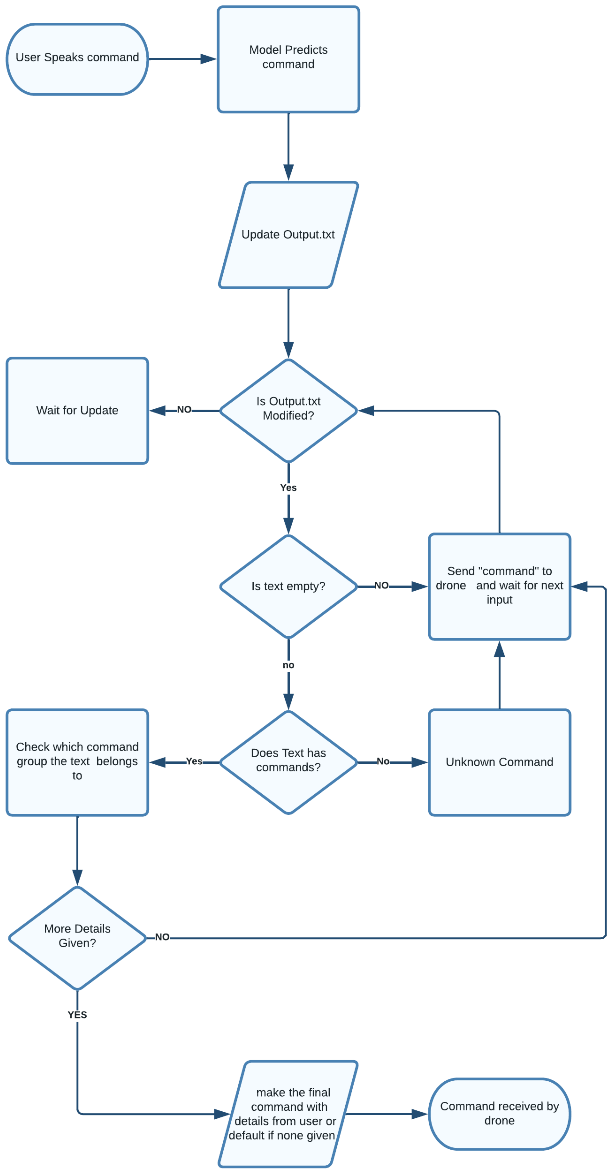

Fig. 2. flowchart of the project.

B. P-Level Feature: A fixed number of Voice Commands are recognized via Speech Recognition.

1) Approach: To implement the P-Level feature, Iused the Vosk voice detection model for offline speech recognition. The model was trained to recognize a fixed number of voice commands, such as "go," "stop," "turn right," and "turn left." The recognition model was integrated into the robotics application to convert spoken commands into text.

2) Hardware and Software :

- Hardware: - Microphone: Used to capture the user's voice commands.
- Software: - Vosk voice detection model: Used for offline speech recognition. The model was trained to recognize the fixed set of voice commands.

3) Algorithms or Models : - Vosk voice detection model: This model utilizes automatic speech recognition (ASR) algorithms to convert spoken language into text.

4) Step-by-step guide :

1) The user speaks a voice command, such as "go" or "turn right."
2) The microphone captures the user's voice.
3) The Vosk voice detection model processes the audio input and converts it into text.
4) The converted text is checked against the fixed set of recognized voice commands.
5) If a match is found, the corresponding action associated with the recognized command is executed.

P Level Feature Demo - https://youtu.be/2Bqn_rrCUiY

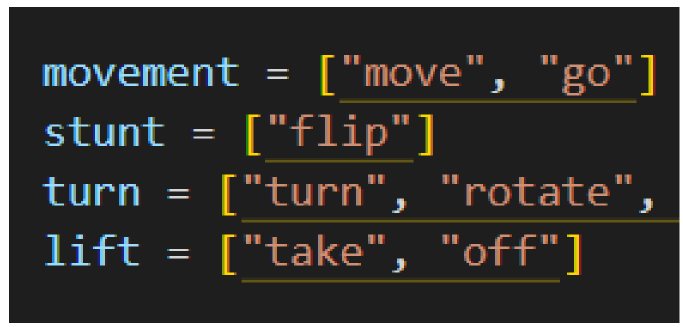

Fig. 3. Command Groups

C. C-Level Feature: Robot Moves according to simple voice commands: go/stop/turn right/turn left

1) Approach: To implement the C-Level feature, Iextended the P-Level feature by enabling the robot to move according to the recognized voice commands. When the user speaks commands like "go," "stop," "turn right," or "turn left," the robot performs the corresponding movements.

2) Hardware and Software :

- Hardware: - Robot platform (Tello Drone) : Provides the physical means for movement.
- Software: - Command execution module: Translates the recognized voice commands into specific robot movements.

3) Algorithms or Models : - Command execution algorithm: This algorithm maps the recognized voice commands to the corresponding robot movements.

4) Step-by-step guide :

1) The user speaks a voice command, such as "go" or "turn right."
2) The voice command is recognized using the Vosk voice detection model.
3) The recognized command is passed to the command execution module.
4) The command execution module maps the recognized command to the corresponding robot movement.

5) The robot performs the instructed movement, such as moving forward, stopping, turning right, or turning left.

C Level Feature Demo - https://youtu.be/4Ya7ZfpJOPI

```
sending command: command to 192.168.10.1
Done!!! sent command: command to 192.168.10.1
File Modified!
sending command: battery? to 192.168.10.1
Done!!! sent command: battery? to 192.168.10.1
File Modified!
sending command: takeoff to 192.168.10.1
Done!!! sent command: takeoff to 192.168.10.1
```
-8 CRLF {} MagicPython 3.8.10 64-bit Go Live

Fig. 4. Commands given to drone.

D. D-Level Feature: Robot Communicates back information with speech and/or buzzer noise

1) Approach: To implement the D-Level feature, Ienabled the robot to communicate information back to the user using speech or buzzer noise. The robot provides feedback in response to specific commands or events, such as reporting battery level or signaling completion of a task.

2) Hardware and Software :

- Hardware: - Speakers: Used for speech output.
- Software: - Text-to-speech module: Converts text into speech for audible output.

3) Algorithms or Models : - Text-to-speech algorithm: This algorithm converts text information into audible speech.

4) Step-by-step guide:

1) The robot receives a command or event that requires feedback.
2) The robot processes the command or event and generates the appropriate information or response.
3) If speech output is required, the text-to-speech module converts the information into audible speech and plays it through the speakers.

D Level Feature Demo -
https://www.youtube.com/watch?v=WyBMXis_vN4

E. HD-Level Feature: Robot Can handle "go forward/backward X meters" or "rotate left/right Y angles", input with speech comands

1) Approach: To implement the HD-Level feature, I extended the command recognition and execution capabilities to handle more specific movement commands. The robot can now interpret commands such as "go forward 5 meters" or "rotate left 90 degrees" and perform the corresponding movements accordingly.

2) Hardware and Software :

- Hardware: - Same as the D-Level feature.
- Software: - Same as the D-Level feature.

3) Algorithms or Models : - Same as the D-Level feature.

4) Step-by-step guide:

1) The user speaks a specific movement command, such as "go forward 5 meters" or "rotate left 90 degrees."
2) The voice command is recognized using the Vosk voice detection model.
3) The recognized command is passed to the command execution module.
4) The command execution module analyzes the command text to extract the specific distance or angle value.
5) The robot performs the instructed movement, such as moving forward/backward the specified distance or rotating left/right by the specified angle.

HD Level Feature Demo -
https://www.youtube.com/watch?v=YsD01k3uU0c

Note: The HD-Level feature required the installation of the word2num Python library to accurately convert words into numeric values for the specific command details, such as distances and angles. The command execution module customized the robot's response based on the extracted values, or used default values if no specific details were provided in the command.

```python
# Check if the word in stunt group is in the text
elif any([x in txt for x in stunt]):
    if "left" in txt:
        tello.send_command("flip l")
    elif "right" in txt:
        tello.send_command("flip r")
    elif "forward" in txt:
        tello.send_command("flip f")
    elif "back" in txt:
        tello.send_command("flip b")
    else:
        tello.send_command("flip f")
```

Fig. 5. Algorithm in action.

IV. Discussion

The results of the project demonstrated the successful implementation of voice recognition and control for a robot. The system was able to accurately interpret predefined commands and execute the corresponding movements. The integration of the Tello library allowed for communication between the robot and the user, providing relevant information on battery level and flight time.

Some unexpected challenges were encountered during the project, including the need to handle different variations of commands and account for potential errors in speech recognition. However, these challenges were overcome through careful design and implementation.

Limitations of the project include the reliance on offline speech recognition, which may not be as accurate as cloud-based solutions. The system's vocabulary was also limited to the predefined commands, and it lacked the ability to handle more complex natural language interactions. Additionally, the project focused on basic movement commands and did not incorporate advanced perception or navigation capabilities.

Future improvements could involve integrating more advanced speech recognition models to enhance accuracy, expanding the vocabulary and supporting natural language understanding, and incorporating additional sensors for improved perception and navigation. Moreover, the project could benefit from exploring cloud-based solutions for real-time speech recognition and leveraging advanced machine learning techniques for more robust command interpretation.

V. My Project Journey

Throughout the project, I gained valuable experience in implementing speech recognition and control for a robot. I became proficient in using the Vosk voice detection model and understanding its output in JSON format. The integration of the Tello library allowed me to establish communication between the robot and the drone, enabling command execution and information retrieval.

There were many thrilling moments during the project, such as the "eureka" moment when I successfully integrated the speech recognition system and saw the robot respond to voice commands. Fixing bugs and troubleshooting issues also provided a sense of accomplishment and growth as a developer.

This project has ignited my passion for robotics and related fields. It has shown me the potential of voice control as an intuitive and convenient interaction method for robots. I am excited to continue exploring and expanding my knowledge in robotics, machine learning, and human-robot interaction.

Project Github -
https://github.com/cyber-panther/Voice-Cotrol-Drone-SIT310

VI. Future Work

A. Proposal for a Dream Robotics Project:

The aim of this project is to develop an advanced robot capable of interacting in an urban environment using a combination of hardware and software technologies. The robot should incorporate computer vision capabilities for a yet-to-be-decided purpose and have the ability to recognize speech commands and provide appropriate outputs.

B. Background and Existing Research:

Urban environments present unique challenges for robots, requiring them to navigate complex terrains, interact with objects and people, and perform various tasks. Existing research in robotics has focused on addressing specific aspects of urban robotics, such as navigation algorithms, object recognition, and human-robot interaction. However, there is still a need for a versatile robot that combines these capabilities into a single system.

C. Objectives and Scope:

The objectives of the proposed project include:

1) Designing and building a robot capable of maneuvering in an urban environment.
2) Implementing computer vision algorithms for object recognition and scene understanding.
3) Developing speech recognition and synthesis capabilities for human-robot interaction.
4) Integrating these components to enable the robot to perform complex tasks based on speech commands and visual inputs.

D. The scope of the project includes:

1) Designing a robust robot platform capable of traversing various urban terrains.
2) Implementing computer vision algorithms for object detection, tracking, and recognition.
3) Developing a speech recognition system to understand spoken commands.
4) Integrating a text-to-speech module to provide audible outputs.
5) Creating an intuitive user interface for command input and receiving feedback from the robot.
6) Testing and refining the system to ensure reliable performance in urban environments.

E. Hardware and Software:

1) Hardware:
   - Robot platform: A customizable and agile robot platform with sensors for navigation and obstacle avoidance.
   - Camera module: High-resolution camera(s) for computer vision tasks.
   - Microphones and speakers: To enable speech recognition and synthesis.
   - Onboard computer: Powerful processing unit for running algorithms and controlling robot movements.
2) Software:
   - Computer vision libraries: OpenCV or similar libraries for object detection and recognition.
   - Speech recognition libraries: CMU Sphinx or Google Cloud Speech-to-Text API for speech recognition.
   - Text-to-speech libraries: eSpeak or Google Text-to-Speech API for speech synthesis.
   - Programming languages: Python or C++ for system integration and control.

Thinking outside the box, additional hardware and software components can be considered based on specific project requirements and constraints. For example, if the computer vision application involves 3D mapping or augmented reality, depth sensors (e.g., LiDAR or depth cameras) can be incorporated. Similarly, machine learning frameworks like TensorFlow or PyTorch can be utilized for advanced object recognition and scene understanding.

By combining the skills and knowledge gained from previous projects, the proposed robot will have the potential to revolutionize urban robotics, enabling enhanced interaction and task execution in urban environments.

Note: The project Idea given here emerged from my discussion with ChatGPT [4] about my previous projects and how they can all come together to make a big dream project and give credit where credit's due, ChatGPT made its citation itself.

References

[1] alphacephei, "Vosk offline speech recognition api," 2023, last accessed 11 June 2023. [Online]. Available: https://alphacephei.com/vosk/

[2] word2number, 2017, last accessed 11 June 2023. [Online]. Available: https://pypi.org/project/word2number/

[3] pyttsx3, 2020, last accessed 11 June 2023. [Online]. Available: https://pypi.org/project/pyttsx3/

[4] OpenAI, "Chatgpt," https://openai.com/research/chatgpt, 2021, accessed: June 11, 2023.