DESIGN AND DEVELOPMENT OF A CONTROL INTERFACE FOR A VOICE-OPERATED DRONE

D.M.L.A.B. Dissanayaka

2024

Design and Development of a Control Interface for a Voice-Operated Drone

A Thesis Submitted for the Degree of Master of Computer Science

D.M.L.A.B. DISSANAYAKA

University of Colombo School of Computing

2024

DECLARATION

Name of the student: DMLAB Dissanayaka
Registration number: 2018/MCS/016
Name of the Degree Program: Master in Computer Science
Project/Thesis title: Design and Development of a Control Interface for a Voice-Operated Drone

1. The project/thesis is my original work and has not been submitted previously for a degree at this or any other University/Institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.
2. I understand what plagiarism is, the various types of plagiarism, how to avoid it, what my resources are, who can help me if I am unsure about a research or plagiarism issue, as well as what the consequences are at University of Colombo School of Computing (UCSC) for plagiarism.
3. I understand that ignorance is not an excuse for plagiarism and that I am responsible for clarifying, asking questions and utilizing all available resources in order to educate myself and prevent myself from plagiarizing.
4. I am also aware of the dangers of using online plagiarism checkers and sites that offer essays for sale. I understand that if I use these resources, I am solely responsible for the consequences of my actions.
5. I assure that any work I submit with my name on it will reflect my own ideas and effort. I will properly cite all material that is not my own.
6. I understand that there is no acceptable excuse for committing plagiarism and that doing so is a violation of the Student Code of Conduct.

Signature of the Student	**Date (DD/MM/YYYY)**
	2024/09/14

Certified by Supervisor(s)

This is to certify that this project/thesis is based on the work of the above-mentioned student under my/our supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

	Supervisor 1	**Supervisor 2**	**Supervisor 3**
Name	Dr K D Sandaruwan		
Signature			
Date	2024.09.15		

ACKNOWLEDGEMENTS

This journey towards exploring and understanding the complexities of voice-operated drone technology has been both challenging and immensely rewarding. As I stand at the culmination of this academic endeavor, I am filled with gratitude for those who have supported, guided, and inspired me along the way.

First and foremost, I extend my deepest thanks to my thesis advisor, Dr. KD Sandaruwan, whose expertise, understanding, and patience added considerably to my graduate experience. Your guidance was invaluable not just academically but also in teaching me the critical thinking and problem-solving skills necessary to navigate the complexities of this project.

I cannot express enough thanks to my family and friends for their understanding and endless love through the duration of my studies. Your unwavering belief in my abilities gave me the strength and confidence to persevere through challenges.

This thesis stands as a milestone in my academic journey, and it would not have been achievable without the collective support and encouragement from everyone mentioned and many others unmentioned. I am profoundly grateful for each contribution to my journey.

ABSTRACT

The use of drones, or Unmanned Aerial Vehicles (UAVs), is growing in many areas such as agriculture, surveillance, delivery, and entertainment. However, controlling drones often requires special training, which makes them difficult to use for people without technical skills. This thesis aims to solve this problem by creating a new control system that allows users to operate drones using voice commands.

The main idea is to make drone operation easier and more natural by allowing users to simply speak commands to control the drone. This system uses advanced speech recognition technology from Google to understand and convert voice commands into drone actions. The DJI Tello Edu drone was chosen for this research because it is small, programmable, and suitable for indoor use. The design included creating a list of voice commands and matching them to specific drone movements.

To test the system, various experiments were conducted to measure its performance and gather user feedback. The results showed that the voice command system had a high accuracy rate of 99.6% in quiet environments. Even in noisy or windy conditions, the system performed well, though with slightly lower accuracy. Users found the system easy to use and appreciated the simplicity of controlling the drone with their voice. They also noted that such a system could make drones more useful for everyday tasks, increasing efficiency in areas like inspections and emergency responses.

In conclusion, this research makes drones more accessible by allowing them to be controlled through voice commands. This new method simplifies the use of drones and could lead to their wider adoption in many fields. The voice-operated control system not only makes drone flying easier but also opens up new possibilities for how drones can be used in daily life and work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

UAV	Unmanned Arial Vehicle
SDK	software development kit
DHCP	Dynamic Host Configuration Protocol
UDP	User Datagram Protocol
VPS	Vison Positioning System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
UI	User Interface
IDE	Integrated Development Environment
API	Application Programming Interface
TOF	Time of Flight
CSV	Comma Separated Values
RPC	Remote Procedure Call

CHAPTER 1

INTRODUCTION

In recent years, the integration of Unmanned Arial Vehicles (UAVs) into various industries has brought significant advancements in technology and has revolutionized the way tasks are performed in sectors such as agriculture, surveillance, delivery and entertainment (Kardasz and Doskocz, 2016). The effective operation of these aerial systems often demands substantial expertise and experience, restricting their accessibility to a limited pool of skilled operators. Consequently, the demand for more intuitive and user-friendly control interfaces for drones that can imitate the proficiency of experienced drone pilots are increasing. It has led to the exploration of alternative methods for their operation, with voice-operated control systems emerging as a promising solution. This research focuses on the design and development of a sophisticated control interface for a voice-operated drone, aimed at replicating the decision-making and maneuvering capabilities of seasoned human pilots in compact and domestic environments.

1.1 Motivation

The motivation behind this research stems from the need to enhance the accessibility and usability of drones in various applications. One of the exiting and critical usage of UAVs is to reach and visualize the places that people cannot access easily during a disaster management. Such as collapsed buildings in earthquakes, blocked tunnel, fire rescues, in case of terrorist attacks and etc. Nowadays experts use remotely controlled ground robots to reach and access these situations (Wu, 2023). But if it can use a small drone with minimal sensors, it can gather a lot of valuable information about the situation in a very small time.

Traditional control interfaces for drones often require specialized training and complex manual operations, which can limit their usability among non-expert users. To use a UAV in mentioned compact and hazarded environments with limited human resources it needs to have a simple method of controlling. Using voice is the most common and familiar way of communicating among humans. By developing a voice-operated control interface, it aims to simplify the drone operation process, making it more accessible to a wider range of users, including those with limited technical expertise. This technology has the potential to broaden the scope of drone applications and facilitate its integration into everyday tasks, leading to increased efficiency and productivity in diverse fields.

1.2 Statement of the problem

Despite the advancements in drone technology, the usability of drones remains limited due to the complexities associated with traditional control interfaces. These interfaces often require specialized training and manual operations, posing significant barriers for non-expert users who may benefit from utilizing drones for various applications. Furthermore, the existing control interfaces lack the intuitive interaction and user-friendly experience necessary for efficient and seamless drone operation. Consequently, there is a pressing need to develop a more accessible and user-friendly control interface that enables individuals with minimal technical expertise to operate drones effectively. This problem highlights the necessity of exploring alternative control methods, such as voice-operated systems, to enhance the usability and accessibility of drones across diverse applications. This research aims to address these challenges by designing and developing a control interface that leverages voice recognition technology to enable intuitive and efficient drone operation, catering to the needs of both expert and non-expert users.

1.3 Research Aims and Objectives

1.3.1 Aim

The primary aim of this research is to design and develop a control interface for a voice-operated drone that emulates the decision-making capabilities and flight behavior of an experienced drone pilot in a domestic environment. By leveraging advanced algorithms and machine learning capabilities, the interface will enable users to intuitively interact with the drone, replicating the exact control and precision typically associated with skilled human operators. This research aims to contribute to the democratization of drone technology by creating a user-friendly interface that empowers a broader demographic to leverage the full potential of UAVs in diverse applications.

1.3.2 Objectives

1. Identify a set of voice commands that are generic to handling a drone in domestic environment.
2. To establish a clear mapping between specific voice commands and the corresponding drone control actions.
3. To investigate the key factors that affect the mapping between voice commands and drone actions, such as the voice recognition characteristics, the drone's capabilities and limitations, and the environmental conditions.
4. To gather drone command sequences from a hand-held controller.

5. Study of sharing the expert drone pilot control skills with nowise pilot using voice commands.
6. To evaluate the performance and usability of voice-controlled drones in comparison to other control methods, such as remote controls or mobile apps, in terms of accuracy, reliability, safety, efficiency, and user satisfaction.

1.4 Scope

The proposed research aims to investigate the mapping between voice commands and drone actions, with the objective of developing and optimizing techniques for voice control of drones. The research will focus on the following aspects:

1. Voice commands: The research will identify and define a domain specific set of voice commands to interpret as drone control commands.
2. Control commands: The research will use expert drone pilots to control the drone according to voice command and then gather control commands with the respective properties.
3. Mapping techniques: The research will explore and compare different techniques for establishing and optimizing the mapping between voice commands and drone actions. The research will investigate the advantages and limitations of each technique and propose refinements and modifications to improve their effectiveness and efficiency.
4. Performance evaluation: The research will evaluate the performance and usability of the voice-controlled drone system in comparison to other control methods, such as remote controls or mobile apps, in terms of accuracy, reliability, safety, efficiency, and user satisfaction. The research will use both objective and subjective measures to assess the performance and usability of the system, such as task completion time, error rate, and user feedback.

1.5 Structure of the Thesis

This thesis is structured into several chapters that collectively provide a comprehensive overview of the design and development process of the voice-operated drone control interface. Chapter 2 reviews the existing literature and research related to drone control interfaces and voice recognition technology, highlighting the current state of the field and identifying the gaps that this research aims to address. Chapter 3 outlines the methodology employed in the design and development of the control interface, including the selection of hardware and software components, as well as the implementation approach. Also, the detailed design and architecture of the control interface, discussing the key components and their functionalities. Chapter 4 covers the implementation and testing phase, providing insights into the performance evaluation and user experience assessment. And also, discusses the results obtained from the testing phase and analyzes the implications of the findings. Finally, Chapter 5 concludes the thesis by summarizing the key contributions, discussing the limitations of the research, and providing recommendations for future work in this domain.

CHAPTER 2

LITERATURE REVIEW

The integration of Unmanned Aerial Vehicles (UAVs), commonly known as drones, into various sectors has been transformative. Traditionally, the control interfaces for these UAVs have relied heavily on radio controllers, which, despite their effectiveness, pose significant challenges due to their complexity and the expertise required for their operation. As technology advanced, the search for more intuitive and user-friendly control methods escalated, leading to the exploration of voice-operated controls. This chapter reviews the evolution of drone control interfaces, the challenges associated with traditional methods, and the emerging research in voice-operated drone control systems.

Radio controllers have been the standard method for operating drones. These devices use radio waves to send commands from the operator to the drone, allowing for precise control over its movements. Despite their widespread use, radio controllers are often complex, requiring significant skill and practice to master. The complicated nature of these controllers limits their usability to experienced operators, creating a barrier for novice users (Anand et al., 2018). The complexity of radio controllers stems from the need to manually adjust various parameters such as throttle, pitch, yaw, and roll, often simultaneously. This multi-dimensional control system, while providing high precision, can be overwhelming for new users. The steep learning curve associated with radio controllers can discourage potential users, thereby limiting the broader adoption of drone technology in everyday applications (Contreras et al., 2020a).

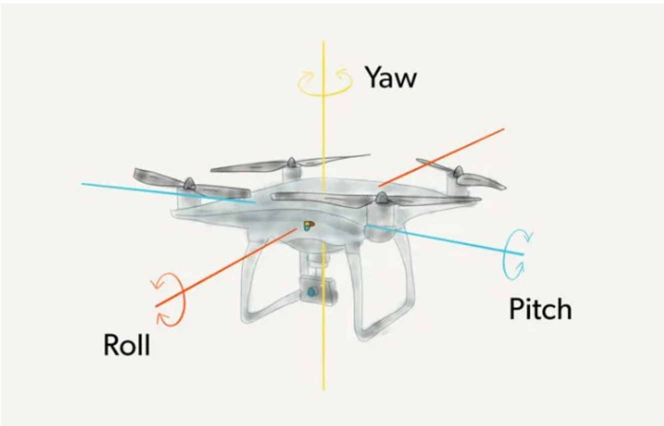


Figure 1. Pitch Roll and Yaw

First modern proposal for UAVs were made to the British Air Ministry by Archibald Low in 1961 to develop unmanned defense air craft to use against German airships (Contreras et al., 2020a, p. 3). Form Low's development of first radio-controlled UAV for military; the industry has improved in multiple aspects. Size of UAVs has been developed from range of fixed wing multi-Ton aircrafts to small hand-held quad copters. Nowadays, UAVs are not only controlled by professional UAV pilots, but also users with less aviation experience (Anand and Mathiyazaghan, 2016). These users need much more simpler and natural control interfaces to maneuver the UAVs. Most natural way for humans to control is using voice commands. Hence, voice control of drones is a rapidly developing field with numerous studies and research papers exploring the challenges and opportunities of this novel interaction paradigm (2.2 Contreras et al., 2020b, p. 5). One of the primary issues is the mapping between voice commands and drone actions, which requires establishing a clear and intuitive correspondence between the user's spoken words and the specific actions that need to be performed by the drone (Suarez Fernandez et al., 2016).

(Anand and Mathiyazaghan, 2016) used phenome matching in order to interpret voice commands in to control commands. They have broken down each voice command in to small segments. Then they matched each segment with 45 phenomes in English language. They had predefined 10 voice commands (Table 1. Set of commands by research) with their phenome collections. Highest probability of phenome matching has been the match for the voice command. Subjected quad copter is also developed in this study. They defined ten collections of drone control actions for each voice command. Intention of the project was to control the drone entirely by voice commands and they have succeeded.

(Kumaar et al., 2018) Study used eight control commands (Table 1) and used a small Deep Neural Network (DNN) to interpret voice commands to control commands. Since there is limited set of commands; study concluded with that a drone does not need an extensive speech recognition system to identify voice commands because of the limited command set. Their trained model size was less than 4 Mega Bytes in size. One off the shelf quadcopter and custom build one used for evaluation. First one used Python API to send controller commands to the drone after determining commands from outside hardware. And Embedded system took in place to run voice recognition algorithm as a part of custom-made quadcopter.

According to the command combination of The study (Menshchikov et al., 2019) discuss about how to implement term Body Machine Interface (BMI) with controlling a UAV. Mainly with voice commands and body gestures together. With the limited set of five voice commands, they managed to implement embedded AI system to control drone. But in evaluation it revealed the latency of decision making is greater in embedded system than the use general purpose computer.

(Contreras et al., 2020b) Implemented an algorithm for command interpretation using both cloud-based speech recognition system and phenome matching in the same system. They have built a domain-based dictionary to do phenome matching. Once they get the text from voice to text cloud service then, the text is used as a hypothesis to the phenome matching using Levenshtein distance. Study concludes with that the phenome matching improved the level of accuracy in instruction recognition. They used Spanish native speakers in the study and they got better results with Spanish voice commands. They used only nine control commands with predefined control actions. It proposes to use more languages for voice recognition.

Table 1. Set of commands by research

Research Reference	Input	Set of commands
(Anand et al., 2018)	Voice	Start Motors, Fly low, Fly High, Go Left, Go Right, Go Forward, Go Backward, Turn Left, Turn Right, Hover
(Kumaar et al., 2018)	Voice	Takeoff, Land, Forward, Backward, Left, Right, Up & Down
(Menshchikov et al., 2019)	Gesture	Rolling, Pitching, Yawing
(Chandarana et al., 2017)	Voice	Forward, Backward, Right, Left, Up, Down, Circle, Spiral
(Contreras et al., 2020a)	Voice	Up, Down, Go Right, Go Left, Go Forward, Go Back, Turn Left, Turn Right, Stop
(Jones et al., 2010)	Voice	Left, Right, Forward, Come back to me

Set of commands selected in each research in Table 1 helps to decide the common set of commands that needs to be used in voice operated drone system as in Table 4.

(Siegert et al., 2020) compared some leading speech recognition cloud-based solutions at the time and published results with different aspects. There are plenty of studies used different voice recognition mechanisms and they found their technique is feasible for respective studies. Hence this research is going to use one of the existing services for convert speech to text.

(Thu and Gavrilov, 2017) states that there are four parameters related to drone flight control. Roll rate, Pitch rate, Yaw rate and thrust. Study provides a systematic design and modelling process for realistic flight control applications using mentioned for parameters.

(Kim et al., 2021) Used manufacturer provided library for send control signals to the drone using different set of parameters. Which explains that the manufacturers of drones make their own ways of exposing the controlling interface of a drone. Drone Simulation kits provide user friendly commands like ‘land’ in order to make controlling more user friendly(“Welcome to AirSim,” 2023). This study is going to select a quadcopter with having one of the existing controller API to command the drone.

However, there are challenges associated with voice-operated systems. The accuracy of speech recognition can be affected by background noise, accents, and speech impediments (Chandarana et al., 2017). Furthermore, the implementation of voice control requires sophisticated software capable of understanding and processing natural language commands accurately and efficiently (Basak et al., 2022).

Despite these challenges, voice control of drones has several potential benefits, such as increased accessibility, safety, and efficiency. For example, voice control can enable people with physical disabilities or limited mobility to operate drones more easily and independently. Additionally, voice control can reduce the cognitive load and distraction associated with other control methods, such as remote controls or mobile apps (Contreras et al., 2020b). The primary advantage of voice-operated drone control is its intuitiveness. By leveraging natural language, these systems can make drone operation more accessible to a wider audience. Additionally, voice control can enhance multitasking capabilities, allowing operators to focus on other tasks while controlling the drone verbally.

The transition from traditional radio controllers to more intuitive control methods, such as voice-operated systems, represents a significant advancement in drone technology. While radio controllers provide precise control, their complexity limits their accessibility. Voice-operated control systems offer a promising alternative, simplifying the user interface and making drone operation more intuitive. Ongoing research continues to address the challenges associated with speech recognition and natural language processing, paving the way for broader adoption of voice-controlled drones in various fields.

CHAPTER 3

METHODOLOGY

This research will follow a mixed-methods approach, combining quantitative and qualitative data collection and analysis methods to investigate the mapping between voice commands and drone actions. The methodology will involve the following steps:

3.1 Selecting the drone

Research focused on indoor environments with small maneuvering spaces. Selected drone needs to be small in size and has to be capable of making small adjustments on the flight. It should be capable of communicate with the control center easily and reliably. By the nature, it has to be a programmable drone. And has to be available and affordable in the research conducting country.

Figure 2. Tello Edu drone

After evaluating mentioned factors, it was decided to continue the research with DJI Tello Edu quadcopter. It is small in size; width, height, depths are 98, 41 and 92.5 milli meters in order. Less in weight as 87 grams. Used Tello drones were available in an affordable price range at the start of this Research. Tello EDU can use WIFI or Bluetooth to communicate with ground control. Nowadays, both communication interfaces are built in both mobile and laptop computers by default. This drone was built for support education purposes. It comes with a build in SDK to support commanding and retrieve diagnostic sensor data.

Tello Edu uses WIFI as the network interface and User Datagram Protocol (UDP) as the communication protocol when it is using Software Development Kit (SDK). Drone itself contains the Dynamic Host Configuration Protocol (DHCP) server and it can manage multiple devices connected at the same time. That feature allows to use one device to control the drone and another device to collect diagnostic / sensor data.

3.2 Speech recognition

Automatic Speech Recognition (ASR) has been evolved over decades. Almost all of the modern-day computers around human supports speech recognition. There are multiple vendors who provide cloud-based text to speech services such as Google, Amazon, IBM, Microsoft etc.

Google Speech to Text service is the most familiar service to the subjected novice users voice characteristics in this research. Subjected users live in a demographic area with a lot of Android based mobile devices are popular and voice commanding, voice typing is popular at the time. Because of this popularity, Google speech-to-text service has improved a lot with its usage. It is capable of recognizing nuances in speech, such as accents and dialects and speech patterns, which significantly reduces errors and improves the quality of transcriptions. It offers both real-time and streaming speech recognition capabilities. This means that it can transcribe speech as it happens, which is particularly useful for applications such as immediate feedback system as in this research. The service is designed to perform well in noisy environments. It can filter out background noise and focus on the speech, which is crucial for applications in public spaces, outdoor environments, or in situations with background chatter. Pricing model is very cost-effective and it really helped to minimize the cost to zero for the speech-to-text service in the time of this research. With this accuracy, reliability, cost-effectiveness and high availability, out of above-mentioned vendors, it was decided to use Google speech-to-text service in this research to transcribe voice commands into text.

Subjected novice users speak Sinhala and Tamil as their mother tongue. But everybody is capable of speak in English language with the vocabulary that needs for this voice operation commands. And English language models are more mature than the Sinhala or Tamil models in the service because of the more usage. English selected as the language for voice commands because of these reasons.

3.3 Voice commands

The research will identify and define a specific set of voice commands to interpret as drone control commands within the study.

Tello drone has basic eight commands that can be moved across three axes x, y and z. And it can be rotated on z axis. Each of these four movements can have positive or negative values.

Table 2. Tello EDU Basic movements represents the four positive and four negative movement names. Voice commands needs to support these basic eight commands in any drone movement.

Table 2. Tello EDU Basic movements

Axis	Action	Positive movement name	Negative movement name
x	Move / Pitch	Forward	Backward
y	Move / Roll	Left	Right
z	Move / Throttle	Up	Down
z	Rotate / Yaw	Clockwise / Left	Counter clockwise / Right

Axes can be defined by the right-hand rule. As illustrated in Figure 3. Right hand rule for Tello Edu 3D coordinates system pointer finger will direct to positive x, middle finger directs to positive y and thumb directs to positive z.

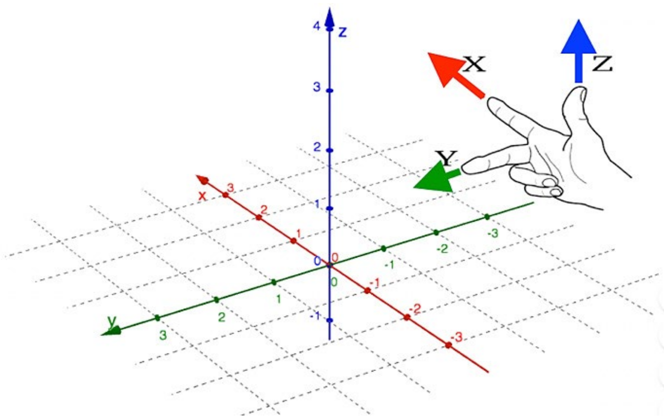


Figure 3. Right hand rule for Tello Edu 3D coordinates system

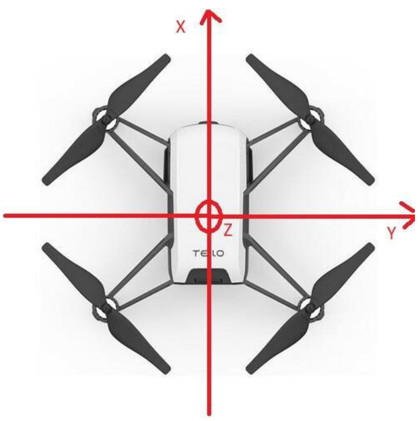


Figure 4. Tello EDU 3D coordinate axes

Apart from the above commands, SDK have a set of commands that are required for the operations mentioned in Table 3.

Table 3 Tello Edu built in SDK commands

SDK command	Description
command	Drone will enter to SDK mode.
takeoff	Auto Takeoff.
land	Auto Land
emergency	Stops motors immediately

"command" is the command to turn on the SDK mode that prepare the drone to accept inputs as UDP packets. "takeoff" and "land" commands use Vison Positioning System (VPS) to properly position in mid-air after take-off and safe landing by adjusting for the uneven or bad landing surfaces. "emergency" is the command to stop the motors immediately in any time of the flight. That command is required in these operations avoid collisions and guarantee the safety of all who participates in the flight sessions. It needs a command to stop all the movements and hold the position in the midair as well. If drone is about to collide or executing wrong movement then, it can use this command to stop movements and hover on the exact position at the time.

Table 4 represent the set of natural commands derived from above mentioned use cases. Research use these thirteen commands as the base natural commands.

Table 4. Natural command set

Natural command	Axis	Polarity	Description
Go Forward	x	Positive	Move to positive direction of x axis
Go Back	x	Negative	Move to negative direction of y axis
Go Left	y	Positive	Move to positive side of y axis
Go Right	y	Negative	Move to negative side of y axis
Go Up	z	Positive	Move to positive side of z axis
Go Down	z	Negative	Move to negative side of z axis
Turn Left	z	-	Rotate clockwise on z axis
Turn Right	z	-	Rotate counter clockwise on z axis
Connect	-	-	Issue "command" command to enter SDK mode
Takeoff	-	-	Take off the drone
Land	-	-	Land the drone
Emergency	-	-	Turn off motors
Hover	x, y, z	-	Stop all movement and hold the position in mid air

When it comes to the actions on and axis, it always expects the amount to be moved. This amount can be and exact value measure by a standard unit of measure or, it can be an amount decided by the drone pilot. Experienced pilot can decide what should be the normal, small or big amounts of movements for the given context by their experience. Hence, the research let the pilot to decide the movements under three natural amount commands as mentioned in Table 5. Natural commands for Amounts.

Table 5. Natural commands for Amounts

Natural command for Amount	Description
Normal	Normal amount of movement or rotation. This is the default amount if amount is not mentioned.
Little	Smaller amount of movement or rotation.
Lot	Larger amount of movement or rotation.

$$Total\ natural\ commands = (Movement\ commands \times Amount\ commands) + SDK\ commands + Hover \quad (1)$$

Total natural commands = (8 * 3) + 4 + 1

Total natural commands = 29

Research considers all 29 natural commands mentioned above to be executed using voice commands.

3.4 Expert Pilot Data collection

An expert drone pilot control the drone according to each selected voice command and the log files will be collected for each session. Tello Edu equipped with Inertial Measurement Unit (IMU). It is a combination of inertial sensors such as gyroscope and accelerometers. IMU can provide the data about velocity and position. This drone does not have Global Positioning System (GPS) sensor capabilities. It works only according to the local coordinates system of the drone. External program is needed to convert commands into global coordinates system compatible commands using the state of the drone using a feedback loop. It was decided to not to do that in this research due to the complexity.

There are few opensource Android applications already built to utilize the Tello Edu SDK and control the drone by providing virtual touch-based joysticks as in the real remote controllers. (“jithin8mathew/Tello_object_detection_demo_application: Building an Android Application to control Tello Drone and perform real-time object detection using YOLOv5,” n.d.) is one of the open-source libraries created with the capability to control Tello Edu using the SDK. It has been cleaned up and modified to use in the purpose of this research (Dissanayaka, 2024).

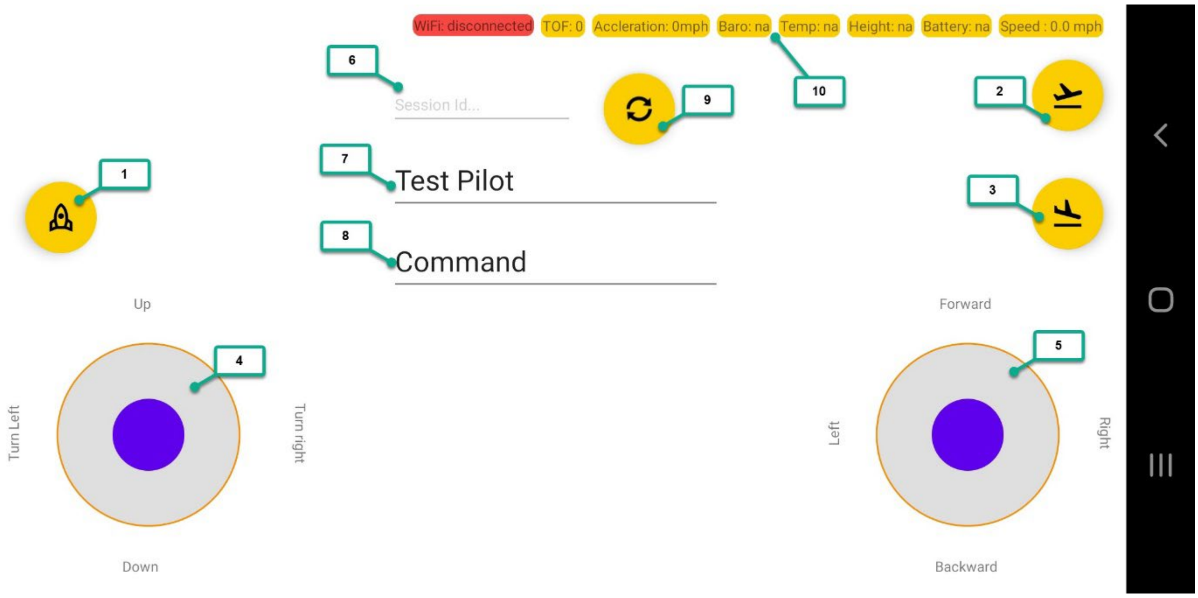


Figure 5. Android app to log pilot sessions

User Interface (UI) modified as shown in Figure 5. Android app to log pilot sessions. Table 6 explains the UI of the Android ("Android | Do More With Google on Android Phones & Devices," n.d.) application. Android Studio ("Download Android Studio & App Tools - Android Developers," n.d.) has been used as the IDE and Java ("Java | Oracle," n.d.) used as the programming language for this application development.

Table 6. Android Application UI components

#	**UI component**	**Description**
1	Connect Button	Enable SDK mode in drone
2	Take off	Takeoff drone
3	Land	Land the drone
4	Left joystick	Control Throttle (UP/Down) and Yaw (Rotate left right)
5	Right joystick	Control Pitch (Forward/Back) and Roll (Left/Right)
6	Session Id text area	Display the active session Id for the log file
7	Pilot text area	Name of the pilot to create the log file
8	Command text area	Current running command for this log file
9	Refresh Button	Generate new session Id
10	Sensor data row	Received sensor data from drone

Once the Android device connected to the Tello Drone WIFI, then it can trigger connect button to send "command" command and turn the SDK mode on in the Tello drone. Immediately the drone will start to send diagnostic data to Android device. Some of these data will be displayed in the sensor data row.
Log file will create as soon as it clicks the takeoff button. Pilot can update the pilot's name and command name before takeoff. Session id is generated using date and time. Log file is a text file with single row for each entry and comma delimited values for one row. File name contains the information to identify the pilot, command and session time.

Ex. log file name format: *log_{Pilot Name}_{Command}_{Session Id}.txt*
Ex. log file name: *log_Test_Pilot_Command_0304_103939.txt*

Android application use "rc" command in Tello SDK to control the drone using virtual joysticks. This command needs four parameters to be sent with it. Roll, pitch, throttle and yaw in order. Values ranges are from -100 to +100 for all these four parameters.

Ex. command format: *rc {roll} {pitch} {throttle} {yaw}*
Ex. command: *rc 0 0 0 0*

These parameters meant to be the lever force in a physical remote controller joystick. Virtual joystick control created to simulate this force value using two circles. One is big and anchored in UI. Other one is smaller and can move inside the bigger circle. One joystick has four controls. Each control gets 90º degree angle space from the joystick moving circle. Strength is calculated by the distance percentage from the center of the circle with compared to the radius of the outer circle as illustrated in Figure 6. All eight Joystick commands placed in similar manner to the DJI Tello Edu stock Android application ("Tello App - Download Center - DJI," n.d.).

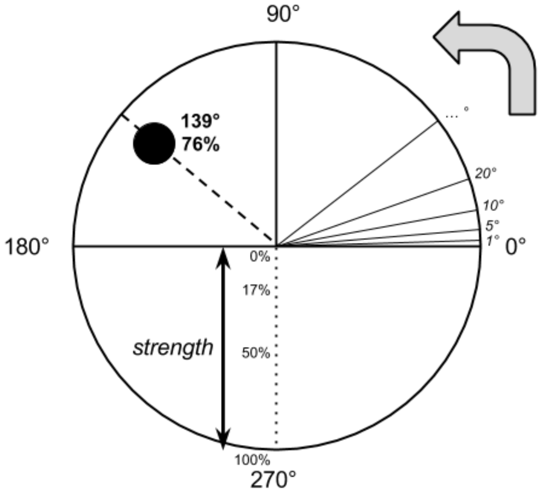
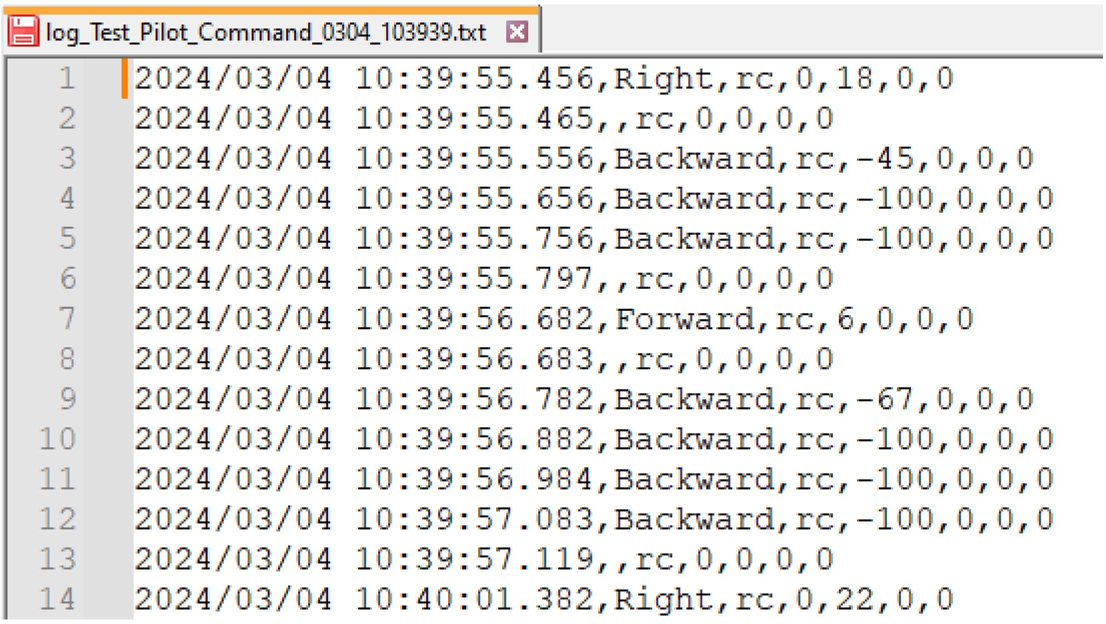


Figure 6. Virtual Joystick measurements

With the movement of the joystick, it will feed the running "rc" command in to the log file in frequency of 10Hz. Each log entry will contain the timestamp, joystick action, command name, roll, pitch, throttle and yaw values as a comma delimited string. This log file will be used to generate the command sequence for all the movement command variations.

Ex. log row format: *{time},{joystick Action},{command},{roll},{pitch},{throttle},{yaw}*
Ex. log row: *2024/03/04 10:39:55.556,Backward,rc,-45,0,0,0*



log_Test_Pilot_Command_0304_103939.txt

```
2024/03/04 10:39:55.456,Right,rc,0,18,0,0
2024/03/04 10:39:55.465,,rc,0,0,0,0
2024/03/04 10:39:55.556,Backward,rc,-45,0,0,0
2024/03/04 10:39:55.656,Backward,rc,-100,0,0,0
2024/03/04 10:39:55.756,Backward,rc,-100,0,0,0
2024/03/04 10:39:55.797,,rc,0,0,0,0
2024/03/04 10:39:56.682,Forward,rc,6,0,0,0
2024/03/04 10:39:56.683,,rc,0,0,0,0
2024/03/04 10:39:56.782,Backward,rc,-67,0,0,0
2024/03/04 10:39:56.882,Backward,rc,-100,0,0,0
2024/03/04 10:39:56.984,Backward,rc,-100,0,0,0
2024/03/04 10:39:57.083,Backward,rc,-100,0,0,0
2024/03/04 10:39:57.119,,rc,0,0,0,0
2024/03/04 10:40:01.382,Right,rc,0,22,0,0
```

Figure 7. Android application command data log file

3.5 Library of voice command to drone commands mappings

Data logs collected from the Android app is used to generate the collection of drone commands that can be mapped to each voice command. Experienced pilot drives multiple trials for the same command. For each of these trials, the duration can be varied and the commands for respective time intervals can be vary. It needs to average all the command time values series in to one time value series. All the data needs to be adjusted to same duration scale. Scale has been derived by average time duration of same command executions. Following formula has been used to scale the time.

$$New\ Time = Original\ Time \times \frac{New\ Duration}{original\ Duration} \tag{2}$$

All experienced flight logs have been collected in to one Excel spreadsheet as in Figure 21. Collection of all experienced flight logs. Then data was cleaned to remove pauses in between and other commands such as takeoff and landing. Then all data was labeled using a 'session_id' to separate each execution command sets. Column 'timeline' added with calculating the accumulated time from the start to end of each session.

It took the average of actual time duration of each type of command session and round it to the nearest 100th millisecond to get the new duration value. Then it created the 'time_scaled' column using new time formula as in Figure 22. Time Scaled column calculated

$$\text{New Duration} = Round\ (Average(Duration\ of\ each\ session), 100ms) \tag{3}$$

Then it has been used a new sheet in the Excel file to calculate the rest of the scaled parameters as in Figure 22. Time Scaled column calculated There is a sheet per each drone command. All session data with 'time_scaled' column and respective drone command parameter column extracted in to this new sheet. Scatter chart was drown using all these values. Polynomial trendline is drown with order of 2 as in Figure 24. Polynomial trendline for all session data Formula for 'y' axis in that curve has been used to calculate the scaled command parameter value for the new timeline. New time series is in 100ms frequency and ranged from 0 to the new duration calculated above. Figure 25. illustrates the formula used to calculate 'y'. In the next column, another formula has been used to generate the list of commands for the command mapping library as well.

This calculated data to be used as the final collection of drone commands for the respective voice command. Research creates a library with 24 collections of commands for all the movement voice commands and another 5 collections with containing exact SDK command for connect, takeoff, land, emergency and hover. This library will be used in the Voice command to drone command interpreter. Dataset for this library is illustrated in Table 25. Command Mapping File

There are multiple ways of issuing the same command. Go up command can be issued as go up, fly up, move up etc. Ultimately all these commands need to interpreted in to go up command. Because go up is the command that the system is going to issue to the drone. A dictionary has been created with mapping various voice commands in to related voice command as in Table 26. When it gets a lots of voice command variations to one drone command, it can use a trained classification model with machine learning to derive this mapping. It was decided to use a simple dictionary in this research to avoid over complexity for the research.

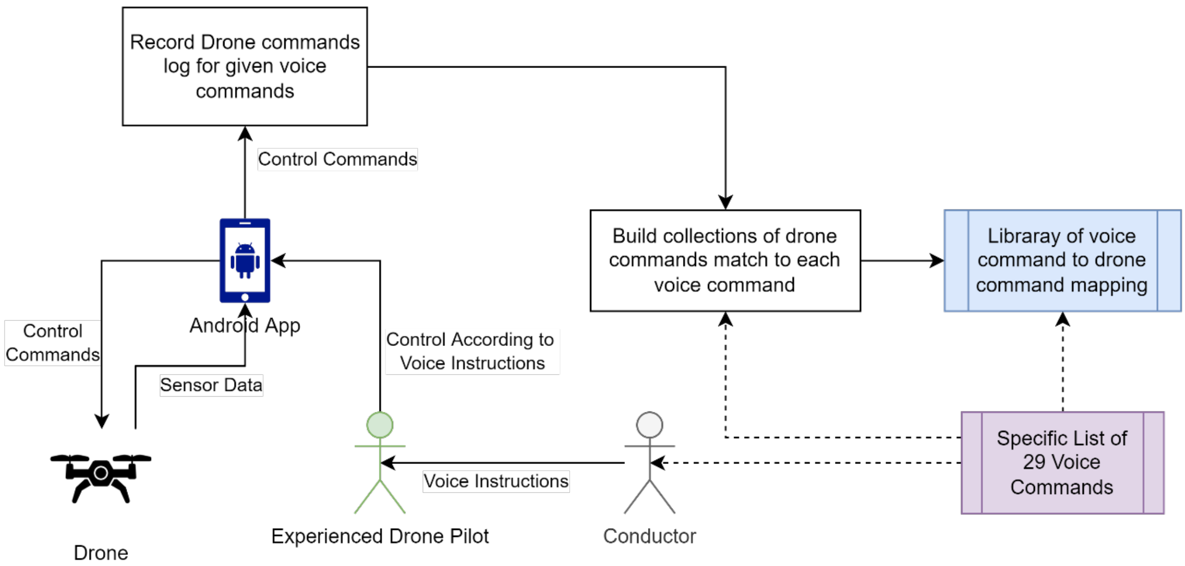


Figure 8. Expert Pilot data collection work flow.

3.6 Text to Drone Command Interpreter

Research has developed a software to command the drone control interface with interpreting recognized voice commands in to drone control commands using the library that built above. It was decided to create windows forms application using .Net Core ("Download .NET (Linux, macOS, and Windows)," n.d.) framework in C# language ("C# | Modern, open-source programming language for .NET," n.d.). Visual Studio 2022 ("Visual Studio," n.d.) is been used as the Integrated Development Environment (IDE).

This application is capable of capturing the audio from the device default microphone and stream the audio signal to Google Speech to Text API. At the first pause of speech the Api return the set of text results it could transcribe, with the rate of confidence. Windows app will pick the most confident result's text and pass it to find the voice command in the library. First it will traverse through the dictionary to find the matching drone command. If it found, then application will search the command mapping for the matching Tello SDK command. If it found, then it will execute the list of mapped drone commands to the Tello SDK command or else it will command to hold the position on air with "hover" command.

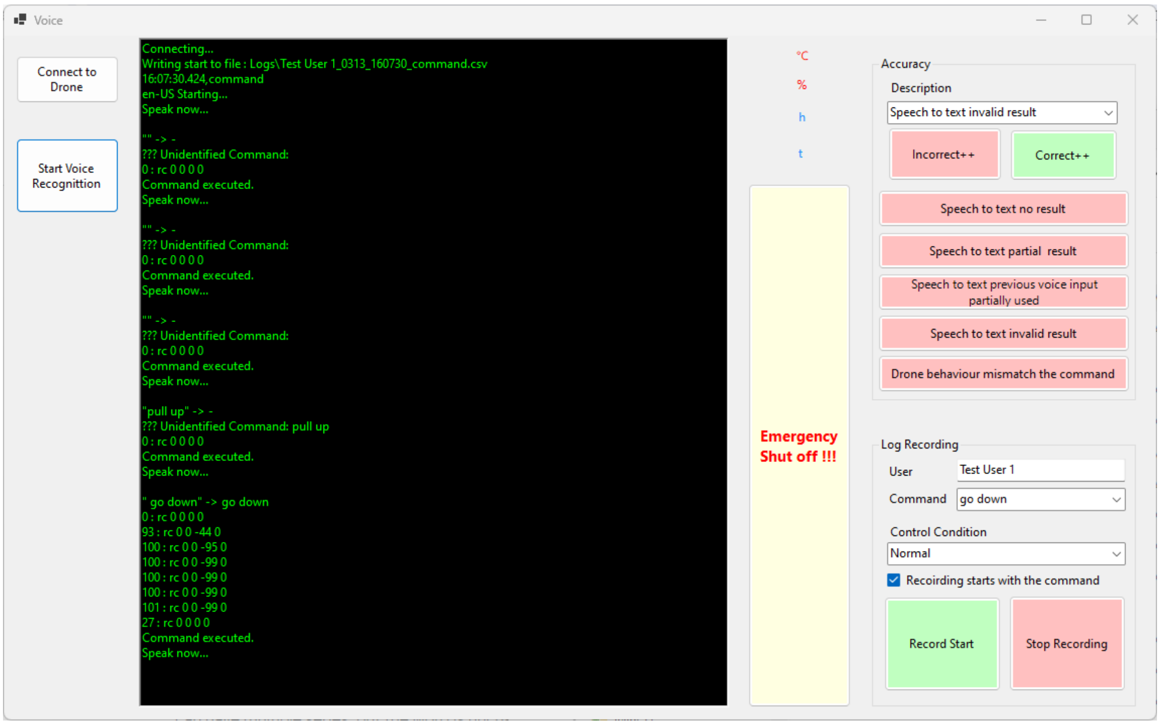


Figure 9. Voice commanding windows application

Figure 8 represent the UI of the windows application. Conductor of the experiment is responsible to handle this UI. Users name can be entered in user input field and there is a command input field as well. First the Tello drone needs to be connected to the windows application hosted machine using WIFI. Then user can click the connect button to enable the SDK mode in drone or click on Start voice button to start Google speech to text service stream and issue voice command as "connect". It can manually start or stop logging sensor data to the file using start record and stop record buttons. There is a check box option to log automatically when a voice command being executed. If the system was able to recognize the drone command for the issued voice command, then it updates the log entries with the recognized command automatically. There is an emergency shut off button created to execute the "emergency" command, which power offs the motors immediately. Since this interface does not provide the control capabilities as in Android application, this button helps to avoid damages to the drone and the participants.

UI contains an accuracy section. Accuracy counter stores the correct and incorrect executions in another log file named ‘Counters.txt’. Conductor of the experiment is responsible to increment the counters after every voice command execution. Even the other log files discarded counters needs to be updated. Purpose of these counters are to calculate the overall accuracy throughout the period of research. Inserting the description is important to analyze errors in evaluation phase. Description of the incorrect reason is needed to classify the errors. Five well known incorrect reasons has been added to the UI to quickly log the status. If it found a new reason, then it can be added to the log by typing the reason in ‘Description’ fields and clicking ‘Incorrect++’ button. ‘Correct++’ button is for log correct executions. Table 7 contains the list of known accuracy errors.

Table 7. Known Accuracy Errors

Known Accuracy Errors
Speech to text no result
Speech to text partial result
Speech to text previous voice input partially used
Speech to text previous voice input partially used
Speech to text invalid result

Ex. Correct counter file name: *Counters.txt*

Ex. Correct counter data row format: *{Timestamp},{correct/incorrect},{Description}*

Ex. Correct counter data row: *2024/03/04 10:39:55.456,incorrect,no text result*

Voice windows application log the time delay between the issuance of the voice command and the highest confident response from the speech to text service to evaluate the Response time. Also logs the delay from issuance of the command to issuance of the drone command.

Ex. Delays file name: *Delays.txt*

Ex. Delays file data format: *{Timestamp},{Text delay / command delay},{ticks}*

Ex. Delays file data format: *2024/02/28 16:53:31.818,Text delay,2340*

3.7 Drone Diagnostic Data

Once the drone enters to the SDK mode, then it will start to send the UDP packets containing the diagnostic data to the IP who initiated the command. Drone sends packets in frequency of 10Hz. IP who initiated the command can listen to the port 8890 to collect these UDP packets. Message in this packet contains sensor information from the IMU and also from the barometer and Time of Flight (TOF) range sensor. Example diagnostic message looks as follows. It contains different key value pairs concatenated in to one string.

As soon as the windows app connects to the Tello Edu drone, it will start to send the diagnostic data UDP packets to the windows app executed machine. In a different thread the windows app is listening to the port 8890 to collect these sensor data UDP packets. Windows app is built to log this sensor data information to a Comma Separated Values CSV file in the same format as mentioned in Figure 9. Android application sensor data log file

Ex. Sensor data filename format: *{Pilot}_{Session id}_{command}.csv*

Ex. Sensor data filename: *Pilot1_0215_124551_left little.csv*

This file is a CSV file used comma as the delimiter. It contains pilot name, timestamp, command and all the values mentioned in Table 8.

Table 8. Tello diagnostic data parameters and other all parameter values from UDP packet message.

Ex. Sensor data format:

{Pilot},{timestamp_ms},{command},{pitch},{roll},{yaw},{vgx},{vgy},{vgz},{templ},{temph},{tof},{h},{bat},{baro},{time},{agx},{agy},{agz}

Ex. Sensor data row:

Pilot 1,98719.3855,command,-4,-4,0,0,0,0,64,66,10,0,77,24.16,0,-76.00,72.00,-994.00

Test User 1_0212_170205_command.csv

```
Pilot,timestamp_ms,command,pitch,roll,yaw,vgx,vgy,vgz,templ,temph,tof,h,bat,baro,time,agx,agy,agz
Test User 1,98719.3855,command,-4,-4,0,0,0,0,64,66,10,0,77,24.16,0,-76.00,72.00,-994.00
Test User 1,98813.3151,command,-4,-4,0,0,0,0,64,66,10,0,77,24.41,0,-76.00,72.00,-994.00
Test User 1,98915.3912,command,-4,-4,0,0,0,0,64,66,10,0,77,24.22,0,-78.00,69.00,-994.00
Test User 1,99017.2397,command,-4,-4,0,0,0,0,64,66,10,0,77,24.14,0,-78.00,69.00,-992.00
Test User 1,99119.2244,command,-4,-4,0,0,0,0,64,66,10,0,77,24.19,0,-78.00,71.00,-994.00
Test User 1,99221.1812,command,-4,-4,0,0,0,0,64,66,10,0,77,24.14,0,-78.00,71.00,-994.00
Test User 1,99323.2447,command,-4,-4,0,0,0,0,64,66,10,0,77,24.07,0,-77.00,72.00,-994.00
Test User 1,99425.6777,command,-4,-4,0,0,0,0,64,66,10,0,77,24.18,0,-79.00,69.00,-992.00
Test User 1,99527.386,command,-4,-4,0,0,0,0,64,66,10,0,77,24.39,0,-79.00,71.00,-995.00
```

Figure 10. Android application sensor data log file

These sensor data files are used in evaluation phase to compare the experienced pilot's session performance vs voice operated performance.

Ex:

pitch:0;roll:0;yaw:0;vgx:0;vgy0;vgz:0;templ:0;temph:0;tof:0;h:0;bat:0;baro:0;time:0;agx:0.00;agy:0.00;agz:0.00;\r\n

Description of useful parameters for this research mentioned in Table 8.

Table 8. Tello diagnostic data parameters

Parameter	Description
pitch	Pitch angle in degrees
roll	Roll angle in degrees
yaw	Yaw angle in degrees
vgx	x-axis speed in decimeters/second
vgy	y-axis speed in decimeters/second
vgz	z-axis speed in decimeters/second
h	Height relative to take-off point in decimeters
tof	Time of flight sensor value in centimeters
baro	Barometer height detection in meters
time	Motor running time in seconds
agx	x-axis acceleration centimeters/second^2
agy	y-axis acceleration centimeters/second^2
agz	z-axis acceleration in centimeters/second^2

If a drone fly in the same flight path twice in the same manner; then the 'pitch', 'roll', difference of 'yaw' and differences of 'height', 'tof', 'baro' parameter values have to be the same in both flight logs at a given point of time. So, the comparison between the experienced pilots flight log and novice user's voice commanded flight log will determine the success of the process.

3.8 Experimental setup

3.8.1 Pilot Setup

First the research needed to collect data on the experienced pilot's trials. The setup environment was an empty room with enough room space to fly the drone freely. Wind level of this room needed to be minimal because the wind can affect the outcome. Floor of this room needed to have uneven texture. Because the drones VPS try to hold in to a position when it is not moving or wind caused the movement. Reflective or textured floors needed to be decorated with some random objects laying on the subjected area. That minimized the VPS interfering with the pilot's control.

Research needed to record 24 movements. Variations of movement commands mentioned in Table 4. Conductor of the trial provided the instruction to the pilot with what command to execute. There was a waiting time between each trial execution for drone to become stable in the mid-air.

Android application pre-installed mobile device has been provided to the pilot. After explaining the UI, the pilot had some time to get familiar with the drone and the control interface.

Then the trials started. Each command had to do the trial for 20 times. If the pilot is not satisfied with a trial the, that trial was repeated. Four batteries have been used in the trials. Trials interrupted multiple times because of the low battery capacity and also had to wait till the drone's temperature comes down. Finally, it was able to collect 480 trial data sets for 24 voice control commands.

3.8.2 User Setup

Condition of the room was same as the pilot trials for all users. Except the wind level, light level and the noise levels were changed while experimenting. A laptop computer used with an external microphone attached as the hardware. Conductor of the trial operate the voice windows application and microphone is positioned to capture the user's voice. After configuring the set up and introduction to the user the trials begun. First user was allowed to issue random commands without connecting to drone to test the speech to test API. Then user was informed about the "hover" command to stop in the midair or "emergency" command to shut down motors in crisis situation.

Then the conductor directed the user to issue all 24 commands in random order but to issue each command at least one time. Starts with connect, take off and ending with land. If the conductor is not satisfied with the trial because of any interference then, the trial was re done. Same problems as before were there regarding the batteries and the temperature of the drone.

When a trial is completed, it was changed the external conditions of the room one at a time. Two additional levels of noise, light and wind was used other than the normal room condition. Hence, one user had to face 9 trials. All the sensor data logs are collected to be used in Evaluation phase.

At the end of the trials, user satisfaction evaluation form was filled by all the users precented in Figure 26 and Figure 27.

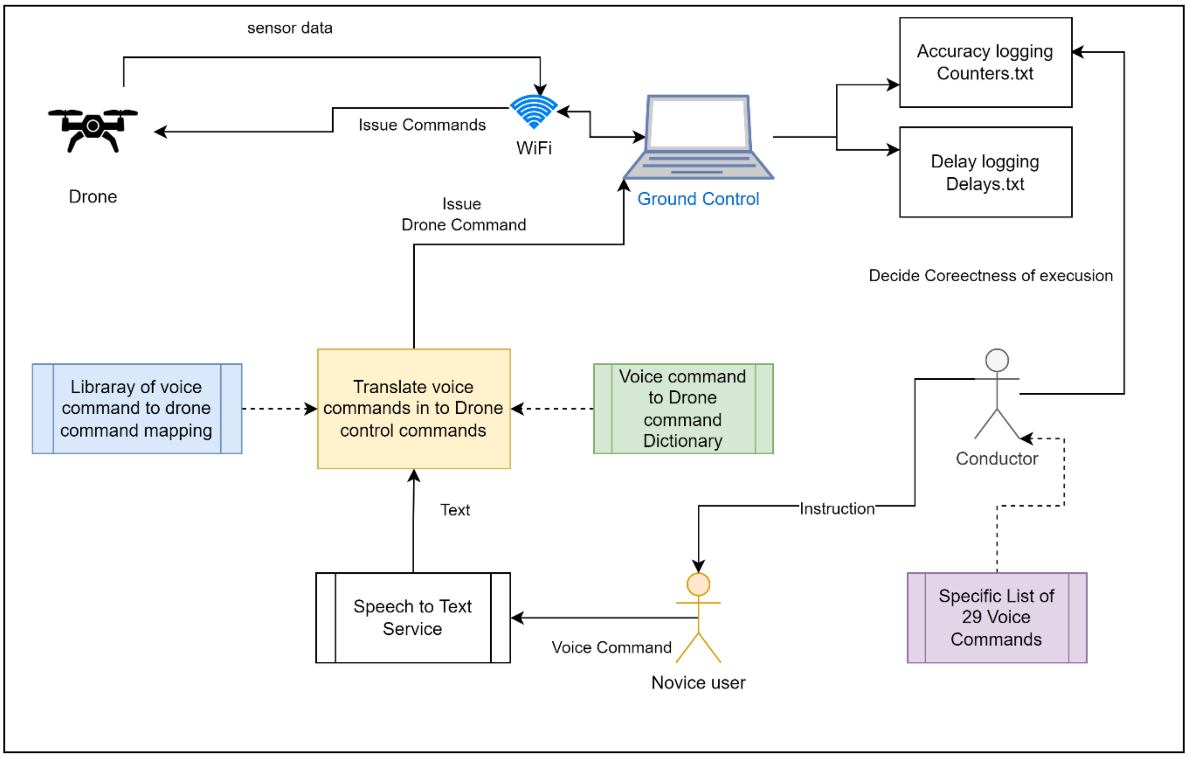


Figure 11. Novice user trial workflow

CHAPTER 4

EVALUATION AND RESULTS

System has been used to maneuver a drone by a novice user through the voice commands. Evaluation measured how precise the system controls the drone with compared to an experienced pilot. Evaluation consisted of two main parts. A quantitative performance evaluation and a qualitative user feedback evaluation. The purpose of the performance evaluation was to measure the accuracy and efficiency of the system in interpreting voice commands and executing drone actions. The purpose of the user feedback evaluation was to assess the user satisfaction and usability of the system, as well as to identify potential areas for improvement.

10 novice users have been put through the experiment. All 29 voice commands have been tested. It was created controlled environment to maneuver the drone. Different noise, light and wind conditioned introduced to the environment while testing.

4.1 Performance Evaluation

To evaluate the performance of the voice-operated drone control system, it has been conducted a series of tests in different scenarios and with different types of commands. Then measured the accuracy and efficiency of the system in terms of correctly interpreting voice commands and executing corresponding drone actions. And also measure the time to respond.
Confusion matrix has been created using the collected data in Counters.csv (Figure 48) file during user trials. That file contains the correctness of each execution of voice commands. Counters Excel file being created using the csv file. All log entries have been labeled to be used in matrix as in Figure 49.

Table 9. Confusion Matrix

		Actual	
		Positive	Negative
Predicted	Positive	244 (TP)	0 (FP)
	Negative	10 (FN)	81 (TN)

4.1.1 Accuracy

Evaluates how accurately the drone responds to voice commands. It will calculate overall accuracy and classify the types of errors occurred.

According to the Table 9. Confusion Matrix, it can calculate Recall and Accuracy as follows.

$$Recall = \frac{True\ Possitive}{True\ Positive + False\ Negative} \tag{4}$$

Recall = 2743 / (2743 + 10)

Recall = 0.9964

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total} \tag{5}$$

Accuracy = (2743 + 81) / 2834

Accuracy = 0.9965

According to the recall and accuracy values, system is capable of executing the given command correctly 99.6% of the time.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{6}$$

Precision = 2743 / (2743 + 0)

Precision = 1

Within the workflow of the project, system does not expect any negative outcome. System designed to use the 'hover' command if Google speech to text returned any unknown text. That is why the false positive result count is 0. And that 0 makes the precision to become 100%. How ever, there was one situation that can occurred false negative command. If the error is classified as 'Speech to text partial result' then, the next voice command contained part of current command in speech to text result, and the result was obviously wrong.

Command matching

Compare the user issued voice commands with the corresponding drone actions. Voice windows application hold the counters for the correct and incorrect number of executions in the Counters.txt file.

$$Accuracy = \frac{Correctly\ executed\ Commands}{\text{Correctly executed of Commands} + \text{Incorrectly Executed commands}} * 100 \tag{7}$$

Accuracy = (2834 / (2834 + 138)) * 100

Accuracy = 95.3566%

Error analysis.

Analyze instances where the drone deviates from the expected behavior. Classify errors based on types.

Table 10. Error Analysis

Error	**Count**	**Percentage**
Speech to text no result	24	26.37%
Speech to text partial result	5	5.49%
Speech to text previous voice input partially used	10	10.99%
Speech to text invalid result	26	28.57%
Drone behavior mismatch the command	26	28.57%
Total	91	

Response time

Voice windows application logged all the delays from voice command issuance to transcribed text receiving in Delay.txt file. Following statistical measurements has been extracted from the data set.

Table 11. Speech to text delay statistics

Average	332.541 milliseconds
Max	524.3 milliseconds
Min	136.3 milliseconds
Median	331.15 milliseconds
Standard Deviation	109.596 milliseconds
Count	1582

Time to respond to the voice command. Calculated by the time difference from issuing the voice command to start executing drone action.

Table 12. Speech to command execution delay statistics

Average	2071.399 milliseconds
Max	3592.9 milliseconds
Min	520.8 milliseconds
Median	520.8 milliseconds
Standard Deviation	887.424 milliseconds
Count	1582

4.1.2 Precision

The baseline for the evaluation created using the command library. All the commands have been executed using the Voice application in a Controlled room environment. Wind, light and noise factors have been kept to the optimal state to not to disturb the flight movements. Texture of the room floor is decorated with random items to avoid VPS module repositioning. There were 10 executions per each command per 24 movement commands. Diagnostics were collected in to the log files.

Total of 24 baselines has been created for 24 movement command variations. Height has to be derived from 'baro' value. Because 'h' value is in decimeters and it is not precise enough, and 'tof' value will be affected by the height of objects on the floor. Diagnostics provides 'baro' value in meters with two decimal points. It can convert to centimeters and that would be precise enough unit to measure the elevation changes.

Horizontal plane movements such as 'pitch', 'roll' and 'yaw' use degrees as the angle measurement. Since the drone always adjust it to 0 'pitch' and 0 'yaw' when it is not on movement, it can consider the two values as it is in the evaluation. But, the 'yaw' value is relative to the direction that the drone is turned on. In that case, the difference of 'yaw' value considered in evaluation.

Then all the user trial diagnostic log files processed. First it cleaned the data in all log files. Log files were continuous files per user session. Those files contained some garbage data. Table 13 contains the list of reasons caused to remove some log entries.

Table 13. Reasons to generate garbage log entries

Reasons for garbage logs
Battery state caused landing
No command for 15 seconds caused landing
Google API problems cause to restart Voice application.
Conductor had to interfere to avoid collision
Drone needed repositioning to continue testing
Drone's VPS caused un expected movements
Drone collided
Propeller came loose

Precision has been measured under two categories for the user trials. Spatial precision and Temporal precision.

Spatial Precision.

Measure how closely the drone follows spatial instructions with compared to the baseline. Log files being prepared ass mentioned below to evaluate both the spatial and temporal precisions.

All log entries have been grouped by the 24 commands and created multiple csv files. All other data has been removed except 'timestamp_ms', 'pitch', 'roll', 'yaw' and 'baro' columns. All baseline files and user trial log files updated with new calculated column 'timeline'. It represents the timeline for each trial session. This 'timeline' has been used to draw scatter charts in all baseline and user trial log files. Table 14 contains the x and y axes data columns in log files that has been used to draw the scatter charts. Scatter y axis column represents the log value that matters for the relevant type of drone movement. Average duration of each session is used to decide the timeline to draw the final comparison chart.

Table 14. Log data columns for evaluation

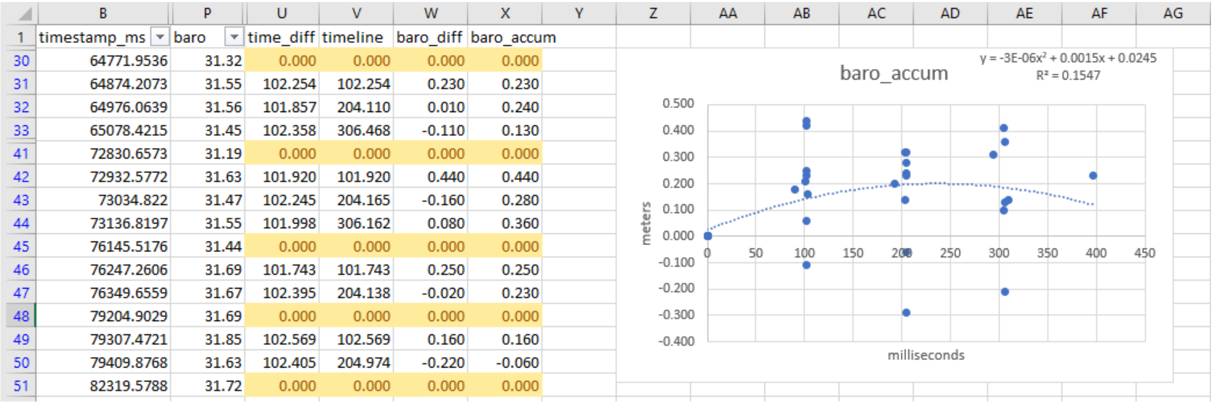
Movement command	**Scatter x axis**	**Scatter y axis**
Left, Right	timeline column	roll column
Forward, Back	timeline column	pitch column
Up, Down	timeline column	Calculated 'baro_accum' column. Accumulated baro value differences for each session.
Turn Left, Turn Right	timeline column	Calculated 'yaw_accum' column. Accumulated yaw angle differences for each session.

Then it drawn the polynomial trendline curve using order of 2. Quadratic equation for the curve has been collected. Both baseline log file and user trial log files went under this process to collect quadratic formulas. Then it found the difference of these formulas as the formula of error. This difference formula can calculate the error at any given point of time.

Following is the above-mentioned process for 'Go up little' command.

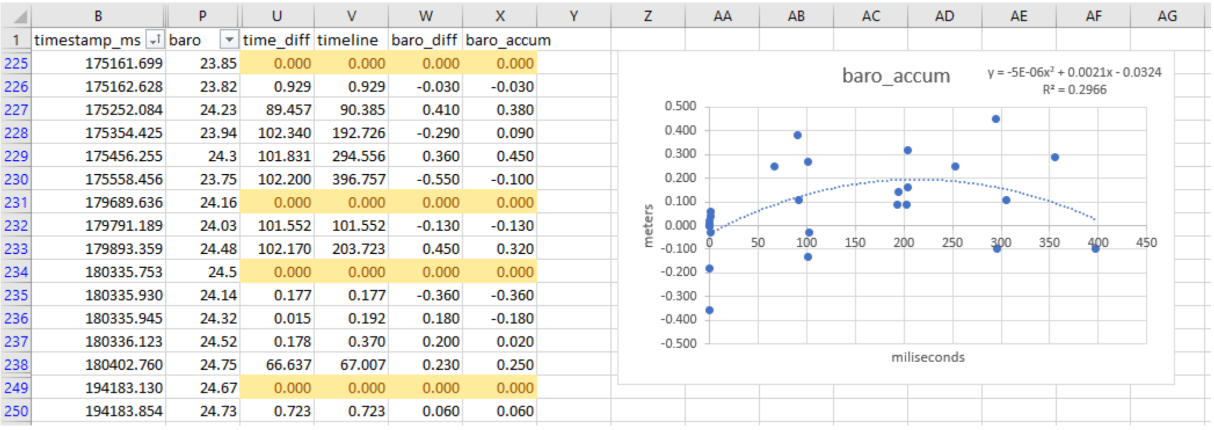
Figure 10. Baseline log file evaluation preparation illustrates the cleaned log file with calculated 'timeline' and 'baro_accum' columns. Scatter chart was drawn with mentioned data. Polynomial curve has been drawn. Then the same process applied to the user trials file as in

Figure 11. Average duration collected from baseline file 'timeline' as 293.934 milliseconds. It was decided to use 300ms as the duration to draw the final chart.



	timestamp_ms	baro	time_diff	timeline	baro_diff	baro_accum
30	64771.9536	31.32	0.000	0.000	0.000	0.000
31	64874.2073	31.55	102.254	102.254	0.230	0.230
32	64976.0639	31.56	101.857	204.110	0.010	0.240
33	65078.4215	31.45	102.358	306.468	-0.110	0.130
41	72830.6573	31.19	0.000	0.000	0.000	0.000
42	72932.5772	31.63	101.920	101.920	0.440	0.440
43	73034.822	31.47	102.245	204.165	-0.160	0.280
44	73136.8197	31.55	101.998	306.162	0.080	0.360
45	76145.5176	31.44	0.000	0.000	0.000	0.000
46	76247.2606	31.69	101.743	101.743	0.250	0.250
47	76349.6559	31.67	102.395	204.138	-0.020	0.230
48	79204.9029	31.69	0.000	0.000	0.000	0.000
49	79307.4721	31.85	102.569	102.569	0.160	0.160
50	79409.8768	31.63	102.405	204.974	-0.220	-0.060
51	82319.5788	31.72	0.000	0.000	0.000	0.000

Figure 12. Baseline log file evaluation preparation



	timestamp_ms	baro	time_diff	timeline	baro_diff	baro_accum
225	175161.699	23.85	0.000	0.000	0.000	0.000
226	175162.628	23.82	0.929	0.929	-0.030	-0.030
227	175252.084	24.23	89.457	90.385	0.410	0.380
228	175354.425	23.94	102.340	192.726	-0.290	0.090
229	175456.255	24.3	101.831	294.556	0.360	0.450
230	175558.456	23.75	102.200	396.757	-0.550	-0.100
231	179689.636	24.16	0.000	0.000	0.000	0.000
232	179791.189	24.03	101.552	101.552	-0.130	-0.130
233	179893.359	24.48	102.170	203.723	0.450	0.320
234	180335.753	24.5	0.000	0.000	0.000	0.000
235	180335.930	24.14	0.177	0.177	-0.360	-0.360
236	180335.945	24.32	0.015	0.192	0.180	-0.180
237	180336.123	24.52	0.178	0.370	0.200	0.020
238	180402.760	24.75	66.637	67.007	0.230	0.250
249	194183.130	24.67	0.000	0.000	0.000	0.000
250	194183.854	24.73	0.723	0.723	0.060	0.060

Figure 13. User trails log file evaluation preparation

Following b(y) and u(y) are the quadratic equations of baseline and user trials in order.

$$b(y) = -0.000003x^2 + 0.0015x + 0.0245 \tag{8}$$

$$u(y) = -0.000005x^2 + 0.0021x + 0.0324 \tag{9}$$

The difference formula f(y) has been derived by subtracting b(y) from u(y).

$$f(y) = -0.000002x^2 + 0.0006x + 0.0569 \tag{10}$$

New timeline created in 50 milliseconds frequency and 300ms as the max time as mentioned before. Then all above three formulas has been used to generate 'baro_accum' values as in Table 15. Scaled values of baseline, user trials and error Then the graph in Figure 12. Scaled curves for baseline, user trials and error has been drown to represent the baseline, user trials vs error trend lines.

Table 15. Scaled values of baseline, user trials and error

timeline	user trials	baseline	error
0	0.0324	0.0245	0.0569
50	0.1249	0.092	0.0819
100	0.1924	0.1445	0.0969
150	0.2349	0.182	0.1019
200	0.2524	0.2045	0.0969
250	0.2449	0.212	0.0819
300	0.2124	0.2045	0.0569

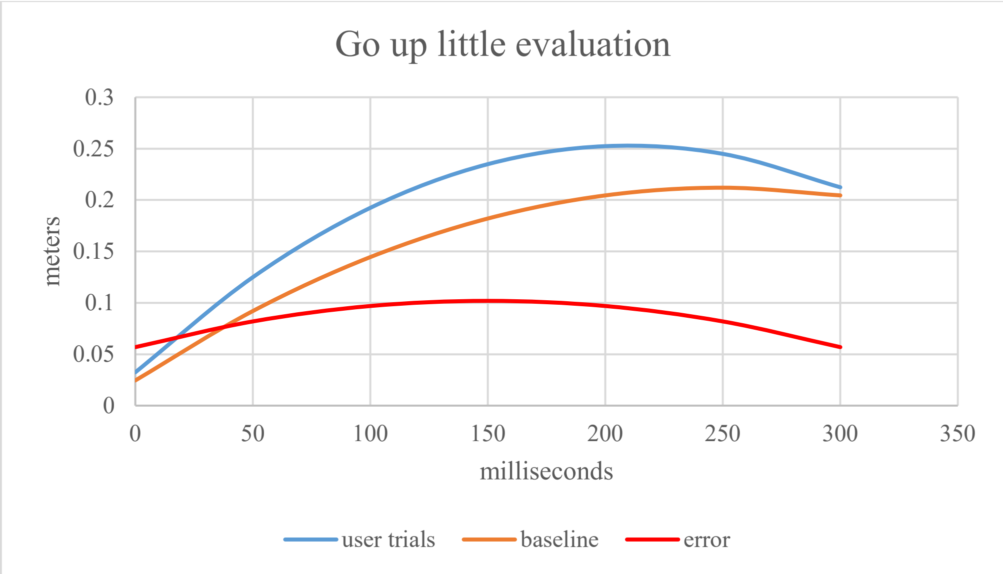


Figure 14. Scaled curves for baseline, user trials and error

Graphs, Tables and formulas for all the 24 drone commands can be found in Error trends by command section in APPENDICES.

Temporal Precision.

Measure the time it takes for the drone to complete each action after receiving the voice command vs time to complete by the experienced pilot. Same prepared log files above are used to evaluate. First it averaged the durations in the baseline files for each session and then it averaged all the user trial file durations for each user trial session. Time difference express the temporal precision of each command.

Table 16. Temporal precision in normal condition

Command	Baseline average duration	User trial average duration	Duration difference in milliseconds
GO UP	685.115	711.309	-26.194
GO UP Little	293.934	270.819	23.115
GO UP Lot	796.29	886.375	-90.085
GO Down	610.458	623.63	-13.172
GO Down Little	228.429	224.388	4.041
GO Down Lot	558.511	584.788	-26.277
GO Left	2498.756	2804.53	-305.774
GO Left Little	306.652	310.451	-3.799
GO Left Lot	5977.217	7345.587	-1368.37
GO Right	2294.288	2192.554	101.734
GO Right Little	1102.244	1070.8709	31.3731
GO Right Lot	5226.993	5403.261	-176.268
GO Forward	1581.268	1530.214	51.054
GO Forward Little	2159.55	2499.508	-339.958
GO Forward Lot	2297.505	2771.736	-474.231
GO Back	1377.065	1958.067	-581.002
GO Back Little	2295.606	2654.456	-358.85
GO Back Lot	2548.691	2684.164	-135.473
Turn Left	407.874	442.586	-34.712
Turn Left Little	152.472	182.419	-29.947
Turn Left Lot	1376.864	1395.581	-18.717
Turn Right	407.754	450.386	-42.632
Turn Right Little	150.946	159.553	-8.607
Turn Right Lot	1350.254	1355.478	-5.224

4.1.3 Robustness

Test the system's robustness under different environmental conditions. Evaluate its performance in various factors that may affect voice recognition and drone operation.

4.1.4 Noise level.

Noice level in a park has been simulated by playing an mp3 track near the user by an external speaker to simulate noise. Logs collected in noise condition compared with the previously prepared baseline files. Process is the same as previous. Graphs, Tables and formulas for spatial precision on noisy condition located in Error trends by command with noise section in APPENDICES.

Temporal precision in noisy condition mentioned in Table 17.

Table 17. Temporal precision in noisy condition

Command	Baseline average duration	User trial average duration	Duration difference in milliseconds
GO UP	685.115	713.230	-28.115
GO UP Little	293.934	271.550	22.384
GO UP Lot	796.29	888.768	-92.478
GO Down	610.458	625.314	-14.856
GO Down Little	228.429	224.994	3.435
GO Down Lot	558.511	586.367	-27.856
GO Left	2498.756	2812.102	-313.346
GO Left Little	306.652	311.289	-4.637
GO Left Lot	5977.217	7365.420	-1388.203
GO Right	2294.288	2198.474	95.814
GO Right Little	1102.244	1073.762	28.482
GO Right Lot	5226.993	5417.850	-190.857
GO Forward	1581.268	1534.346	46.922
GO Forward Little	2159.55	2506.257	-346.707
GO Forward Lot	2297.505	2779.220	-481.715
GO Back	1377.065	1963.354	-586.289
GO Back Little	2295.606	2661.623	-366.017
GO Back Lot	2548.691	2691.411	-142.720
Turn Left	407.874	443.781	-35.907
Turn Left Little	152.472	182.912	-30.440
Turn Left Lot	1376.864	1399.349	-22.485
Turn Right	407.754	451.602	-43.848
Turn Right Little	150.946	159.984	-9.038
Turn Right Lot	1350.254	1359.138	-8.884

4.1.5 Wind level.

Artificial wind level was introduced across the drone flight field using a pedestal fan. Logs collected in wind condition compared with the previously prepared baseline files. Process is the same as previous. Graphs, Tables and formulas for spatial precision on windy condition located in Error trends by command with wind section in APPENDICES.

Temporal precision in windy condition mentioned in Table 18. Temporal precision in windy condition

Table 18. Temporal precision in windy condition

Command	**Baseline average duration**	**User trial average duration**	**Duration difference in milliseconds**
GO UP	685.115	725.354	-40.239
GO UP Little	293.934	276.167	17.767
GO UP Lot	796.29	903.877	-107.587
GO Down	610.458	635.944	-25.486
GO Down Little	228.429	228.819	-0.390
GO Down Lot	558.511	596.335	-37.824
GO Left	2498.756	2859.908	-361.152
GO Left Little	306.652	316.581	-9.929
GO Left Lot	5977.217	7490.632	-1513.415
GO Right	2294.288	2235.848	58.440
GO Right Little	1102.244	1092.016	10.228
GO Right Lot	5226.993	5509.953	-282.960
GO Forward	1581.268	1560.429	20.839
GO Forward Little	2159.55	2548.863	-389.313
GO Forward Lot	2297.505	2826.466	-528.961
GO Back	1377.065	1996.731	-619.666
GO Back Little	2295.606	2706.871	-411.265
GO Back Lot	2548.691	2737.165	-188.474
Turn Left	407.874	451.325	-43.451
Turn Left Little	152.472	186.021	-33.549
Turn Left Lot	1376.864	1423.138	-46.274
Turn Right	407.754	459.279	-51.525
Turn Right Little	150.946	162.704	-11.758
Turn Right Lot	1350.254	1382.243	-31.989

4.1.6 Test system response with different user accents and speech patterns.

No significant speech patterns were found while the trials. All of the subjects were using English a s their second language. Mostly Google Speech to text API was not confused with the inputs. Due to lack of variety in accent and speech patterns, this evaluation was not conducted.

4.2 User Feedback Evaluation

After the user trials each user was provided with the feedback form. Google forms was used as the user feedback platform. Considered factors such as following mentioned in Figure 26 and Figure 27.

4.2.1 Ease of learning,

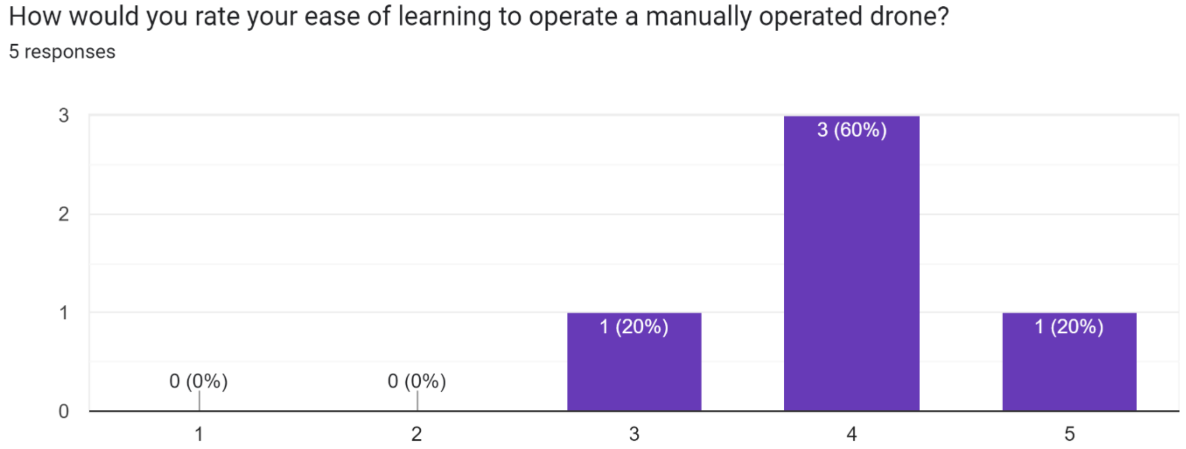


Figure 15. Ease of learning feedback result

4.2.2 Ease of use

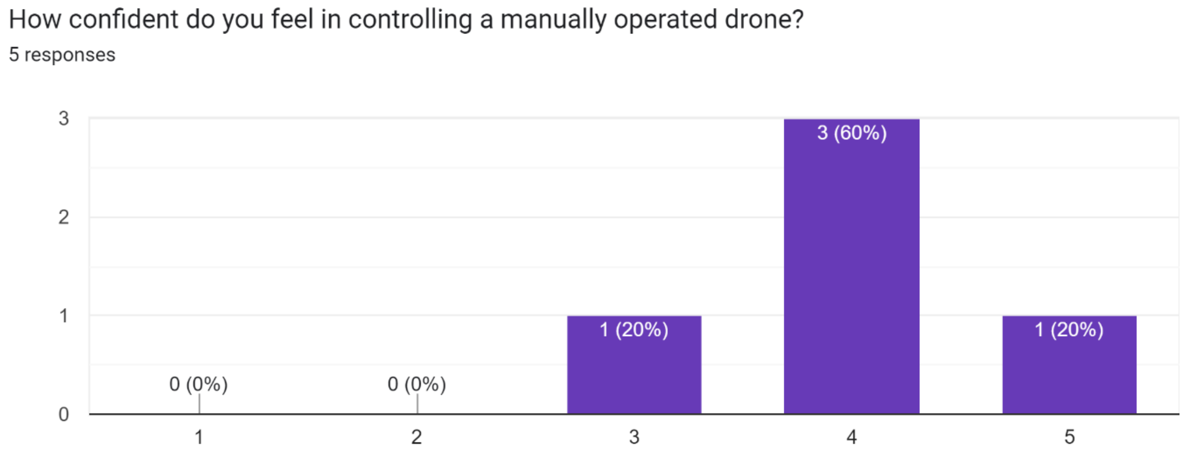


Figure 16. Ease of use feedback result

4.2.3 Overall user satisfaction.

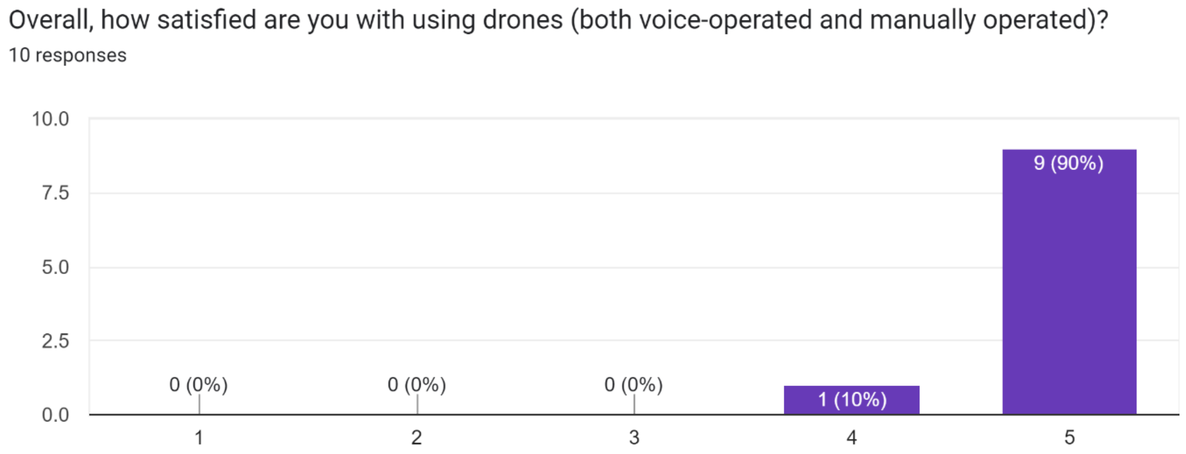


Figure 17. Overall user Satisfaction

4.2.4 Intuitiveness of the voice commands and the system's responsiveness.

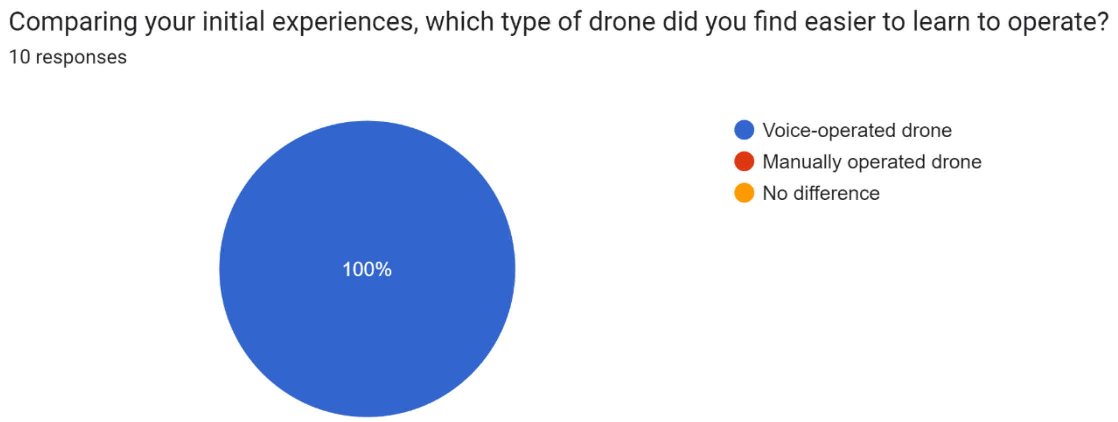


Figure 18. Easier to learn rating

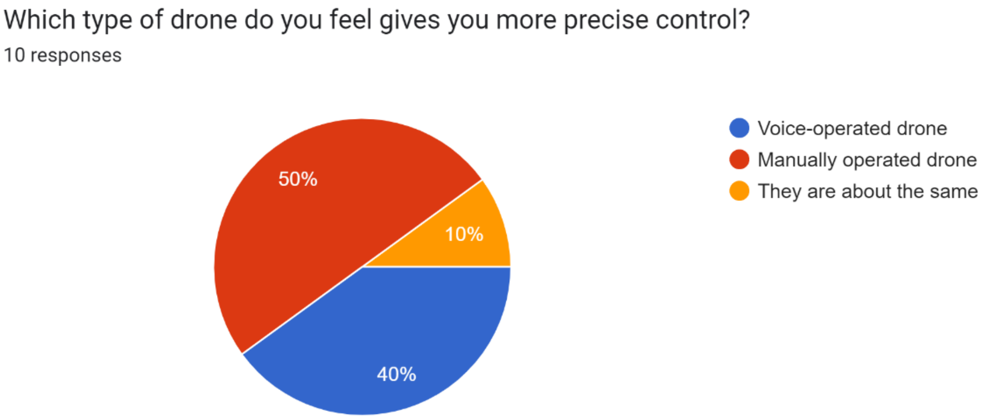


Figure 19. More Precise control rating

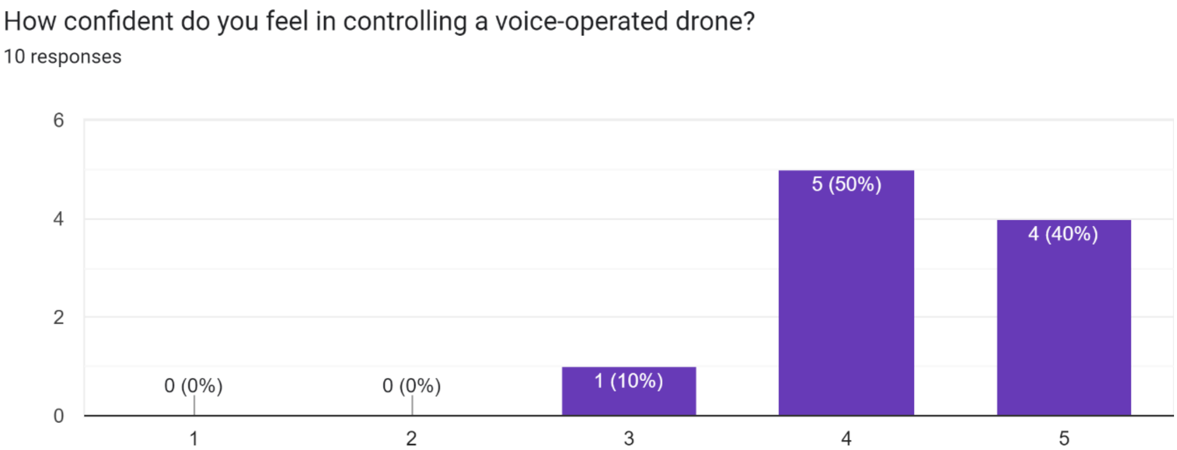


Figure 20. Confidence in voice-operated drone

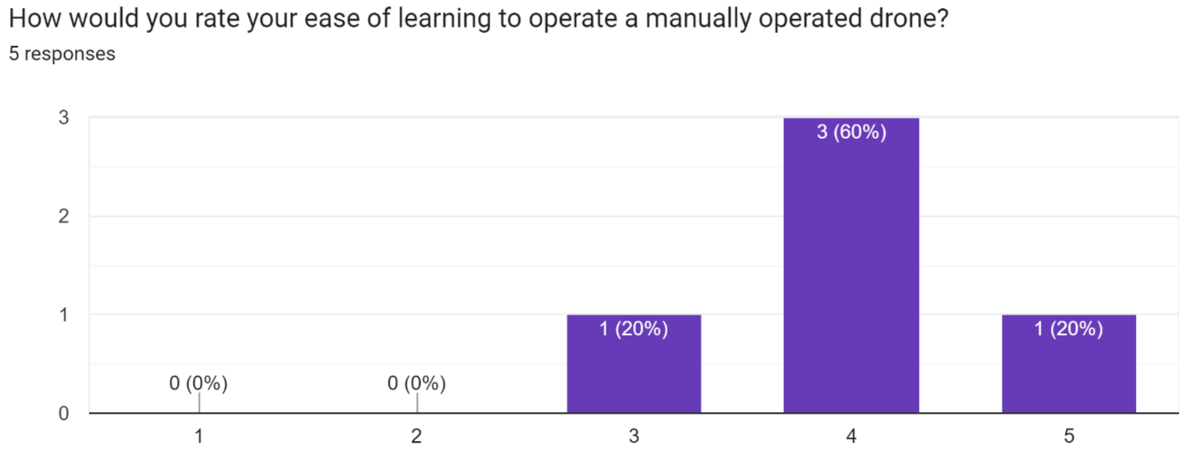


Figure 21. Ease of learning rating

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

This research aimed to design and develop a voice-operated control system for drones, making drone technology more accessible and user-friendly. The primary objective was to replace traditional, complex manual controls with a simple, intuitive interface that allows users to control drones using voice commands. The project successfully demonstrated that voice commands could effectively control drones, eliminating the need for specialized training and making drone technology accessible to a broader audience.

The system was built using advanced speech recognition technology from Google, integrated with the DJI Tello Edu drone. The choice of the DJI Tello Edu drone was driven by its suitability for indoor environments, programmability, and affordability. The voice command system achieved a high accuracy rate of 99.6% in controlled environments, showcasing its potential for reliable and efficient drone operation. The research included extensive testing under various conditions, including noisy and windy environments, to evaluate the system's robustness and adaptability.

One of the key achievements of this research is the significant simplification of drone operation. Users found the system easy to use and appreciated the convenience of controlling the drone with their voice. This accessibility could lead to wider adoption of drone technology in various fields, such as inspections, emergency responses, agriculture, surveillance, and entertainment. The ability to control drones through voice commands opens up new possibilities for integrating drones into everyday tasks and professional applications, enhancing efficiency and productivity.

The voice-operated control system developed in this research represents a substantial step forward in making drone technology more user-friendly. By providing an intuitive interface, the system reduces the barriers to entry for using drones, allowing individuals with limited technical expertise to operate drones effectively. This democratization of drone technology has the potential to expand the use of drones in numerous industries and applications, contributing to increased efficiency and innovative solutions in various sectors.

5.2 Future Work

While this research made great progress, there are several areas for future improvement. One key area is improving the system to handle different accents, languages, and noisy environments better. This could include better handling of accents, dialects, and languages other than English, as well as reducing the impact of background noise and wind. Enhanced voice recognition algorithms and noise-cancellation techniques could further increase the system's robustness and accuracy.

Adding more and varied voice commands could enhance the system’s functionality. This could involve commands for specific tasks or sequences of actions, making the drone capable of more sophisticated operations. Developing a more comprehensive command library could enable users to perform complex maneuvers and tasks more efficiently.

Using advanced machine learning could help the system adapt to individual users' speech patterns and preferences, thereby increasing accuracy and user satisfaction. For example, a machine learning model that learns from user interactions could progressively enhance the precision of command recognition. Personalized models could improve the system’s performance for diverse user groups.

Making the system reliable for outdoor use, where conditions can change significantly, would expand its applicability. This would involve addressing challenges related to GPS integration, weather conditions, and longer communication ranges. Developing algorithms that can compensate for environmental variations could enhance the system's versatility.

The DJI Tello Edu drone does not have GPS capabilities and responds to commands based only on its local coordinate system. An external program is needed to convert global coordinate system commands into local coordinate system commands using a rotational matrices-based algorithm. Developing this conversion program would enhance this research by enabling more precise and versatile drone operations.

The delay associated with the speech-to-text module could be reduced if there were a program capable of transcribing speech to text on local computers. This would decrease response time and remove dependency on the Google speech-to-text service. Implementing local speech recognition could make the system faster and more reliable.

Incorporating all processes into a mobile device is another enhancement. This would simplify usage, allowing both the experienced pilot and the novice user to use the same device to complete the process. A mobile application integrating voice recognition, drone control, and user feedback mechanisms could provide a seamless user experience.

Developing a more intuitive and interactive user interface, possibly including visual feedback and error correction mechanisms, could further enhance the user experience. Real-time feedback on command execution and visual cues could help users refine their commands and achieve better control.

5.3 Limitations

Despite the promising results, this research had several limitations. The system was mostly tested indoors, and its performance in outdoor settings with changing conditions still needs to be fully tested and confirmed. Environmental factors such as wind, light, and obstacles in outdoor environments could impact the system's effectiveness.

The system was highly accurate in quiet conditions, but its performance was slightly lower in noisy or windy environments. Further refinement is needed to maintain high accuracy across all conditions. Advanced noise-cancellation and context-aware algorithms could help mitigate these issues.

The current set of voice commands is basic and limited to simple drone movements. More complex commands and sequences were not explored in this research. Future research should expand the command library to include more advanced functions and multi-step operations.

The system's ability to adapt to different users, especially those with strong accents or speech impairments, was not extensively tested. Future research should focus on making the system more inclusive by incorporating a wider range of speech patterns and dialects. Personalized models and adaptive learning techniques could improve the system's performance for diverse users.

One significant limitation is that voice command operation does not offer continuous control like traditional radio controllers, which allow precise adjustments with fraction-of-a-second accuracy. Voice commands are discrete actions that may not provide the same level of precision and smooth control. Future improvements could explore hybrid control systems that combine voice commands with more precise control methods.

The DJI Tello Edu drone’s Vision Positioning System (VPS) is always in a feedback loop with the drone’s flight controller, which can interfere with the ‘hover’ state or the movements of the drone. The drone itself does not provide any diagnostic information about how much the VPS affects maneuvering. It would be more accurate if a drone that can measure the self-adjustments made by the flight controller were used.

The vertical movement stability of the Tello Edu drone is much higher than its horizontal movement stability. The experienced pilot had to conduct more trials to get used to the horizontal control range. Although the manufacturer states the battery life of the drone is expected to be 13 minutes, it is much less in practical usage. This research used 4 batteries in user trials but still had to take frequent intervals to charge the batteries.

This research used the Remote Procedure Call (RPC) service of Google speech-to-text. This service has a limitation of 5 minutes for continuous voice streaming. Using Google's Web API speech-to-text service could overcome this time limit.

In conclusion, this thesis has shown that a voice-operated control system for drones is feasible and can make drone technology more accessible and user-friendly. By addressing the identified limitations and pursuing the outlined future work, further advancements can be achieved, broadening the scope and impact of this innovative control method. The integration of voice-operated systems into everyday drone use not only simplifies operation but also opens up new possibilities for their application in various fields, enhancing efficiency and expanding their utility.

APPENDICES

Project Plan

		Task Mode	Task Name	Duration	Start	Finish
1			**Development**	**175 days**	**Mon 5/1/23**	**Fri 12/29/23**
2			Obtain Hardware	30 days	Mon 5/1/23	Fri 6/9/23
3			Prepare Development Environment	15 days	Mon 5/1/23	Fri 5/19/23
4			Develop Intepriter	30 days	Mon 11/20/23	Fri 12/29/23
5			Prepare Evaluation	15 days	Mon 5/1/23	Fri 5/19/23
6			Objective Evaluation	30 days	Tue 9/12/23	Mon 10/23/23
7			Subjective Evaluation	30 days	Tue 10/24/23	Mon 12/4/23
8			**Thesis**	**280 days**	**Fri 3/24/23**	**Wed 4/17/24**
9			Proposal Final Submission	104 days	Fri 3/24/23	Tue 8/15/23
10			Introduction Chapter	30 days	Mon 10/2/23	Fri 11/10/23
11			Literature Review Chapter	30 days	Mon 10/2/23	Fri 11/10/23
12			Method Chapter	30 days	Mon 10/2/23	Fri 11/10/23
13			Interim Report	30 days	Mon 10/2/23	Fri 11/10/23
14			Evaluation Chapter	16 days	Mon 11/13/23	Mon 12/4/23
15			Final Thesis	77 days	Tue 12/5/23	Wed 3/20/24
16			Defence	1 day	Sat 3/23/24	Sat 3/23/24
17			Hardbound Thesis	20 days	Thu 3/21/24	Wed 4/17/24

Figure 22. Project Plan

Files for Building the Library

	time	action	comma	y	x	z	yaw	timegap_calcula	timegap_adjuste	timeline	time_rounded	time_scaled	session_id	action_prefix	action_category	amount_catego	log_entry
2	07:56:04.302	up	rc	0	0	10	0	0	0	0	0	0	1	go	up	little	go up little,0,rc 0 0 10 0
3	07:56:04.309	up	rc	0	0	1	0	7	7	7	0	12	1	go	up	little	go up little,7.000566,rc 0 0 1 0
4	07:56:04.402	up	rc	0	0	27	0	93	93	100	100	174	1	go	up	little	go up little,92.999521,rc 0 0 27 0
5	07:56:04.504	up	rc	0	0	99	0	102	102	202	200	351	1	go	up	little	go up little,102.000427,rc 0 0 99 0
6	07:56:04.604	up	rc	0	0	99	0	100	100	302	300	524	1	go	up	little	go up little,100.000086,rc 0 0 99 0
7	07:56:04.650		rc	0	0	0	0	46	46	348	300	604	1	go	up	little	go up little,45.999675,rc 0 0 0 0
8	07:56:05.165	Clockwise	rc	0	0	0	7	515	515	863	900	1498	1	go	up	little	go up little,514.999847,rc 0 0 0 7
9	07:56:05.166		rc	0	0	0	0	1	1	864	900	1500	1	go	up	little	go up little,1.000171,rc 0 0 0 0
10	07:56:05.266	up	rc	0	0	23	0	0	0	0	0	0	2	go	up	little	go up little,0,rc 0 0 23 0
11	07:56:05.367	up	rc	0	0	100	0	101	101	101	100	198	2	go	up	little	go up little,100.999628,rc 0 0 100 0
12	07:56:05.467	up	rc	0	0	100	0	100	100	201	200	394	2	go	up	little	go up little,100.000086,rc 0 0 100 0
13	07:56:05.568	up	rc	0	0	100	0	101	101	302	300	591	2	go	up	little	go up little,101.000257,rc 0 0 100 0
14	07:56:05.591		rc	0	0	0	0	23	23	325	300	636	2	go	up	little	go up little,23.000152,rc 0 0 0 0
15	07:56:06.031	up	rc	0	0	18	0	440	440	765	800	1498	2	go	up	little	go up little,439.999625,rc 0 0 18 0
16	07:56:06.032		rc	0	0	0	0	1	1	766	800	1500	2	go	up	little	go up little,1.000171,rc 0 0 0 0
17	07:56:06.131	up	rc	0	0	48	0	0	0	0	0	0	3	go	up	little	go up little,0,rc 0 0 48 0
18	07:56:06.232	up	rc	0	0	99	0	101	101	101	100	93	3	go	up	little	go up little,101.000257,rc 0 0 99 0
19	07:56:06.333	up	rc	0	0	100	0	101	101	202	200	186	3	go	up	little	go up little,100.999628,rc 0 0 100 0
20	07:56:06.430	up	rc	0	0	99	0	97	97	299	300	275	3	go	up	little	go up little,97.000203,rc 0 0 99 0
21	07:56:06.502		rc	0	0	0	0	72	72	371	400	341	3	go	up	little	go up little,71.99971,rc 0 0 0 0
22	07:56:07.750	up	rc	0	0	30	0	1248	1248	1619	1600	1487	3	go	up	little	go up little,1248.000422,rc 0 0 30 0
23	07:56:07.764		rc	0	0	0	0	14	14	1633	1600	1500	3	go	up	little	go up little,13.999874,rc 0 0 0 0
24	07:56:07.850	up	rc	0	0	20	0	0	0	0	0	0	4	go	up	little	go up little,0,rc 0 0 20 0
25	07:56:07.953	up	rc	0	0	84	0	103	103	103	100	104	4	go	up	little	go up little,102.999969,rc 0 0 84 0
26	07:56:08.051	up	rc	0	0	99	0	98	98	201	200	202	4	go	up	little	go up little,97.999745,rc 0 0 99 0
27	07:56:08.151	up	rc	0	0	99	0	100	100	301	300	303	4	go	up	little	go up little,100.000086,rc 0 0 99 0
28	07:56:08.247		rc	0	0	0	0	96	96	397	400	399	4	go	up	little	go up little,96.000032,rc 0 0 0 0
29	07:56:09.339	Down	rc	0	0	-20	0	1092	1092	1489	1500	1498	4	go	up	little	go up little,1092.000213,rc 0 0 -20 0
30	07:56:09.341		rc	0	0	0	0	2	2	1491	1500	1500	4	go	up	little	go up little,1.999713,rc 0 0 0 0
31	07:56:09.438	up	rc	0	0	28	0	0	0	0	0	0	5	go	up	little	go up little,0,rc 0 0 28 0
32	07:56:09.538	up	rc	0	0	76	0	100	100	100	100	106	5	go	up	little	go up little,100.000086,rc 0 0 76 0
33	07:56:09.638	up	rc	0	0	100	0	100	100	200	200	212	5	go	up	little	go up little,99.999458,rc 0 0 100 0
34	07:56:09.738	up	rc	0	0	100	0	100	100	300	300	318	5	go	up	little	go up little,100.000086,rc 0 0 100 0
35	07:56:09.825		rc	0	0	0	0	87	87	387	400	410	5	go	up	little	go up little,87.000383,rc 0 0 0 0
36	07:56:10.845	Clockwise	rc	0	0	0	1	1020	1020	1407	1400	1489	5	go	up	little	go up little,1019.999874,rc 0 0 0 1
37	07:56:10.855	up	rc	0	0	0	0	10	10	1417	1400	1500	5	go	up	little	go up little,9.99982,rc 0 0 0 0
38	07:56:10.946	up	rc	0	0	12	0	0	0	0	0	0	6	go	up	little	go up little,0,rc 0 0 12 0
39	07:56:11.047	up	rc	0	0	74	0	101	101	101	100	100	6	go	up	little	go up little,101.000257,rc 0 0 74 0
40	07:56:11.147	up	rc	0	0	100	0	100	100	201	200	199	6	go	up	little	go up little,100.000086,rc 0 0 100 0
41	07:56:11.247	up	rc	0	0	100	0	100	100	301	300	299	6	go	up	little	go up little,100.000086,rc 0 0 100 0
42	07:56:11.347	up	rc	0	0	100	0	100	100	401	400	398	6	go	up	little	go up little,100.000086,rc 0 0 100 0
43	07:56:11.423		rc	0	0	0	0	76	76	477	500	473	6	go	up	little	go up little,75.999764,rc 0 0 0 0
44	07:56:12.446	Down	rc	0	0	-11	0	1023	1023	1500	1500	1488	6	go	up	little	go up little,1022.999757,rc 0 0 -11 0
45	07:56:12.458		rc	0	0	0	0	12	12	1512	1500	1500	6	go	up	little	go up little,12.000161,rc 0 0 0 0
46	07:56:12.547	up	rc	0	0	19	0	0	0	0	0	0	7	go	up	little	go up little,0,rc 0 0 19 0
47	07:56:12.650	up	rc	0	0	76	0	103	103	103	100	81	7	go	up	little	go up little,102.999969,rc 0 0 76 0
48	07:56:12.748	up	rc	0	0	100	0	98	98	201	200	158	7	go	up	little	go up little,97.999745,rc 0 0 100 0

Figure 23. Collection of all experienced flight logs

L98 =J98*1400/J107

	time	action	comma	y	x	z	yaw	timegap_calcula	timegap_adjuste	timeline	time_rounded	time_scaled	ses
97	07:56:25.165	up	rc	0	0	11	0	0	0	0	0	0	
98	07:56:25.265	up	rc	0	0	64	0	100	100	100	100	69	
99	07:56:25.365	up	rc	0	0	100	0	100	100	200	200	137	
100	07:56:25.466	up	rc	0	0	100	0	101	101	301	300	207	
101	07:56:25.567	up	rc	0	0	100	0	101	101	402	400	276	
102	07:56:25.666	up	rc	0	0	100	0	99	99	501	500	344	
103	07:56:25.767	up	rc	0	0	100	0	101	101	602	600	413	
104	07:56:25.866	up	rc	0	0	100	0	99	99	701	700	481	
105	07:56:25.890		rc	0	0	0	0	24	24	725	700	498	
106	07:56:27.192	Clockwise	rc	0	0	0	27	1302	1302	2027	2000	1392	
107	07:56:27.204		rc	0	0	0	0	12	12	2039	2000	1400	

Figure 24. Time Scaled column calculated

	A	B	C	D	E	F
1	go up little					
2	z	time_scaled		time_scaled	z_scaled	drone_commands
3	10	0		0	45	go up little,0,0 0 45 0
4	1	12		100	51	go up little,100,0 0 51 0
5	27	174		200	55	go up little,200,0 0 55 0
6	99	351		300	58	go up little,300,0 0 58 0
7	99	524		400	60	go up little,400,0 0 60 0
8	0	604		500	61	go up little,500,0 0 61 0
9	0	1498		600	61	go up little,600,0 0 61 0
10	0	1500		700	59	go up little,700,0 0 59 0
11	23	0		800	56	go up little,800,0 0 56 0
12	100	198		900	52	go up little,900,0 0 52 0
13	100	394		1000	47	go up little,1000,0 0 47 0
14	100	591		1100	41	go up little,1100,0 0 41 0
15	0	636		1200	33	go up little,1200,0 0 33 0
16	18	1498		1300	24	go up little,1300,0 0 24 0
17	0	1500		1400	14	go up little,1400,0 0 14 0
18	48	0		1500	3	go up little,1500,0 0 3 0
19	99	93				
20	100	186				
21	99	275				
22	0	341				
23	30	1487				
24	0	1500				
25	20	0				
26	84	104				
27	99	202				
28	99	303				
29	0	399				
30	-20	1498				
31	0	1500				
32	28	0				
33	76	106				
34	100	212				
35	100	318				
36	0	410				
37	0	1489				

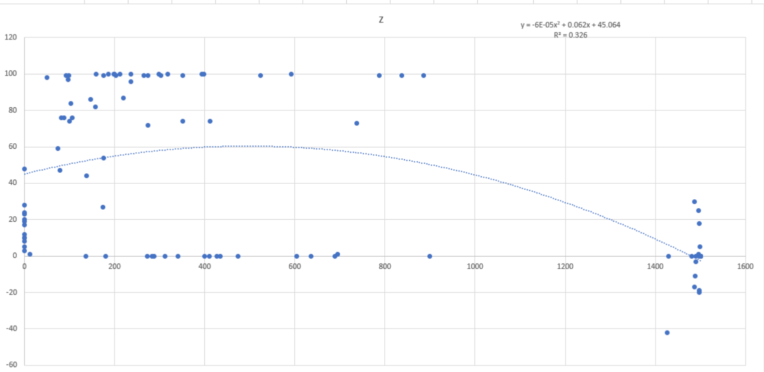


Figure 25. Go up little command collection sheet

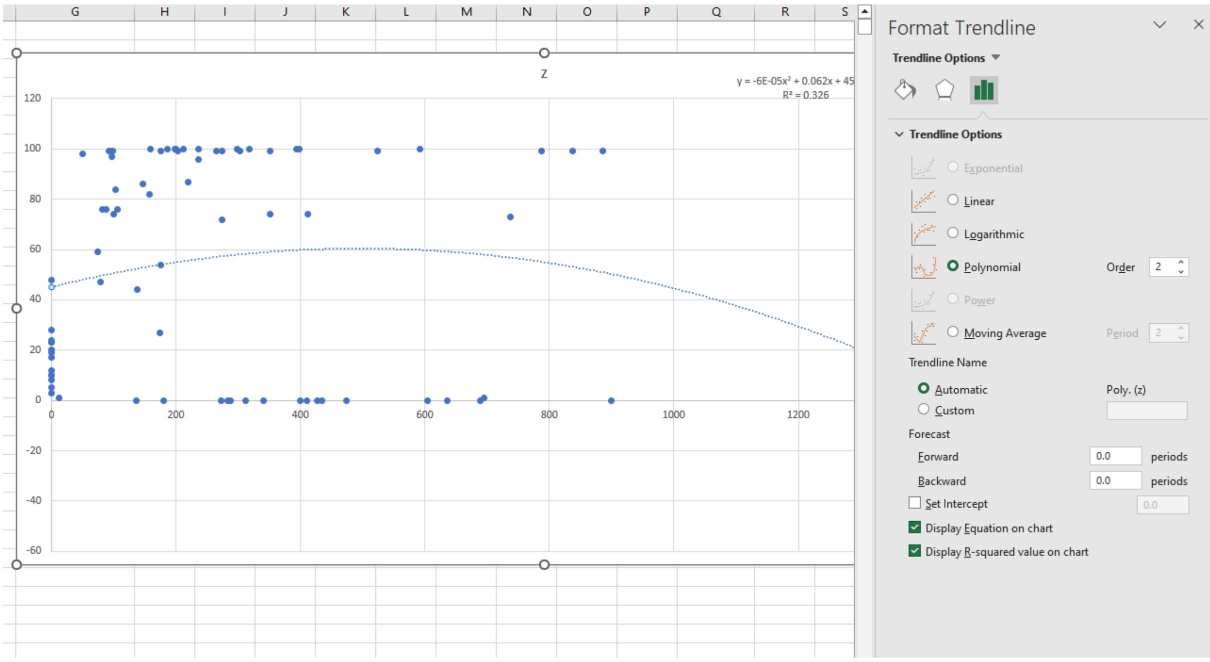


Figure 26. Polynomial trendline for all session data

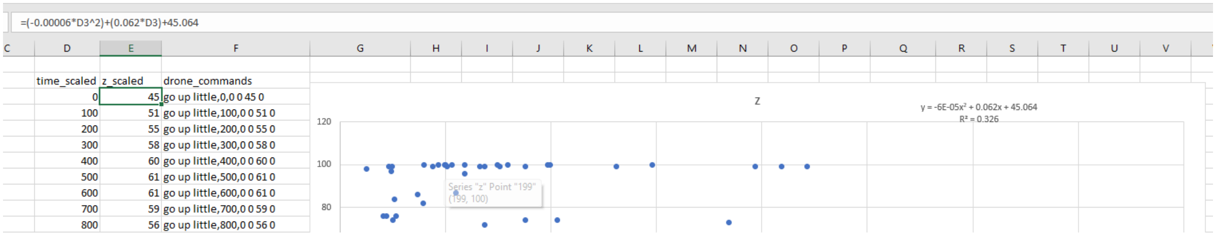


Figure 27. Command parameter calculation using polynomial distribution

User Satisfaction From

Comparative User Satisfaction Survey for Voice Operated vs. Manually Operated Drones

Please answer the following questions based on your experiences with both voice operated and manually operated drones. Your feedback is invaluable.

Section 1: Background Information

Before using the drones, did you have any prior experience with drone technology?

○ Yes ○ No

Have you received any formal training or instruction on operating drones?

○ Yes ○ No

Section 2: Experience with Manually Operated Drones

How long have you been using a manually operated drone?

○ I have not used one ○ Less than a month ○ 1-3 months ○ More than 3 months

What is the range of time that you were able to manually operate drones before?

○ I have not used one ○ Less than 10 minutes ○ Less than 1 hour but, more than 10 minutes ○ More than 1 hour

How would you rate your ease of learning to operate a manually operated drone?

○ Very easy ○ Somewhat easy ○ Neutral ○ Somewhat difficult ○ Very difficult

How confident do you feel in controlling a manually operated drone?

○ Very confident ○ Somewhat confident ○ Neutral ○ Somewhat unconfident ○ Very unconfident

Section 3: Experience with Voice-Operated Drones

How long have you been using a voice-operated drone?

○ I have not used one ○ Less than a month ○ 1-3 months ○ More than 3 months

How would you rate your ease of learning to operate a voice-operated drone?

○ Very easy ○ Somewhat easy ○ Neutral ○ Somewhat difficult ○ Very difficult

How confident do you feel in controlling a voice-operated drone?

○ Very confident ○ Somewhat confident ○ Neutral ○ Somewhat unconfident ○ Very unconfident

Figure 28. User Satisfaction Form part 1

Section 3: Experience with Voice-Operated Drones

How long have you been using a voice-operated drone?

- ○ I have not used one ○ Less than a month ○ 1-3 months ○ More than 3 months

How would you rate your ease of learning to operate a voice-operated drone?

- ○ Very easy ○ Somewhat easy ○ Neutral ○ Somewhat difficult ○ Very difficult

How confident do you feel in controlling a voice-operated drone?

- ○ Very confident ○ Somewhat confident ○ Neutral ○ Somewhat unconfident ○ Very unconfident

Section 4: Comparative Evaluation

Comparing your initial experiences, which type of drone did you find easier to learn to operate?

- ○ Voice-operated drone ○ Manually operated drone ○ No difference

Which type of drone do you feel gives you more precise control?

- ○ Voice-operated drone ○ Manually operated drone ○ They are about the same

How would you compare the enjoyment level between using a voice-operated drone and a manually operated drone?

- ○ I prefer voice-operated ○ I prefer manually operated ○ No preference

For beginners, which type of drone would you recommend based on your experience?

- ○ Voice-operated drone ○ Manually operated drone ○ No recommendation

Section 5: General Satisfaction and Feedback

Overall, how satisfied are you with using drones (both voice-operated and manually operated)?

- ○ Very satisfied ○ Somewhat satisfied ○ Neutral ○ Somewhat dissatisfied ○ Very dissatisfied

What challenges, if any, did you face while learning to operate each type of drone? Please specify:

Do you have any suggestions for making drones more beginner-friendly?

Section 6: Additional Comments

Please provide any additional comments or insights you have about your experience with voice operated and manually operated drones:

Submit

Figure 29. User Satisfaction Form part 2

Error trends by command

Go up command

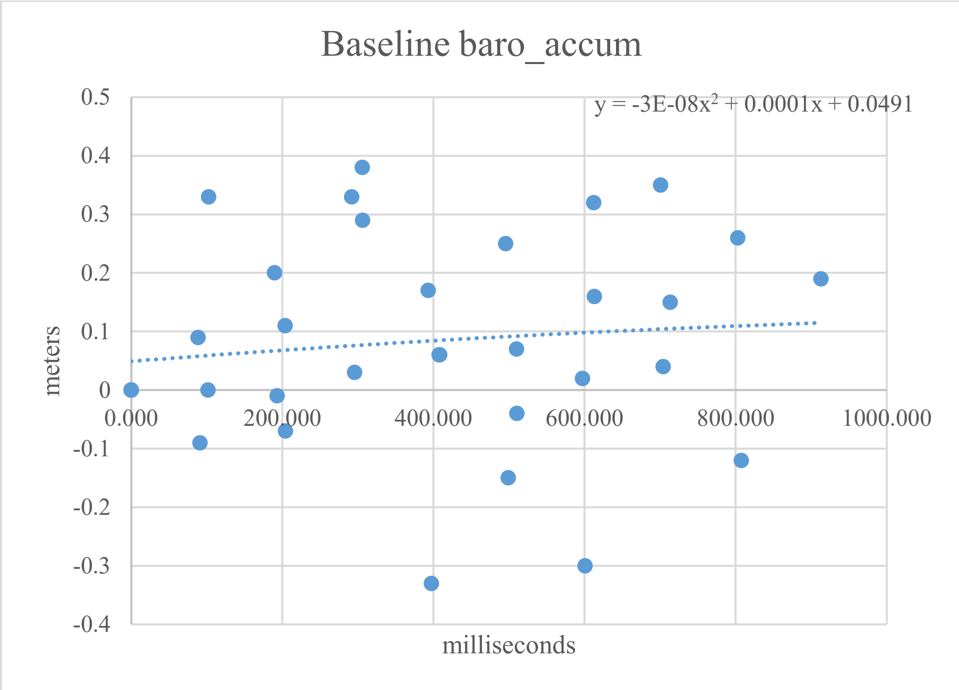


Figure 30. Go up baseline curve

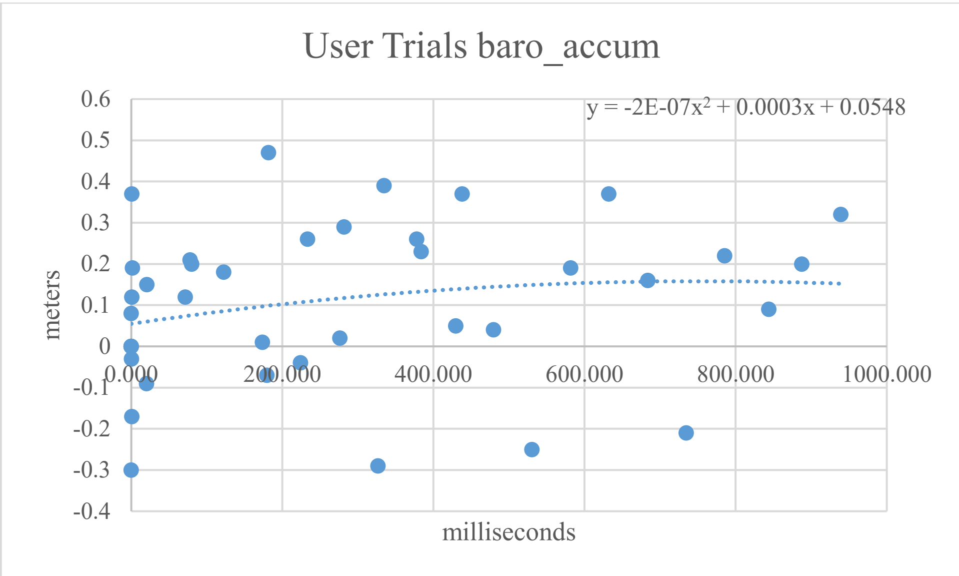
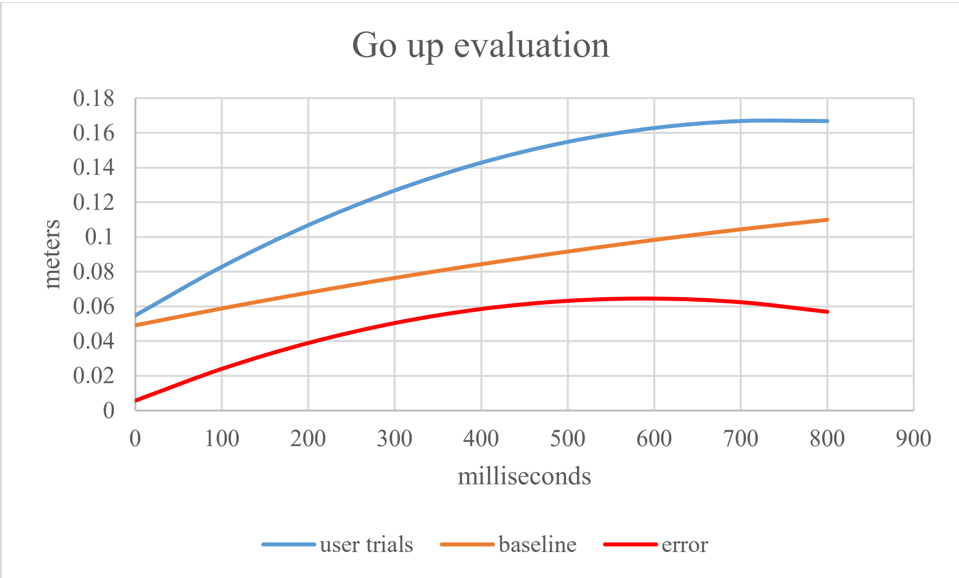


Figure 31. Go up user trial curve

Table 19. Go up scaled curve values

timeline	user trials	baseline	error
0	0.0548	0.0491	0.0057
100	0.0828	0.0588	0.024
200	0.1068	0.0679	0.0389
300	0.1268	0.0764	0.0504
400	0.1428	0.0843	0.0585
500	0.1548	0.0916	0.0632
600	0.1628	0.0983	0.0645
700	0.1668	0.1044	0.0624
800	0.1668	0.1099	0.0569

Table 20. Go up scaled curves



$b(y) = -3\times10^{-8}x^2 + 0.0001x + 0.0491$

$u(y) = -2\times10^{-7}x^2 + 0.0003x + 0.0548$

$f(y) = 1.7\times10^{-7}x^2 - 0.0002x - 0.0057$

Go up lot command

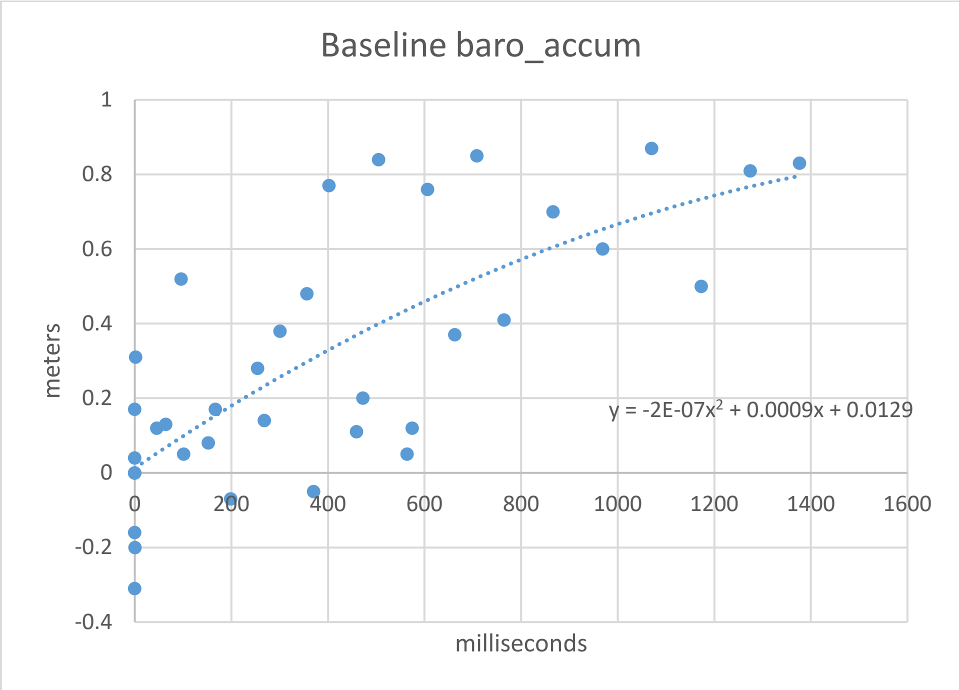


Figure 32. Go up lot baseline curve

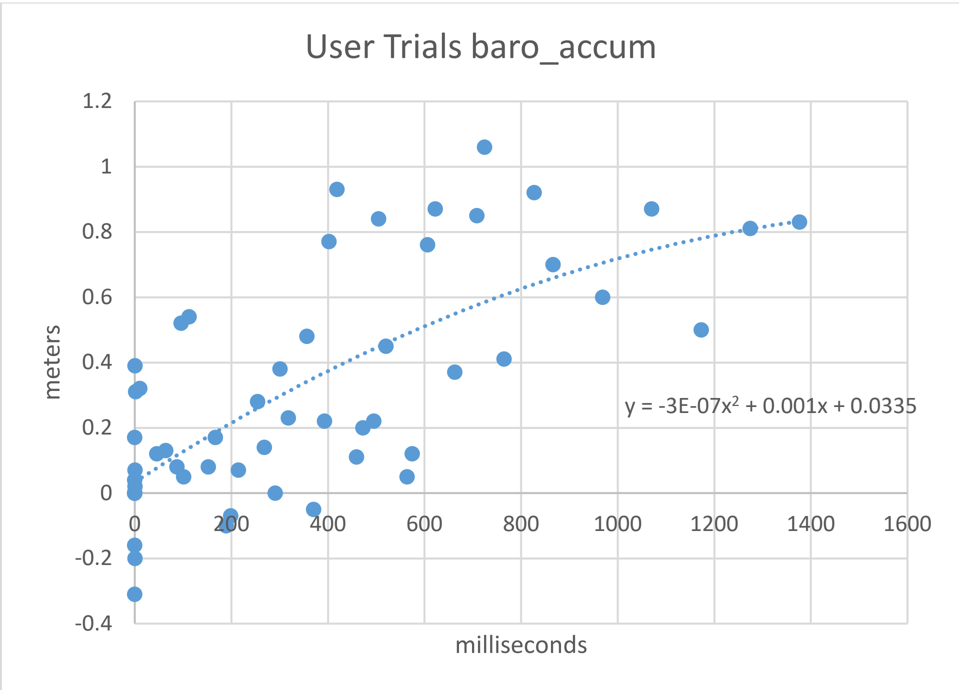


Figure 33. Go up lot user trial curve

Table 21. Go up lot scaled curve values

timeline	user trials	baseline	error
0	0.0129	0.0335	0.0206
100	0.1009	0.1305	0.0296
200	0.1849	0.2215	0.0366
300	0.2649	0.3065	0.0416
400	0.3409	0.3855	0.0446
500	0.4129	0.4585	0.0456
600	0.4809	0.5255	0.0446
700	0.5449	0.5865	0.0416

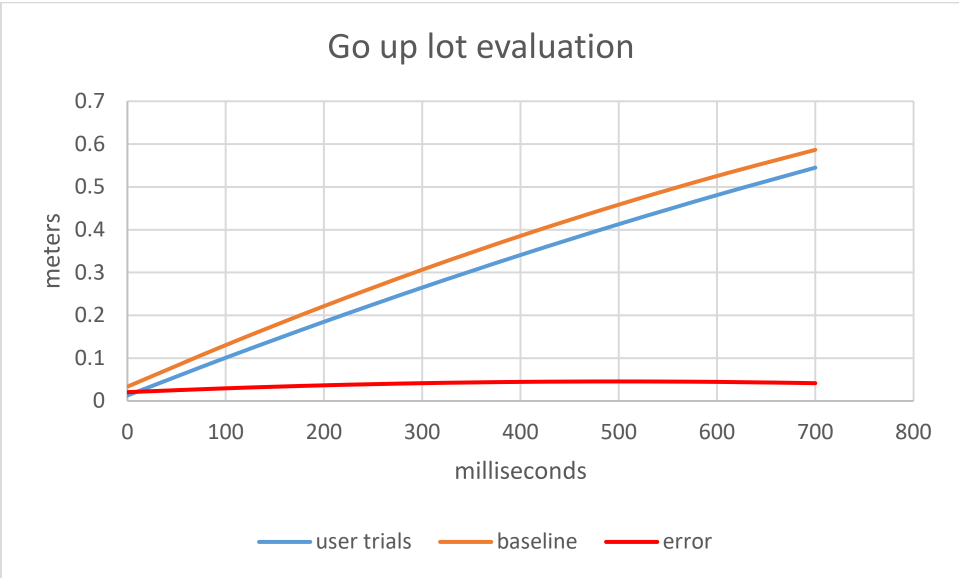


Figure 34. Go up lot scaled curves

$b(y) = -2\times10^{-7}x^2 + 0.0009x + 0.0129$

$u(y) = -3\times10^{-7}x^2 + 0.001x + 0.0335$

$f(y) = -1\times10^{-7}x^2 - 0.0001x - 0.0206$

Go down command

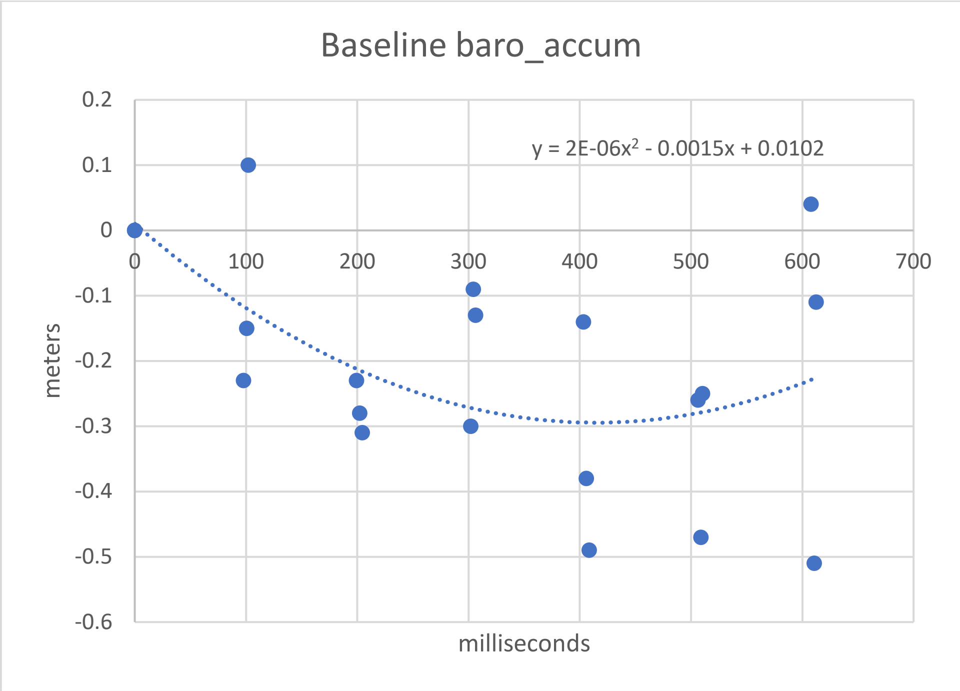


Figure 35 Go down baseline curve.

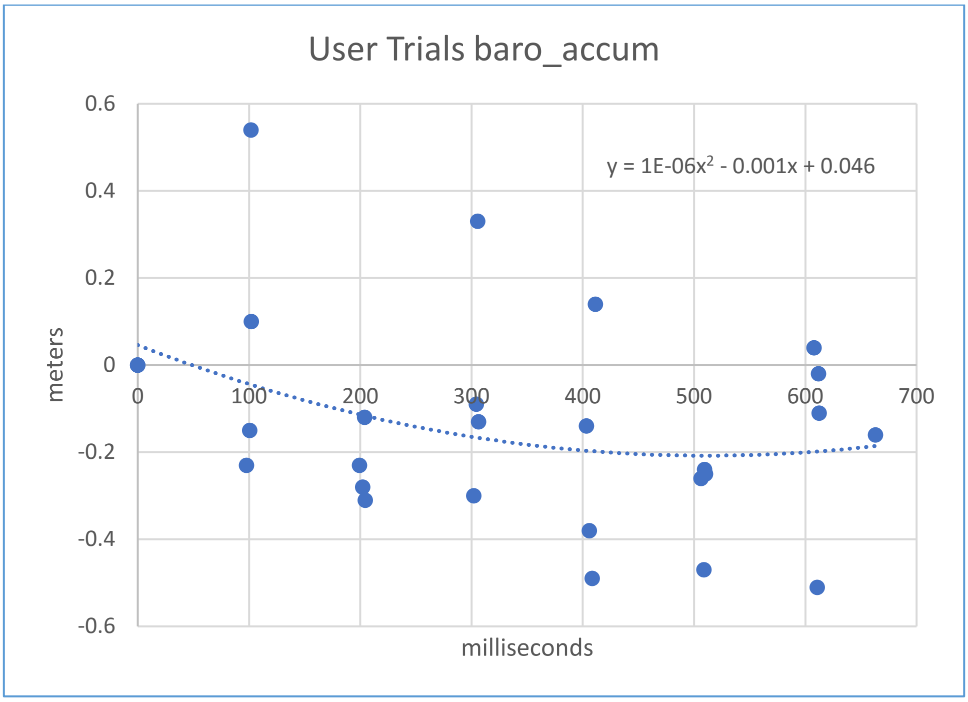


Figure 36. Go down user trial curve

Table 22. Go down scaled curve values

timeline	user trials	baseline	error
0	0.0102	0.046	0.0358
100	-0.1198	-0.044	-0.0042
200	-0.2098	-0.114	-0.0242
300	-0.2598	-0.164	-0.0242
400	-0.2698	-0.194	-0.0042
500	-0.2398	-0.204	0.0358
600	-0.1698	-0.194	0.0958
700	-0.0598	-0.164	0.1758

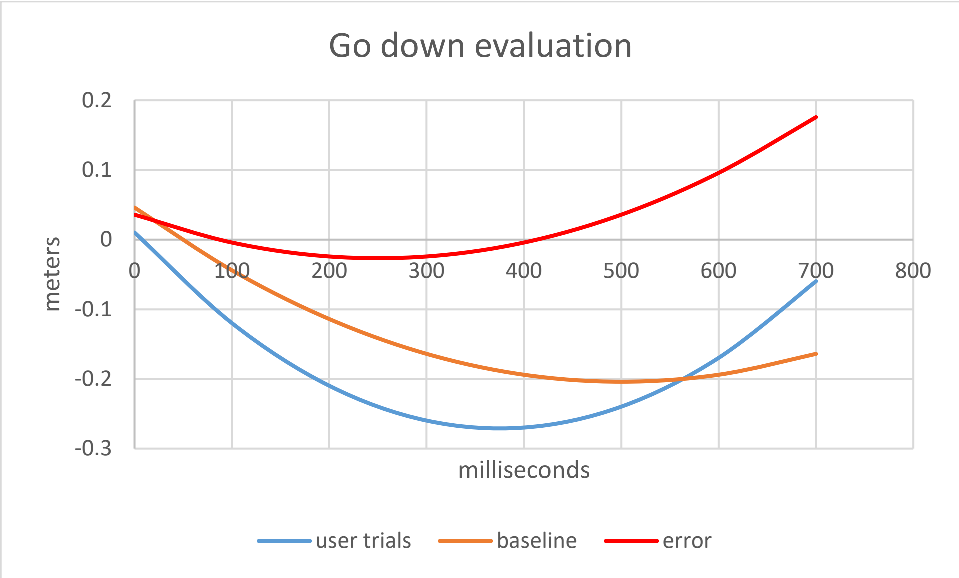


Figure 37. Go down scaled curves.

$b(y) = 2\times10^{-6}x^2 - 0.0015x + 0.0102$

$u(y) = 1\times10^{-6}x^2 - 0.001x + 0.046$

$f(y) = 1\times10^{-6}x^2 - 0.0005x - 0.0358$

Go down little command

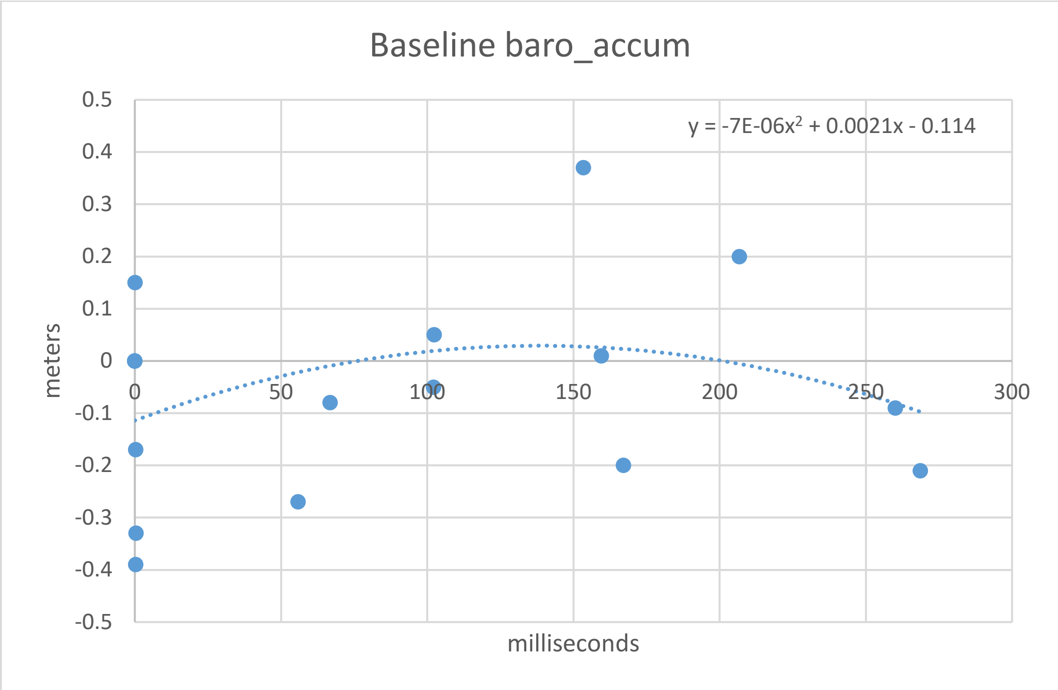


Figure 38. Go down little baseline curve

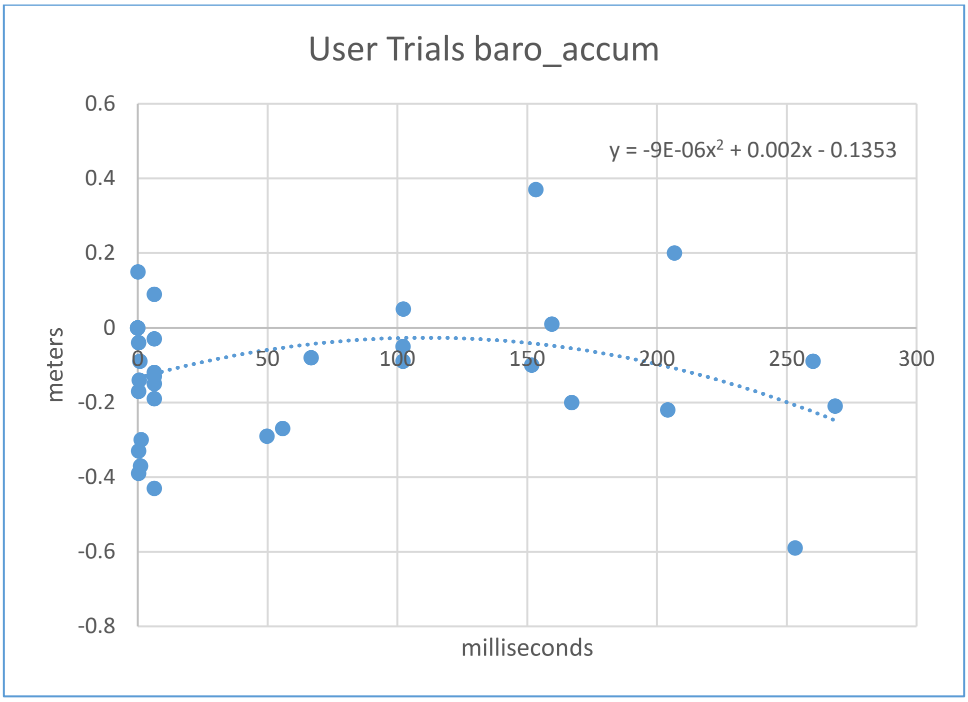


Figure 39. Go down little user trial curve

Table 23. Go down little scaled curve values

timeline	user trials	baseline	error
0	-0.1353	-0.114	0.0213
100	-0.0253	0.026	0.0513
200	-0.0953	0.026	0.1213
300	-0.3453	-0.114	0.2313
400	-0.7753	-0.394	0.3813
500	-1.3853	-0.814	0.5713
600	-2.1753	-1.374	0.8013
700	-3.1453	-2.074	1.0713

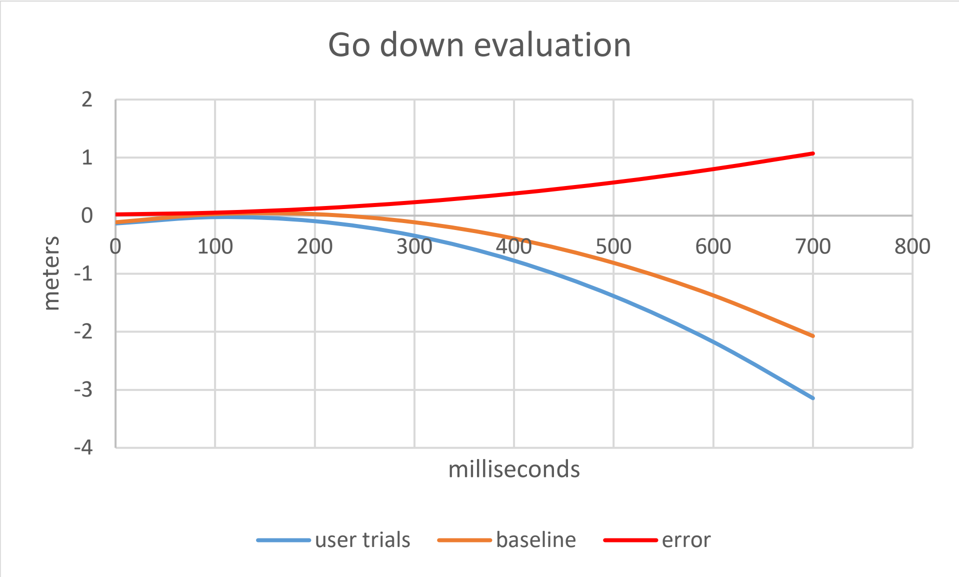


Figure 40. Go down little scaled curves

$b(y) = -7\times10^{-6}x^2 + 0.0021x - 0.114$

$u(y) = -9\times10^{-6}x^2 + 0.002x - 0.1353$

$f(y) = 2\times10^{-6}x2 + 0.0001x + 0.0213$

Go down lot command

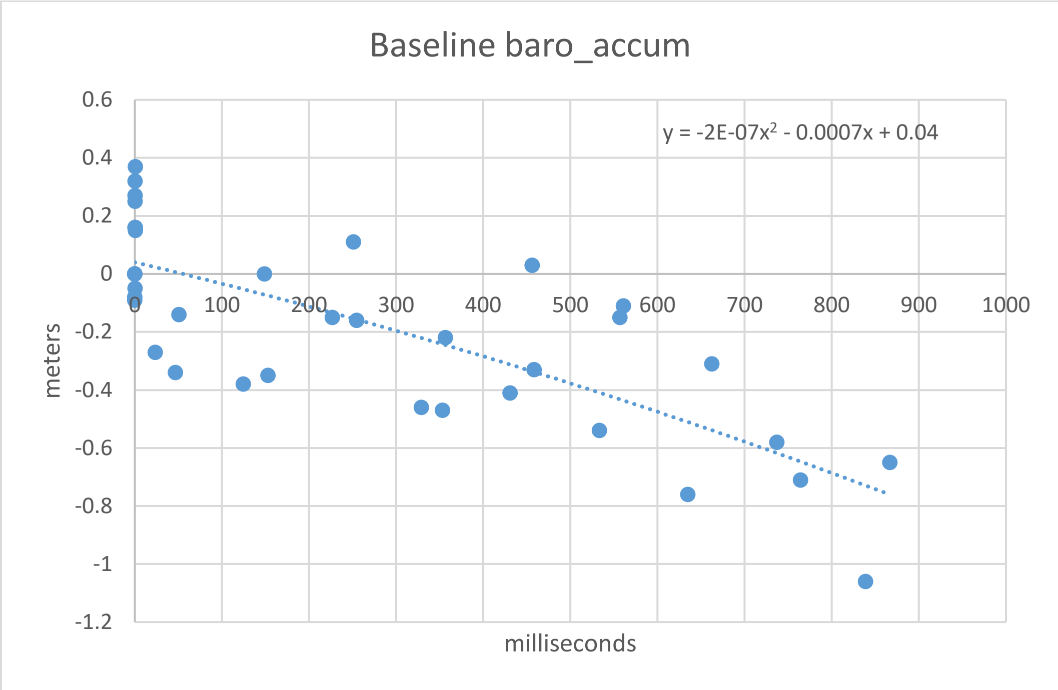


Figure 41. Go down lot baseline curve

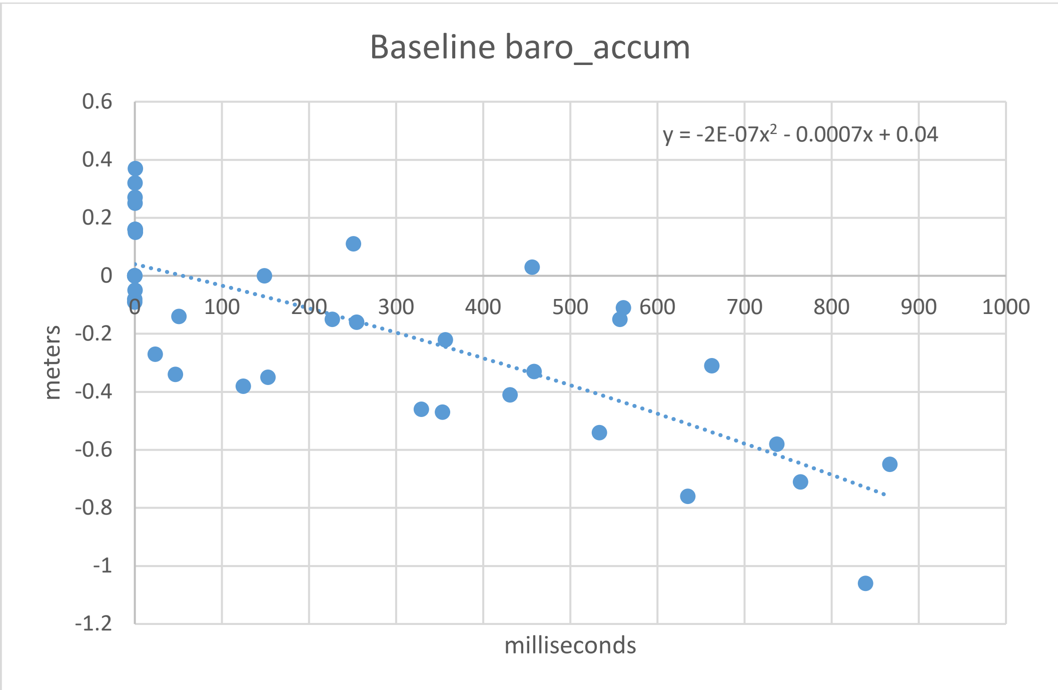


Figure 42. Go down lot user trial curve

Table 24. Go down lot scaled curve values

timeline	user trials	baseline	error
0	-0.0149	-0.04	0.0549
100	-0.1119	-0.112	0.0799
200	-0.2029	-0.188	0.0949
300	-0.2879	-0.268	0.0999
400	-0.3669	-0.352	0.0949
500	-0.4399	-0.44	0.0799
600	-0.5069	-0.532	0.0549
700	-0.5679	-0.628	0.0199

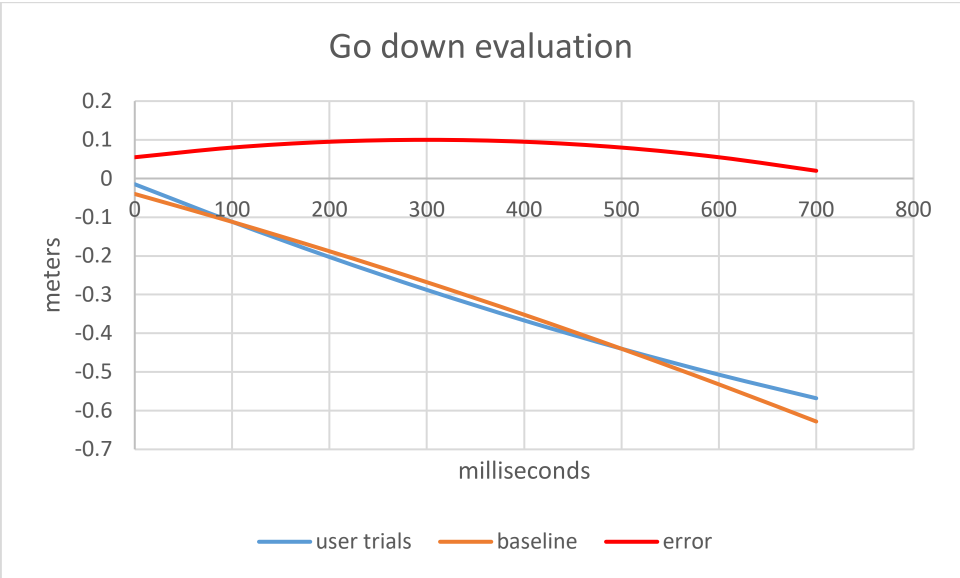


Figure 43. Go down lot scaled curves

$b(y) = -2\times10^{-7}x^2 - 0.0007x + 0.04$

$u(y) = -2\times10^{-7}x^2 - 0.0007x + 0.04$

$f(y) = -5\times10^{-7}x^2+0.0003x+0.0549$

Go up little command already mentioned in EVALUATION AND RESULTS chapter. Tables, Figures and formulas for the rest of the commands are published in file 'Error Trends by Command.docx' on the cloud folder. Click the link below

https://docs.google.com/document/d/1aCRXT1H7Vkz6g4BLzUy41OwUN7CMr8zy/edit?usp=drive_link&ouid=115278402482113003019&rtpof=true&sd=true

Figure 44. Error Trends by Command link

Error trends by command with noise

Tables, Charts and formulas for noisy condition environment is published on file ‘Error Trends by Command with noise.docx’ in shared cloud folder.

https://docs.google.com/document/d/16MmGXlYLoiWi0buf7-C9-tECGeAlwd8p/edit?usp=drive_link&ouid=115278402482113003019&rtpof=true&sd=true

Figure 45. Error Trends by Command noisy link

Error trends by command with wind

Tables, Charts and formulas for windy condition environment is published on file 'Error Trends by Command with wind.docx' in shared cloud folder.

https://docs.google.com/document/d/1uVostKq4plhfBG8BfDEpr_UMItkf01Li/edit?usp=drive_link&ouid=115278402482113003019&rtpof=true&sd=true

Figure 46. Error Trends by Command windy link

Command mapping

Table 25. Command Mapping File

VoiceCommand	Time	RCCommand
command	0	command
hover	0	rc 0 0 0 0
emergency	0	emergency
takeoff	0	takeoff
land	0	land
turn left	0	rc 0 0 0 0
turn left	100	rc 0 0 0 11
turn left	200	rc 0 0 0 62
turn left	300	rc 0 0 0 97
turn left	400	rc 0 0 0 99
turn left	500	rc 0 0 0 0
turn left little	0	rc 0 0 0 0
turn left little	100	rc 0 0 0 66
turn left little	200	rc 0 0 0 99
turn left little	300	rc 0 0 0 99
turn left little	310	rc 0 0 0 0
turn left lot	0	rc 0 0 0
turn left lot	100	rc 0 0 0 7
turn left lot	200	rc 0 0 0 80
turn left lot	300	rc 0 0 0 99
turn left lot	400	rc 0 0 0 99
turn left lot	500	rc 0 0 0 99
turn left lot	600	rc 0 0 0 99
turn left lot	700	rc 0 0 0 99
turn left lot	800	rc 0 0 0 99
turn left lot	900	rc 0 0 0 99
turn left lot	1000	rc 0 0 0 99
turn left lot	1100	rc 0 0 0 99
turn left lot	1200	rc 0 0 0 99
turn left lot	1300	rc 0 0 0 99
turn left lot	1310	rc 0 0 0 0
turn right	0	rc 0 0 0 0
turn right	100	rc 0 0 0 -92
turn right	200	rc 0 0 0 -100
turn right	300	rc 0 0 0 -100
turn right	400	rc 0 0 0 -100
turn right	410	rc 0 0 0 0
turn right little	0	rc 0 0 0 0
turn right little	100	rc 0 0 0 -32
turn right little	200	rc 0 0 0 -100

turn right little	300	rc 0 0 0 0
turn right lot	0	rc 0 0 0 0
turn right lot	100	rc 0 0 0 -52
turn right lot	200	rc 0 0 0 -100
turn right lot	300	rc 0 0 0 -100
turn right lot	400	rc 0 0 0 -100
turn right lot	500	rc 0 0 0 -100
turn right lot	600	rc 0 0 0 -100
turn right lot	700	rc 0 0 0 -100
turn right lot	800	rc 0 0 0 -100
turn right lot	900	rc 0 0 0 -100
turn right lot	1000	rc 0 0 0 -100
turn right lot	1100	rc 0 0 0 -100
turn right lot	1200	rc 0 0 0 -100
turn right lot	1300	rc 0 0 0 0
go back	0	rc 0 0 0 0
go back	100	rc 0 -28 0 0
go back	200	rc 0 -91 0 0
go back	300	rc 0 -99 0 0
go back	400	rc 0 -99 0 0
go back	500	rc 0 -99 0 0
go back	600	rc 0 -99 0 0
go back	700	rc 0 -99 0 0
go back	800	rc 0 -99 0 0
go back	900	rc 0 -99 0 0
go back	1000	rc 0 -99 0 0
go back	1100	rc 0 0 0 0
go back	1400	rc 0 -31 0 0
go back	1410	rc 0 0 0 0
go back little	0	rc 0 0 0 0
go back little	100	rc 0 -37 0 0
go back little	200	rc 0 -98 0 0
go back little	300	rc 0 -99 0 0
go back little	300	rc 0 0 0 0
go back little	1900	rc 0 29 0 0
go back little	2000	rc 0 0 0 0
go back lot	0	rc 0 0 0 0
go back lot	100	rc 0 -54 0 0
go back lot	200	rc 0 -99 0 0
go back lot	300	rc 0 -99 0 0
go back lot	400	rc 0 0 0 0
go back lot	4200	rc 0 -58 0 0
go back lot	4210	rc 0 0 0 0
go forward	0	rc 0 0 0 0
go forward	100	rc 0 5 0 0
go forward	200	rc 0 42 0 0

go forward	300	rc 0 68 0 0
go forward	400	rc 0 74 0 0
go forward	500	rc 0 75 0 0
go forward	600	rc 0 75 0 0
go forward	700	rc 0 74 0 0
go forward	800	rc 0 73 0 0
go forward	900	rc 0 73 0 0
go forward	900	rc 0 0 0 0
go forward	1500	rc 0 -28 0 0
go forward	1510	rc 0 0 0 0
go forward little	0	rc 0 0 0 0
go forward little	200	rc 0 85 0 0
go forward little	300	rc 0 77 0 0
go forward little	400	rc 0 0 0 0
go forward little	1800	rc 0 -8 0 0
go forward little	1810	rc 0 0 0 0
go forward lot	0	rc 0 0 0 0
go forward lot	100	rc 0 43 0 0
go forward lot	200	rc 0 100 0 0
go forward lot	300	rc 0 100 0 0
go forward lot	400	rc 0 100 0 0
go forward lot	500	rc 0 100 0 0
go forward lot	600	rc 0 100 0 0
go forward lot	700	rc 0 100 0 0
go forward lot	800	rc 0 100 0 0
go forward lot	900	rc 0 100 0 0
go forward lot	1000	rc 0 100 0 0
go forward lot	1000	rc 0 0 0 0
go forward lot	1200	rc 0 17 0 0
go forward lot	1300	rc 0 0 0 0
go right	0	rc 0 0 0 0
go right	100	rc 14 0 0 0
go right	200	rc 73 0 0 0
go right	300	rc 90 0 0 0
go right	400	rc 91 0 0 0
go right	410	rc 0 0 0 0
go right little	0	rc 0 0 0 0
go right little	100	rc 12 0 0 0
go right little	200	rc 84 0 0 0
go right little	300	rc 0 0 0 0
go right lot	0	rc 0 0 0 0
go right lot	100	rc 22 0 0 0
go right lot	200	rc 82 0 0 0
go right lot	300	rc 99 0 0 0
go right lot	400	rc 99 0 0 0
go right lot	500	rc 98 0 0 0

go right lot	600	rc 98 0 0 0
go right lot	700	rc 98 0 0 0
go right lot	800	rc 97 0 0 0
go right lot	900	rc 97 0 0 0
go right lot	1000	rc 97 0 0 0
go right lot	1100	rc 97 0 0 0
go right lot	1200	rc 97 0 0 0
go right lot	1300	rc 97 0 0 0
go right lot	1310	rc 0 0 0 0
go left	0	rc 0 0 0 0
go left	100	rc -20 0 0 0
go left	200	rc -99 0 0 0
go left	300	rc -99 0 0 0
go left	400	rc -99 0 0 0
go left	500	rc -99 0 0 0
go left	600	rc 72 0 0 0
go left	700	rc 99 0 0 0
go left	710	rc 0 0 0 0
go left little	0	rc 0 0 0 0
go left little	100	rc -45 0 0 0
go left little	200	rc -100 0 0 0
go left little	300	rc -100 0 0 0
go left little	300	rc 0 0 0 0
go left lot	0	rc 0 0 0 0
go left lot	600	rc -100 0 0 0
go left lot	700	rc -100 0 0 0
go left lot	800	rc -100 0 0 0
go left lot	900	rc -100 0 0 0
go left lot	1000	rc -100 0 0 0
go left lot	1100	rc -100 0 0 0
go left lot	1200	rc -100 0 0 0
go left lot	1300	rc 0 0 0 0
go up	0	rc 0 0 0 0
go up	900	rc 0 0 0 0
go up	100	rc 0 0 43 0
go up	200	rc 0 0 100 0
go up	300	rc 0 0 100 0
go up	400	rc 0 0 100 0
go up	500	rc 0 0 100 0
go up	600	rc 0 0 0 0
go up little	0	rc 0 0 0 0
go up little	100	rc 0 0 40 0
go up little	200	rc 0 0 100 0
go up little	300	rc 0 0 100 0
go up little	300	rc 0 0 0 0
go up little	310	rc 0 0 0 0

go up lot	0	rc 0 0 0 0
go up lot	0	rc 0 0 1 0
go up lot	100	rc 0 0 42 0
go up lot	200	rc 0 0 100 0
go up lot	300	rc 0 0 100 0
go up lot	400	rc 0 0 100 0
go up lot	500	rc 0 0 100 0
go up lot	600	rc 0 0 100 0
go up lot	700	rc 0 0 100 0
go up lot	800	rc 0 0 100 0
go up lot	900	rc 0 0 100 0
go up lot	1000	rc 0 0 100 0
go up lot	1100	rc 0 0 100 0
go up lot	1200	rc 0 0 100 0
go up lot	1300	rc 0 0 100 0
go up lot	1310	rc 0 0 0 0
go down	0	rc 0 0 0 0
go down	100	rc 0 0 -44 0
go down	200	rc 0 0 -95 0
go down	300	rc 0 0 -99 0
go down	400	rc 0 0 -99 0
go down	500	rc 0 0 -99 0
go down	600	rc 0 0 -99 0
go down	610	rc 0 0 0 0
go down little	0	rc 0 0 0 0
go down little	0	rc 0 0 -1 0
go down little	100	rc 0 0 -60 0
go down little	200	rc 0 0 -99 0
go down little	210	rc 0 0 0 0
go down lot	0	rc 0 0 0 0
go down lot	100	rc 0 0 -51 0
go down lot	200	rc 0 0 -97 0
go down lot	300	rc 0 0 -99 0
go down lot	400	rc 0 0 -99 0
go down lot	500	rc 0 0 -99 0
go down lot	600	rc 0 0 -99 0
go down lot	700	rc 0 0 -99 0
go down lot	800	rc 0 0 -99 0
go down lot	900	rc 0 0 -99 0
go down lot	1000	rc 0 0 -99 0
go down lot	1100	rc 0 0 -99 0
go down lot	1200	rc 0 0 -99 0
go down lot	1300	rc 0 0 -99 0
go down lot	1400	rc 0 0 -99 0
go down lot	1500	rc 0 0 -99 0
go down lot	1600	rc 0 0 -99 0

go down lot	1700	rc 0 0 -99 0
go down lot	1800	rc 0 0 0 0

Dictionary

Table 26. Dictionary data

Voice command	Drone Command
Go up	GO UP
Fly up	GO UP
Move up	GO UP
Go up little	GO UP Little
Go up a little	GO UP Little
Go up little bit	GO UP Little
Fly up little	GO UP Little
Fly up a little	GO UP Little
Fly up little bit	GO UP Little
Move up little	GO UP Little
Move up a little	GO UP Little
Move up little bit	GO UP Little
Go up lot	GO UP Lot
go up a lot	GO UP Lot
Fly up lot	GO UP Lot
Fly up a lot	GO UP Lot
Move up lot	GO UP Lot
Move up a lot	GO UP Lot
Go down	GO Down
Fly down	GO Down
Move down	GO Down
Go down little	GO Down Little
Go down a little	GO Down Little
Go down little bit	GO Down Little
Fly down little	GO Down Little
Fly down a little	GO Down Little
Fly down little bit	GO Down Little
Move down little	GO Down Little
Move down a little	GO Down Little
Move down little bit	GO Down Little
Go down lot	GO Down Lot
Go down a lot	GO Down Lot
Fly down lot	GO Down Lot
Fly down a lot	GO Down Lot
Move down lot	GO Down Lot
Move down a lot	GO Down Lot
Go Left	GO Left
Fly Left	GO Left
Move Left	GO Left
Go Left little	GO Left Little
Go Left a little	GO Left Little
Go Left little bit	GO Left Little

Fly Left little	GO Left Little
Fly Left a little	GO Left Little
Fly Left little bit	GO Left Little
Move Left little	GO Left Little
Move Left a little	GO Left Little
Move Left little bit	GO Left Little
Go Left lot	GO Left Lot
go Left a lot	GO Left Lot
Fly Left lot	GO Left Lot
Fly Left a lot	GO Left Lot
Move Left lot	GO Left Lot
Move Left a lot	GO Left Lot
Go Right	GO Right
Fly Right	GO Right
Move Right	GO Right
Go Right little	GO Right Little
Go Right a little	GO Right Little
Go Right little bit	GO Right Little
Fly Right little	GO Right Little
Fly Right a little	GO Right Little
Fly Right little bit	GO Right Little
Move Right little	GO Right Little
Move Right a little	GO Right Little
Move Right little bit	GO Right Little
Go Right lot	GO Right Lot
go Right a lot	GO Right Lot
Fly Right lot	GO Right Lot
Fly Right a lot	GO Right Lot
Move Right lot	GO Right Lot
Move Right a lot	GO Right Lot
Go Forward	GO Forward
Fly Forward	GO Forward
Move Forward	GO Forward
Go Forward little	GO Forward Little
Go Forward a little	GO Forward Little
Go Forward little bit	GO Forward Little
Fly Forward little	GO Forward Little
Fly Forward a little	GO Forward Little
Fly Forward little bit	GO Forward Little
Move Forward little	GO Forward Little
Move Forward a little	GO Forward Little
Move Forward little bit	GO Forward Little
Go Forward lot	GO Forward Lot
go Forward a lot	GO Forward Lot
Fly Forward lot	GO Forward Lot
Fly Forward a lot	GO Forward Lot

Move Forward lot	GO Forward Lot
Move Forward a lot	GO Forward Lot
Go Front	GO Forward
Fly Front	GO Forward
Move Front	GO Forward
Go Front little	GO Forward Little
Go Front a little	GO Forward Little
Go Front little bit	GO Forward Little
Fly Front little	GO Forward Little
Fly Front a little	GO Forward Little
Fly Front little bit	GO Forward Little
Move Front little	GO Forward Little
Move Front a little	GO Forward Little
Move Front little bit	GO Forward Little
Go Front lot	GO Forward Lot
go Front a lot	GO Forward Lot
Fly Front lot	GO Forward Lot
Fly Front a lot	GO Forward Lot
Move Front lot	GO Forward Lot
Move Front a lot	GO Forward Lot
Go Back	GO Back
Fly Back	GO Back
Move Back	GO Back
Go Back little	GO Back Little
Go Back a little	GO Back Little
Go Back little bit	GO Back Little
Fly Back little	GO Back Little
Fly Back a little	GO Back Little
Fly Back little bit	GO Back Little
Move Back little	GO Back Little
Move Back a little	GO Back Little
Move Back little bit	GO Back Little
Go Back lot	GO Back Lot
go Back a lot	GO Back Lot
Fly Back lot	GO Back Lot
Fly Back a lot	GO Back Lot
Move Back lot	GO Back Lot
Move Back a lot	GO Back Lot
Go Backward	GO Back
Fly Backward	GO Back
Move Backward	GO Back
Go Backward little	GO Back Little
Go Backward a little	GO Back Little
Go Backward little bit	GO Back Little
Fly Backward little	GO Back Little
Fly Backward a little	GO Back Little

Fly Backward little bit	GO Back Little
Move Backward little	GO Back Little
Move Backward a little	GO Back Little
Move Backward little bit	GO Back Little
Go Backward lot	GO Back Lot
go Backward a lot	GO Back Lot
Fly Backward lot	GO Back Lot
Fly Backward a lot	GO Back Lot
Move Backward lot	GO Back Lot
Move Backward a lot	GO Back Lot
Go Reverse	GO Back
Fly Reverse	GO Back
Move Reverse	GO Back
Go Reverse little	GO Back Little
Go Reverse a little	GO Back Little
Go Reverse little bit	GO Back Little
Fly Reverse little	GO Back Little
Fly Reverse a little	GO Back Little
Fly Reverse little bit	GO Back Little
Move Reverse little	GO Back Little
Move Reverse a little	GO Back Little
Move Reverse little bit	GO Back Little
Go Reverse lot	GO Back Lot
go Reverse a lot	GO Back Lot
Fly Reverse lot	GO Back Lot
Fly Reverse a lot	GO Back Lot
Move Reverse lot	GO Back Lot
Move Reverse a lot	GO Back Lot
Turn left	Turn left
Turn left little	Turn left little
Turn left a little	Turn left little
Rotate left	Turn left
Rotate left little	Turn left little
Rotate left a little	Turn left little
Turn left lot	Turn left lot
Turn left a lot	Turn left lot
Rotate left lot	Turn left lot
Rotate left a lot	Turn left lot
Turn right	Turn right
Turn right little	Turn right little
Turn right a little	Turn right little
Rotate right	Turn right
Rotate right little	Turn right little
Protect right a little	Turn right little
Turn right lot	Turn right lot
Turn right a lot	Turn right lot

Rotate right lot	Turn right lot
Rotate right a lot	Turn right lot
Hover	Hover
Stop	Hover
Hold	Hover
Wait	Hover
Pause	Hover
Emergency	Emergency
Shutdown	Emergency
Mayday	Emergency
Mayday Command	Emergency
Take off	takeoff
Land	land
Command	command
Connect	command

Log Files

Name	Date modified	Type	Size
Pilot_Test_0306_212501_rc_0_0_25_0_up.xlsx	3/13/2024 11:10 PM	Microsoft Excel W...	61 KB
Delays.xlsx	3/13/2024 11:10 PM	Microsoft Excel W...	63 KB
Delays.txt	3/13/2024 10:39 PM	TXT File	64 KB
Pilot_Test_0306_212340_rc_-15_0_0_0_left.xlsx	3/13/2024 10:34 PM	Microsoft Excel W...	47 KB
Pilot_Test_0306_212501_rc_0_0_25_0_up.csv	3/13/2024 10:01 PM	Microsoft Excel C...	44 KB
Test User 1_0313_150646_Connect .csv	3/13/2024 9:22 PM	Microsoft Excel C...	103 KB
Test User 1_0313_160730_command.csv	3/13/2024 4:07 PM	Microsoft Excel C...	0 KB
Test User 1_0313_160730_command.txt	3/13/2024 4:07 PM	TXT File	1 KB
Counters.txt	3/13/2024 3:10 PM	TXT File	13 KB
Test User 1_0313_151002_command.txt	3/13/2024 3:10 PM	TXT File	1 KB
Test User 1_0313_150836_command.txt	3/13/2024 3:08 PM	TXT File	1 KB
Test User 1_0313_150335_Connect .csv	3/13/2024 3:06 PM	Microsoft Excel C...	132 KB
Test User 3_0313_145820_Connect .csv	3/13/2024 3:03 PM	Microsoft Excel C...	276 KB
Test User 1_0313_145359_Connect .csv	3/13/2024 2:57 PM	Microsoft Excel C...	172 KB
Test User 1_0313_144819_Connect .csv	3/13/2024 2:53 PM	Microsoft Excel C...	264 KB
Test User 1_0313_145159_emergency.txt	3/13/2024 2:51 PM	TXT File	1 KB
Test User 1_0313_145004_emergency.txt	3/13/2024 2:50 PM	TXT File	1 KB
Test User 1_0313_145005_emergency.txt	3/13/2024 2:50 PM	TXT File	1 KB

Figure 47. Voice application log directory

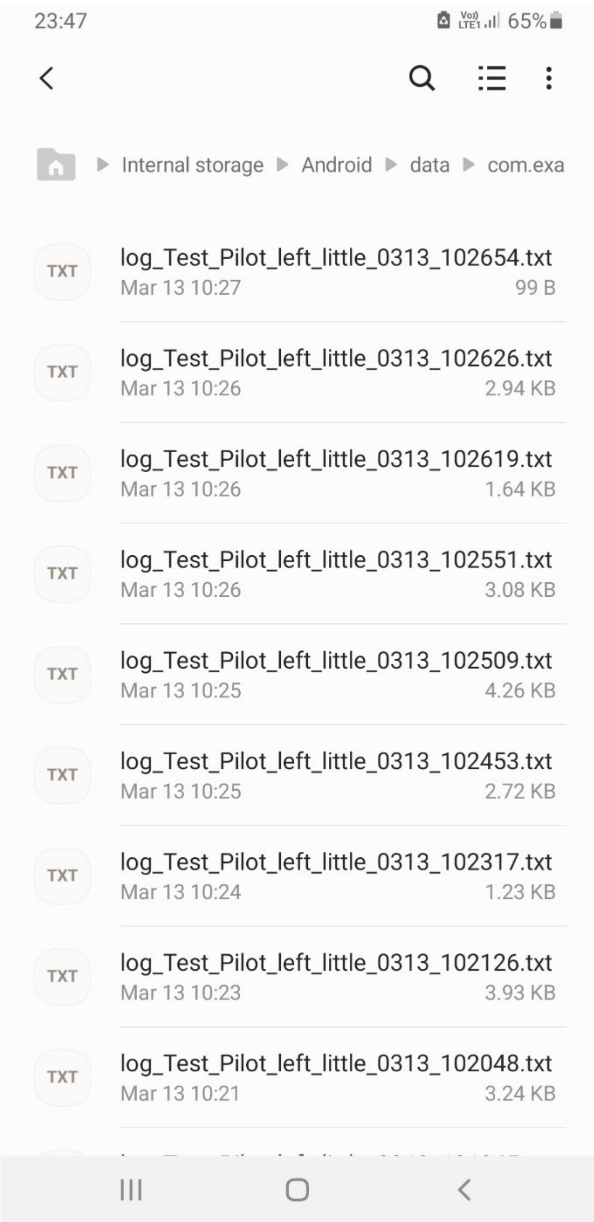


Figure 48. Android application log directory

```
"24/03/07 01:20:15.835,Text delay,8768"
"24/03/07 01:20:15.837,Command delay,28905"
"24/03/07 01:20:45.710,Text delay,18274"
"24/03/07 01:20:45.715,Command delay,72434"
"24/03/07 11:15:24.150,Text delay,1301850"
"24/03/07 11:15:24.152,Command delay,1323884"
"24/03/07 11:15:33.980,Text delay,7068"
"24/03/07 11:15:33.982,Command delay,22790"
"24/03/07 11:16:44.525,Text delay,4810"
"24/03/07 11:16:44.526,Command delay,10956"
"24/03/09 11:13:35.817,Text delay,448114"
"24/03/09 11:13:41.157,Command delay,53853435"
"24/03/09 11:13:42.572,Text delay,5818"
"24/03/09 11:13:46.252,Command delay,36806595"
"24/03/09 11:13:57.174,Text delay,3843"
"24/03/09 11:14:08.618,Command delay,114451048"
"24/03/09 11:14:10.535,Text delay,7780"
"24/03/09 11:14:22.566,Command delay,120311227"
"24/03/09 11:14:32.113,Text delay,464491"
"24/03/09 11:14:43.251,Command delay,111850113"
"24/03/09 11:14:45.956,Text delay,5083"
"24/03/09 11:14:47.991,Command delay,20346979"
"24/03/09 11:14:58.805,Text delay,8155"
"24/03/09 11:18:49.534,Text delay,422089"
"24/03/09 11:19:07.661,Command delay,181690212"
"24/03/09 11:19:09.473,Text delay,10969"
"24/03/09 11:19:22.086,Command delay,126145207"
"24/03/09 11:19:33.099,Text delay,24471"
"24/03/09 11:19:36.751,Command delay,36544229"
"24/03/09 11:19:43.364,Text delay,5559"
"24/03/09 11:19:48.353,Command delay,49895785"
```

Figure 49. Delays log file

```
11:50:38.807,Correct,
11:50:44.424,Correct,
11:50:50.285,Correct,
11:51:09.520,Incorrect,Drone behaviour mismatch the command
11:51:21.298,Correct,
11:53:48.659,Incorrect,Speech to text invalid result
11:54:33.005,Incorrect,Drone behaviour mismatch the command
11:54:43.821,Incorrect,Drone behaviour mismatch the command
11:54:52.384,Correct,
11:55:28.340,Incorrect,Drone behaviour mismatch the command
11:55:53.107,Incorrect,Speech to text invalid result
11:56:04.474,Correct,
11:56:15.347,Correct,
11:56:22.458,Correct,
11:56:31.349,Incorrect,Speech to text invalid result
11:56:49.949,Incorrect,Speech to text invalid result
12:01:02.164,Correct,
12:01:04.438,Incorrect,Speech to text invalid result
12:01:05.231,Incorrect,Speech to text invalid result
12:01:15.656,Incorrect,Speech to text invalid result
12:01:23.536,Incorrect,Speech to text invalid result
12:01:37.541,Correct,
12:01:55.169,Incorrect,Speech to text no result
12:02:04.214,Incorrect,Speech to text no result
12:02:09.965,Correct,
12:02:16.947,Incorrect,Speech to text no result
12:02:27.707,Incorrect,Speech to text no result
12:02:40.823,Incorrect,Speech to text no result
12:02:54.412,Incorrect,Speech to text no result
12:03:00.819,Incorrect,Speech to text no result
12:03:21.507,Incorrect,Speech to text partial  result
12:03:23.325,Incorrect,Speech to text partial  result
12:03:31.999,Incorrect,Speech to text no result
12:03:38.006,Correct,
12:03:44.127,Correct,
12:03:48.735,Correct,
12:03:54.971,Correct,
12:04:05.440,Incorrect,Speech to text partial  result
12:04:25.318,Incorrect,Drone behaviour mismatch the command
12:04:37.125,Incorrect,Speech to text no result
13:38:23.497,Incorrect,Speech to text no result
```

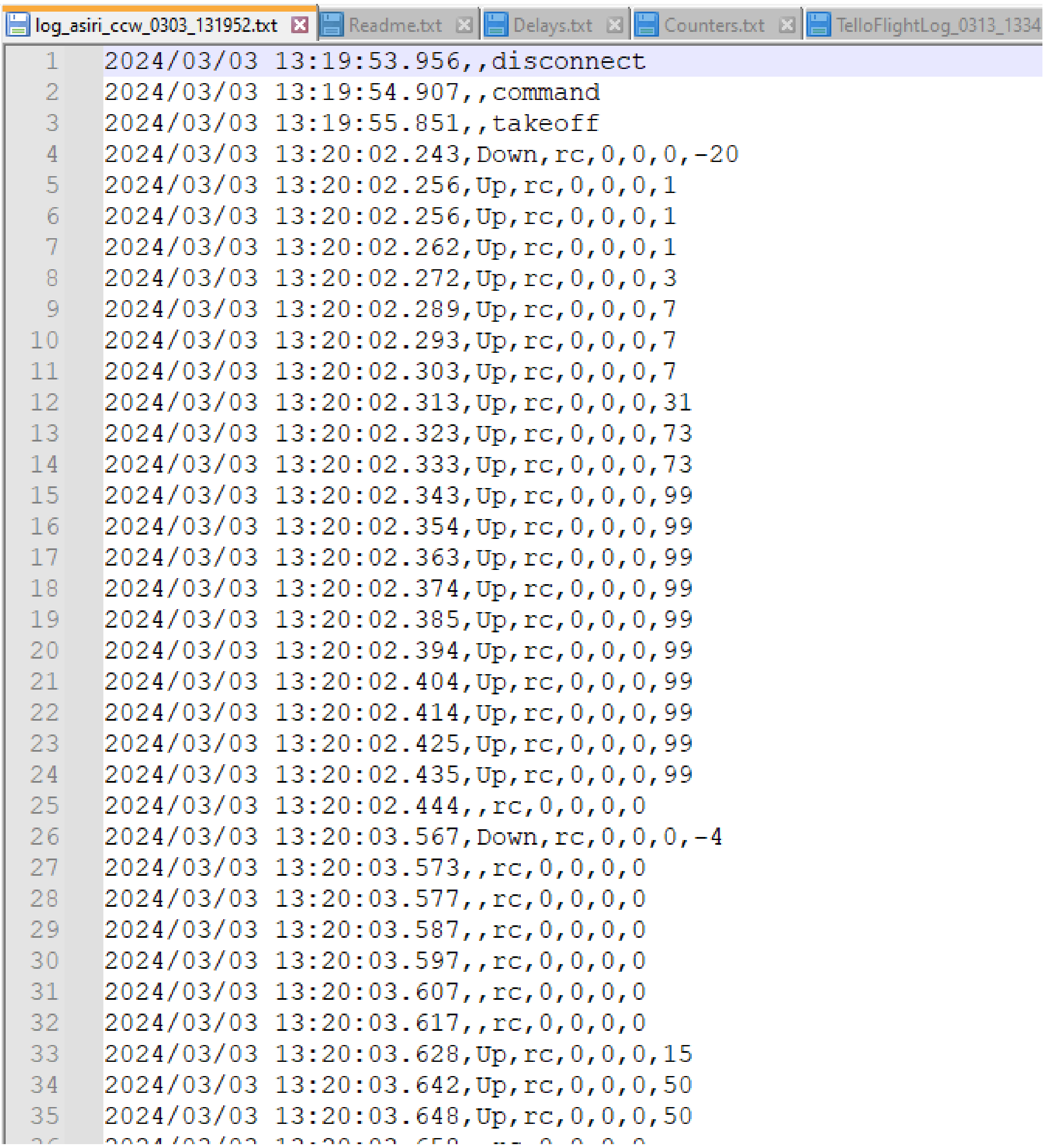
Figure 50. Counters log file

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1				Predicted		Actual										
2	time	result	reason	1	0	1	0									
3	11:27.44.477	Correct		1		1		T	P	TP						
4	11:28.03.610	Correct		1		1		T	P	TP						
5	11:28.09.924	Correct		1		1		T	P	TP						
6	11:28.10.671	Correct		1		1		T	P	TP						
7	11:28.13.872	Correct		1		1		T	P	TP				Actual		
8	11:28.19.460	Correct		1		1		T	P	TP				Positive	Negative	
9	11:28.34.455	Incorrect	Speech to text invalid result	1			1	T	N	TN		Predicted	Positive	2743	0	
10	11:28.35.158	Incorrect	Speech to text invalid result	1			1	T	N	TN			Negative	10	81	
11	11:28.43.350	Correct		1		1		T	P	TP						
12	11:28.49.390	Correct		1		1		T	P	TP						
13	11:28.54.516	Correct		1		1		T	P	TP						
14	11:28.58.650	Correct		1		1		T	P	TP			**Recall**	0.996368		
15	11:29.02.849	Correct		1		1		T	P	TP						
16	11:29.06.817	Correct		1		1		T	P	TP						
17	11:29.11.690	Correct		1		1		T	P	TP			**Precision**	1		
18	11:29.17.559	Correct		1		1		T	P	TP						
19	11:29.21.056	Correct		1		1		T	P	TP						
20	11:29.24.985	Correct		1		1		T	P	TP			**Accuracy**	0.996471		
21	11:29.29.200	Correct		1		1		T	P	TP						
22	11:29.33.509	Correct		1		1		T	P	TP						
23	11:29.38.515	Correct		1		1		T	P	TP			**F-measure**	0.99818		
24	11:30.01.580	Incorrect	Speech to text invalid result	1			1	T	N	TN						
25	11:30.13.105	Incorrect	Speech to text previous voice input partially used		1		1	F	N	FN						
26	11:30.21.610	Correct		1		1		T	P	TP						
27	11:30.28.271	Correct		1		1		T	P	TP						
28	11:30.45.289	Incorrect	Speech to text partial result	1			1	T	N	TN						

Figure 51. Counters Confusion Matrix calculation

1	Pilot	timestamp_ms	command	pitch	roll	yaw	vgx	vgy	vgz	templ	temph	tof	h	bat	baro	time	agx	agy	agz
1243	Test User 1	124812.4361	go right	0	0	1	0	0	0	85	88	181	140	20	29.16	416	9	-14	-931
1244	Test User 1	124914.3902	go right	0	0	1	0	0	0	85	88	181	140	20	29.04	416	-4	-8	-957
1245	Test User 1	125016.521	go right	0	0	1	0	0	0	85	88	181	140	20	29.34	416	-7	4	-967
1246	Test User 1	125119.2828	go right	0	0	1	0	0	0	85	88	180	140	20	29.46	416	73	-2	-947
1247	Test User 1	125220.8177	go right	0	0	1	0	0	0	85	88	179	140	20	29.53	416	-26	24	-1019
1248	Test User 1	125322.3338	go right	0	0	1	0	0	1	85	88	177	140	20	29.36	416	-39	-5	-965
1249	Test User 1	125424.3451	go right	0	0	1	0	0	1	85	88	175	140	20	29.43	416	-55	-12	-1026
1250	Test User 1	125526.4051	go right	0	0	1	0	0	0	85	88	174	140	20	29.38	417	35	-19	-973
1251	Test User 1	125628.4135	go right	0	0	1	0	0	1	85	88	173	140	20	29.22	417	-30	16	-1006
1252	Test User 1	125730.6642	go right	0	0	1	0	0	0	85	88	171	140	20	29.15	417	23	19	-950
1253	Test User 1	125832.1871	go right	0	0	1	0	0	1	85	88	169	140	20	29.32	417	-51	0	-1011
1254	Test User 1	125934.0479	go right	0	0	1	0	0	0	85	88	167	140	20	29.01	417	-15	0	-966
1255	Test User 1	126036.0892	go right	0	0	1	0	0	0	85	87	165	140	20	29.16	417	49	-43	-1026
1256	Test User 1	126138.4467	go right	0	0	1	0	0	0	85	87	163	140	20	29	417	-32	-5	-993
1257	Test User 1	126240.2195	go right	0	0	1	0	0	0	85	87	162	130	20	29.06	417	-7	8	-1041
1258	Test User 1	126341.954	go right	0	0	1	0	0	0	85	87	160	130	20	29.07	417	57	-13	-971
1259	Test User 1	126445.6672	go forward	0	0	1	0	0	0	85	87	159	130	20	29.06	417	-43	6	-1007
1260	Test User 1	126546.2517	go forward	0	0	1	0	0	0	85	87	157	130	20	29.09	418	-12	-12	-971
1261	Test User 1	126648.1694	go forward	0	0	1	0	0	0	85	87	155	130	19	28.93	418	57	0	-903
1262	Test User 1	126750.2444	go forward	0	0	1	0	0	0	85	87	153	130	19	28.96	418	8	4	-971
1263	Test User 1	126852.2349	go forward	-1	0	1	0	0	0	85	87	152	130	19	29.01	418	-22	20	-1000
1264	Test User 1	126954.2707	go forward	-4	0	1	0	0	1	85	87	149	130	19	29.34	418	-23	1	-1108
1265	Test User 1	127056.1201	go forward	-7	0	1	1	0	0	85	87	149	130	19	28.77	418	14	37	-950
1266	Test User 1	127158.1444	go forward	-7	0	1	2	0	0	85	87	148	120	19	29.38	418	-8	0	-1009
1267	Test User 1	127263.6484	go forward	-7	0	1	3	0	1	85	87	147	120	19	29.31	418	24	-49	-1022
1268	Test User 1	127362.0201	go forward	-6	0	1	5	0	0	85	87	147	120	19	28.99	418	-18	31	-974
1269	Test User 1	127464.1231	go forward	-7	0	1	5	0	0	85	87	148	120	19	29.03	418	-8	22	-1055

Figure 52. Diagnostic data log file


log_asiri_ccw_0303_131952.txt | Readme.txt | Delays.txt | Counters.txt | TelloFlightLog_0313_1334

```
2024/03/03 13:19:53.956,,disconnect
2024/03/03 13:19:54.907,,command
2024/03/03 13:19:55.851,,takeoff
2024/03/03 13:20:02.243,Down,rc,0,0,0,-20
2024/03/03 13:20:02.256,Up,rc,0,0,0,1
2024/03/03 13:20:02.256,Up,rc,0,0,0,1
2024/03/03 13:20:02.262,Up,rc,0,0,0,1
2024/03/03 13:20:02.272,Up,rc,0,0,0,3
2024/03/03 13:20:02.289,Up,rc,0,0,0,7
2024/03/03 13:20:02.293,Up,rc,0,0,0,7
2024/03/03 13:20:02.303,Up,rc,0,0,0,7
2024/03/03 13:20:02.313,Up,rc,0,0,0,31
2024/03/03 13:20:02.323,Up,rc,0,0,0,73
2024/03/03 13:20:02.333,Up,rc,0,0,0,73
2024/03/03 13:20:02.343,Up,rc,0,0,0,99
2024/03/03 13:20:02.354,Up,rc,0,0,0,99
2024/03/03 13:20:02.363,Up,rc,0,0,0,99
2024/03/03 13:20:02.374,Up,rc,0,0,0,99
2024/03/03 13:20:02.385,Up,rc,0,0,0,99
2024/03/03 13:20:02.394,Up,rc,0,0,0,99
2024/03/03 13:20:02.404,Up,rc,0,0,0,99
2024/03/03 13:20:02.414,Up,rc,0,0,0,99
2024/03/03 13:20:02.425,Up,rc,0,0,0,99
2024/03/03 13:20:02.435,Up,rc,0,0,0,99
2024/03/03 13:20:02.444,,rc,0,0,0,0
2024/03/03 13:20:03.567,Down,rc,0,0,0,-4
2024/03/03 13:20:03.573,,rc,0,0,0,0
2024/03/03 13:20:03.577,,rc,0,0,0,0
2024/03/03 13:20:03.587,,rc,0,0,0,0
2024/03/03 13:20:03.597,,rc,0,0,0,0
2024/03/03 13:20:03.607,,rc,0,0,0,0
2024/03/03 13:20:03.617,,rc,0,0,0,0
2024/03/03 13:20:03.628,Up,rc,0,0,0,15
2024/03/03 13:20:03.642,Up,rc,0,0,0,50
2024/03/03 13:20:03.648,Up,rc,0,0,0,50
```

Figure 53. Android app command log file

Android Studio Project

Source code is in the GitHub repository https://github.com/asiridissa/Tello_object_detection_demo_application. ("asiridissa/Tello_object_detection_demo_application: Building an Android Application to control Tello Drone and perform real-time object detection using YOLOv5," n.d.).

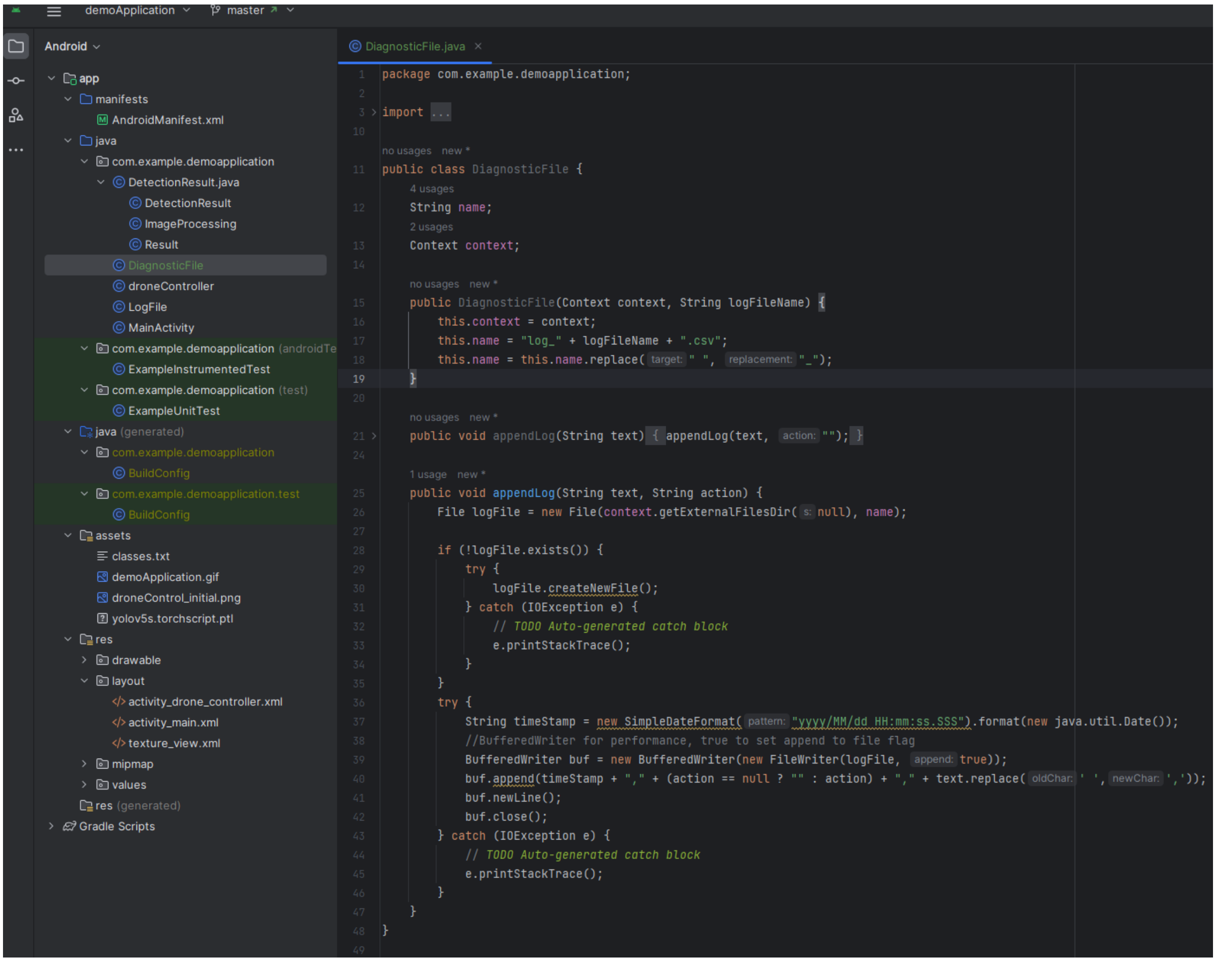


Figure 54. Android Studio project

DiagnosticFile.java   droneController.java ×

```java
        JoystickView leftjoystick = (JoystickView) findViewById(R.id.joystickViewLeft); // left joystick where the angle is the movement angle and strength is the
        leftjoystick.setOnMoveListener((angle, strength) -> {
            String action = "";

            if (angle > 45 && angle <= 135) {
                action = "UP";
                RC[2] = strength;
            }
            if (angle > 226 && angle <= 315) {
                action = "Down";
                strength *= -1;
                RC[2] = strength;
            }
            if (angle > 135 && angle <= 225) {
                action = "Counter clockwise";
                strength *= -1;
                RC[3] = strength;
            }
            if (angle > 316 && angle <= 359 || angle > 0 && angle <= 45) {
                action = "Clockwise";
                RC[3] = strength;
            }

            telloConnect( strCommand: "rc " + RC[0] + " " + RC[1] + " " + RC[2] + " " + RC[3], action); // send the command eg,. 'rc 10 00 32 00'
            Arrays.fill(RC,  val: 0); // reset the array with 0 after every virtual joystick move
            Log.v( tag: "Action",action);
        }, loopInterval: 100);

        JoystickView rightjoystick = (JoystickView) findViewById(R.id.joystickViewRight);
        rightjoystick.setOnMoveListener((angle, strength) -> {
            String action = "";

            if (angle > 45 && angle <= 135) {
                action = "Forward";
                RC[1] = strength;
            }
            if (angle > 226 && angle <= 315) {
                action = "Backward";
                strength *= -1;
                RC[1] = strength;
            }
            if (angle > 135 && angle <= 225) {
                action = "Left";
                strength *= -1;
                RC[0] = strength;
            }
            if (angle > 316 && angle <= 359 || angle > 0 && angle <= 45) {
                action = "Right";
                RC[0] = strength;
            }

            telloConnect( strCommand: "rc " + RC[0] + " " + RC[1] + " " + RC[2] + " " + RC[3], action);
            Arrays.fill(RC,  val: 0); // reset the array with 0 after every virtual joystick move
            Log.v( tag: "Action",action);
        }, loopInterval: 100);

        videoFeedaction = findViewById(R.id.videoFeed);
```

Figure 55. Android App Joystick controller code

Voice Windows Application

Source code is in GitHub repository https://github.com/asiridissa/TelloCLI. (Dissanayaka, 2023)

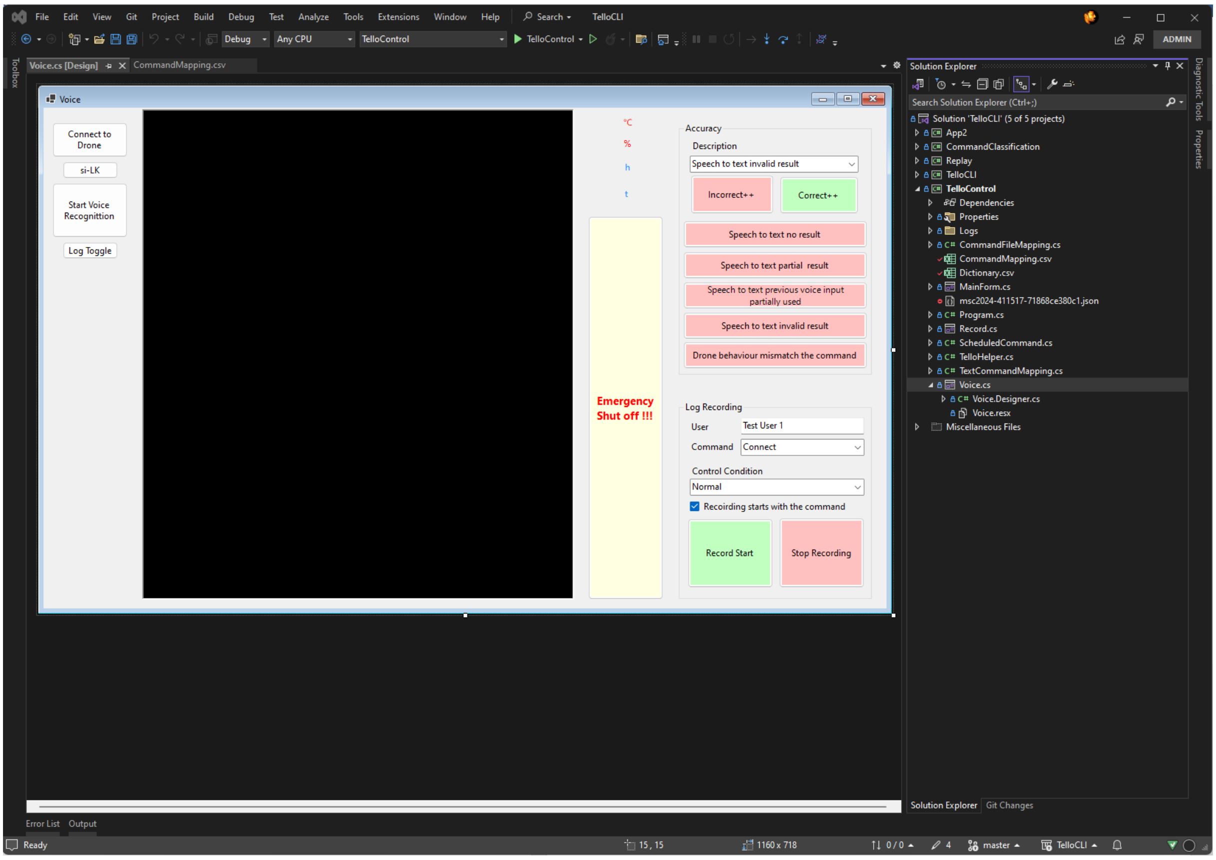


Figure 56. Visual Studio 2022 Voice application solution

```
2 references | Asiri Dissanayaka, 10 hours ago | 1 author, 8 changes
public async Task StartRecognition(string languageCode)
{
    AppendConsole($"{languageCode} Starting...");
    if (waveIn != null)
    {
        // Stop current recording if any
        waveIn.StopRecording();
        waveIn.Dispose();
    }

    var config = new RecognitionConfig
    {
        Encoding = RecognitionConfig.Types.AudioEncoding.Linear16,
        SampleRateHertz = 16000,
        LanguageCode = languageCode,

    };

    var streamingConfig = new StreamingRecognitionConfig
    {
        Config = config,
        InterimResults = false,
    };

    streamingCall = speechClient.StreamingRecognize();
    await streamingCall.WriteAsync(new StreamingRecognizeRequest
    {
        StreamingConfig = streamingConfig
    });

    waveIn = new WaveInEvent
    {
        WaveFormat = new WaveFormat(16000, 1)
    };
    waveIn.DataAvailable += async (sender, args) =>
    {
        await streamingCall.WriteAsync(new StreamingRecognizeRequest
        {
            AudioContent = Google.Protobuf.ByteString.CopyFrom(args.Buffer, 0, args.BytesRecorded)
        });
    };

    waveIn.StartRecording();
    AppendConsole("Speak now...");

    var stopwatch = new Stopwatch();
    Task.Run(async () =>
    {
        await foreach (var response in streamingCall.GetResponseStream())
        {
            stopwatch.Start();
            foreach (var result in response.Results)
            {
                //if (result.IsFinal)
                foreach (var alternative in result.Alternatives)
                {
                    Invoke(new Action(() =>
                    {
                        //AppendConsole($"Transcript: {alternative.Transcript}, Confidence: {alternative.Confidence}, Command : {TextCommandMapping.GetCommandClass(alternative.Transcript)}");
                        Log(delaysFilePath, $"Text delay,{stopwatch.ElapsedTicks}", true); // 1 tick = 100ns
                        IssueVoiceCommand(alternative.Transcript);
                        Log(delaysFilePath, $"Command delay,{stopwatch.ElapsedTicks}", true);
                        stopwatch.Reset();
                    }));
                }
            }
        }
    });
}
```

Figure 57. Google Speech to Text API calling code

Miscellaneous Files

Files that are not included in the appendices can be found in the following Google drive folder.

https://drive.google.com/drive/folders/1GlnsgZXHYZ3StaDfk-UuWptGXfCTqmP6?usp=drive_link

Figure 58 Miscellaneous files link

REFERENCES

Anand, S., Mathiyazaghan, R., 2016. Design and Fabrication of Voice Controlled Unmanned Aerial Vehicle. IAES International Journal of Robotics and Automation (IJRA) 5, 205. https://doi.org/10.11591/ijra.v5i3.pp205-212

Anand, S.S., Mathiyazhagan, R., Kalaimagal, S., 2018. Design and Fabrication of Voice Controlled Unmanned Aerial Vehicle. International Journal of Engineering Research & Technology 4. https://doi.org/10.17577/IJERTCONV4IS26021

Android | Do More With Google on Android Phones & Devices [WWW Document], n.d. URL https://www.android.com/ (accessed 3.20.24).

asiridissa/Tello_object_detection_demo_application: Building an Android Application to control Tello Drone and perform real-time object detection using YOLOv5 [WWW Document], n.d. URL https://github.com/asiridissa/Tello_object_detection_demo_application (accessed 3.5.24).

Basak, S., Agrawal, H., Jena, S., Gite, S., Bachute, M., Pradhan, B., Assiri, M., 2022. Challenges and Limitations in Speech Recognition Technology: A Critical Review of Speech Signal Processing Algorithms, Tools and Systems. Computer Modeling in Engineering & Sciences 135, 1–37. https://doi.org/10.32604/cmes.2022.021755

C# | Modern, open-source programming language for .NET [WWW Document], n.d. . Microsoft. URL https://dotnet.microsoft.com/en-us/languages/csharp (accessed 3.20.24).

Chandarana, M., Meszaros, E., Trujillo, A., Allen, B., 2017. “Fly Like This”: Natural Language Interfaces for UAV Mission Planning.

Contreras, R., Ayala, A., Cruz, F., 2020a. Unmanned Aerial Vehicle Control through Domain-Based Automatic Speech Recognition. Computers 9, 75. https://doi.org/10.3390/computers9030075

Contreras, R., Ayala, A., Cruz, F., 2020b. Unmanned Aerial Vehicle Control through Domain-Based Automatic Speech Recognition. Computers 9, 75. https://doi.org/10.3390/computers9030075

Dissanayaka, A., 2024. asiridissa/Tello_object_detection_demo_application.

Dissanayaka, A., 2023. asiridissa/TelloCLI.

Download Android Studio & App Tools - Android Developers [WWW Document], n.d. URL https://developer.android.com/studio (accessed 3.20.24).

Download .NET (Linux, macOS, and Windows) [WWW Document], n.d. . Microsoft. URL https://dotnet.microsoft.com/en-us/download (accessed 3.20.24).

Java | Oracle [WWW Document], n.d. URL https://www.java.com/en/ (accessed 3.20.24).

jithin8mathew/Tello_object_detection_demo_application: Building an Android Application to control Tello Drone and perform real-time object detection using YOLOv5 [WWW Document], n.d. URL https://github.com/jithin8mathew/Tello_object_detection_demo_application (accessed 3.5.24).

Jones, G., Berthouze, N., Bielski, R., Julier, S., 2010. Towards a situated, multimodal interface for multiple UAV control, in: 2010 IEEE International Conference on Robotics and Automation. Presented at the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010), IEEE, Anchorage, AK, pp. 1739–1744. https://doi.org/10.1109/ROBOT.2010.5509960

Kardasz, P., Doskocz, J., 2016. Drones and Possibilities of Their Using. Journal of Civil & Environmental Engineering 6. https://doi.org/10.4172/2165-784X.1000233

Kim, Soram, Lee, S., Kang, H., Kim, Sion, Ahn, M., 2021. P300 Brain–Computer Interface-Based Drone Control in Virtual and Augmented Reality. Sensors 21, 5765. https://doi.org/10.3390/s21175765

Kumaar, S., Bazaz, T., Kour, S., Gupta, D., M. Vishwanath, R., S N, O., 2018. A Deep Learning Approach to Speech Based Control of Unmanned Aerial Vehicles (UAVs), in: Computer Science & Information Technology (CS & IT). Presented at the 7th International Conference on Soft Computing, Artificial Intelligence and Applications, AIRCC Publishing Corporation, pp. 19–30. https://doi.org/10.5121/csit.2018.81003

Menshchikov, A., Ermilov, D., Dranitsky, I., Kupchenko, L., Panov, M., Fedorov, M., Somov, A., 2019. Data-Driven Body-Machine Interface for Drone Intuitive Control through Voice and Gestures. https://doi.org/10.1109/IECON.2019.8926635

Siegert, I., Sinha, Y., Jokisch, O., Wendemuth, A., 2020. Recognition Performance of Selected Speech Recognition APIs – A Longitudinal Study. pp. 520–529. https://doi.org/10.1007/978-3-030-60276-5_50

Suarez Fernandez, R., Sanchez-Lopez, J.L., Sampedro Pérez, C., Bavle, H., Molina, M., Campoy, P., 2016. Natural user interfaces for human-drone multi-modal interaction. https://doi.org/10.1109/ICUAS.2016.7502665

Tello App - Download Center - DJI [WWW Document], n.d. . DJI Official. URL https://www.dji.com/global/downloads/djiapp/tello (accessed 3.6.24).

Thu, K.M., Gavrilov, A.I., 2017. Designing and Modeling of Quadcopter Control System Using L1 Adaptive Control. Procedia Computer Science 103, 528–535. https://doi.org/10.1016/j.procs.2017.01.046

Visual Studio: IDE and Code Editor for Software Developers and Teams [WWW Document], n.d. . Visual Studio. URL https://visualstudio.microsoft.com (accessed 3.20.24).

Welcome to AirSim, 2023.

Wu, M., 2023. Robotics Applications in Natural Hazards. Highlights in Science, Engineering and Technology 43, 273–279. https://doi.org/10.54097/hset.v43i.7429