

AMERICAN UNIVERSITY OF BEIRUT

PERSONALIZED TELEOPERATION VIA INTENTION
RECOGNITION

by
SERGE RAMZI MGHABGHAB

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
April 2017

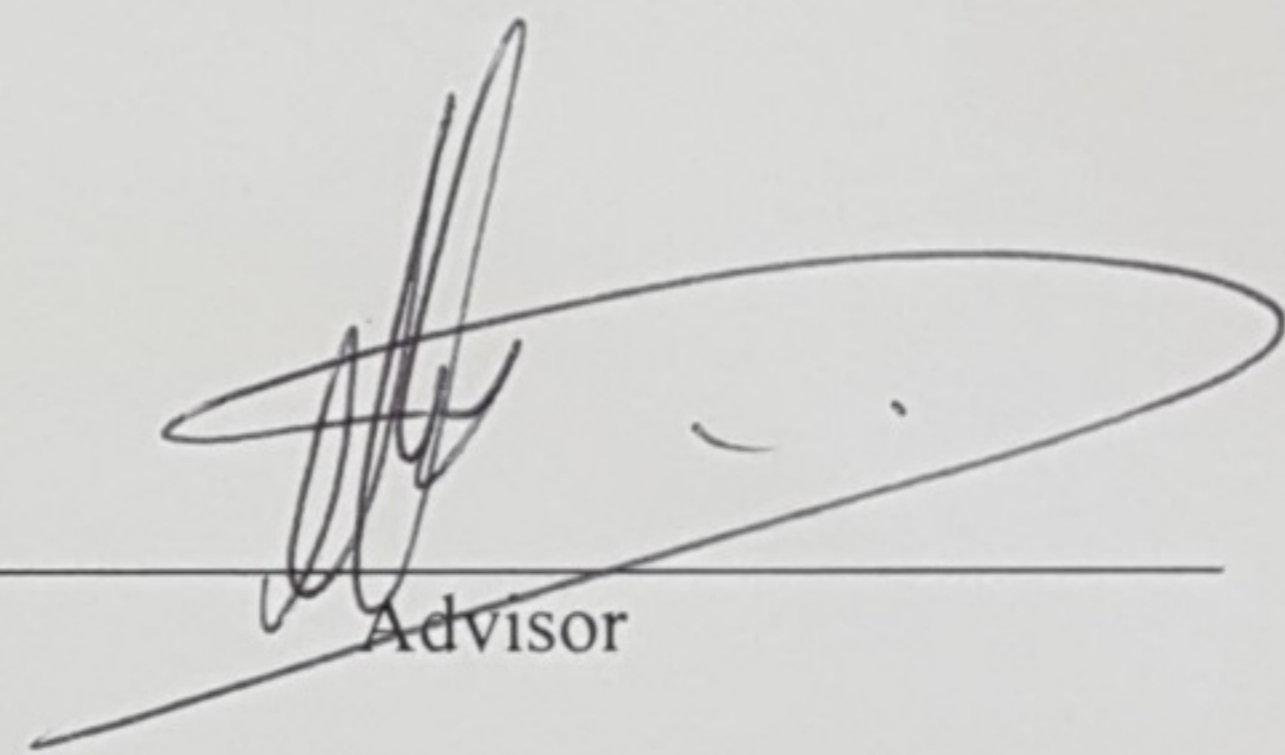
AMERICAN UNIVERSITY OF BEIRUT

PERSONALIZED TELEOPERATION VIA INTENTION
RECOGNITION

by
SERGE RAMZI MGHABGHAB

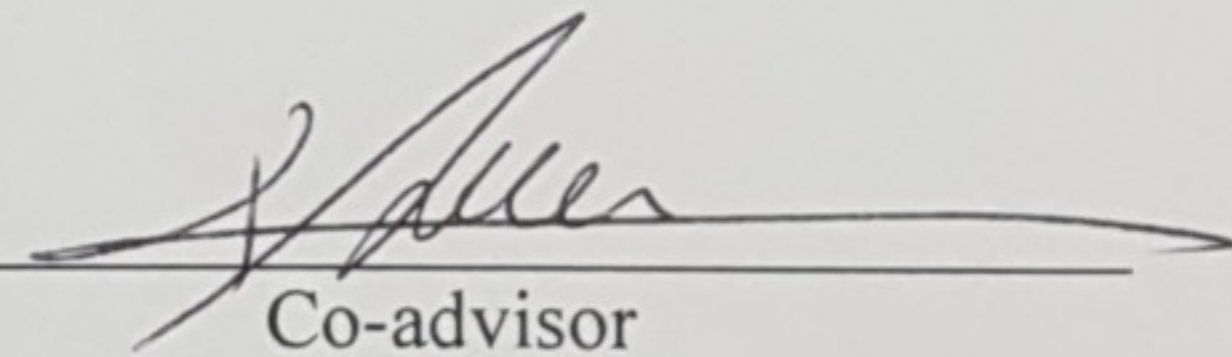
Approved by:

Dr. Imad H. Elhajj, Associate Professor
Department of Electrical Engineering



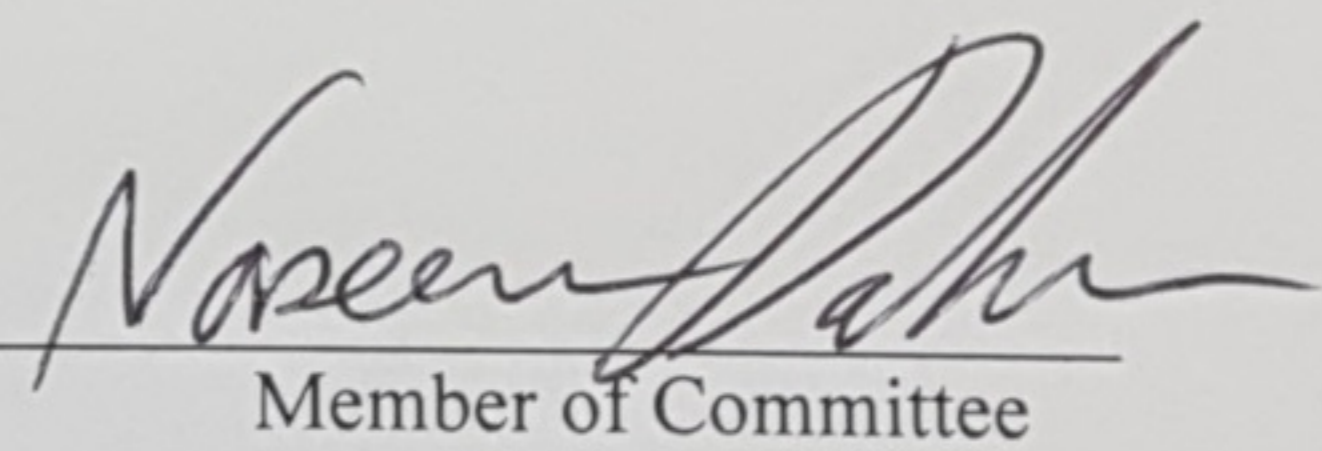
Advisor

Dr. Daniel Asmar, Associate Professor
Department of Mechanical Engineering



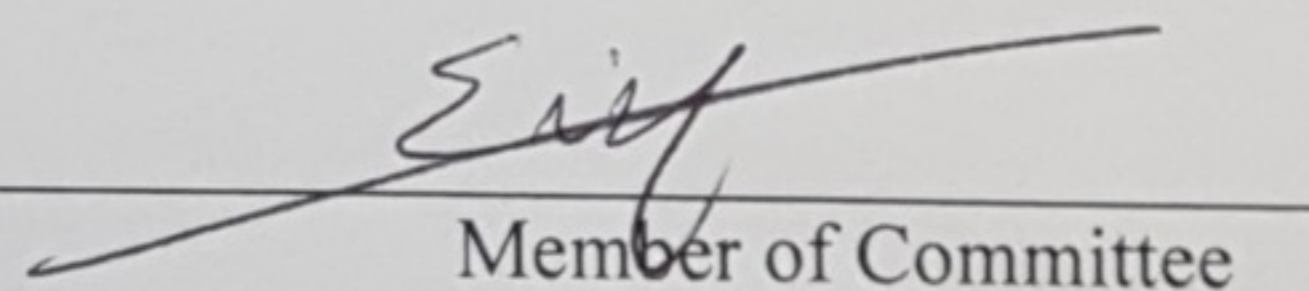
Co-advisor

Dr. Naseem Daher, Assistant Professor
Department of Electrical Engineering



Member of Committee

Dr. Elie Shamma, Assistant Professor
Department of Mechanical Engineering



Member of Committee

Date of thesis defense: April 27, 2017

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: _____ Mghabghab _____ Serge _____ Ramzi _____
Last First Middle

Master's Thesis

Master's Project

Doctoral Dissertation

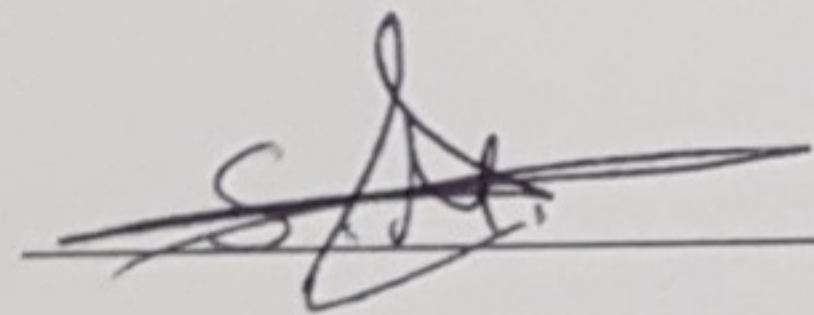
I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes
after:

One ---- year from the date of submission of my thesis, dissertation, or project.

Two ---- years from the date of submission of my thesis, dissertation, or project.

Three years from the date of submission of my thesis, dissertation, or project.



Signature

8 May 2017

Date

ACKNOWLEDGMENTS

Firstly, I would like to thank God for all the progress that I have achieved in my life and especially during my master studies. Also, I would like to thank him for all the help I was given during my academic years.

I would like to express my recognition and sincere gratitude to my advisor Prof. Imad Elhajj and co-advisor Prof. Daniel Asmar for their continuous support of my master studies, their motivation, and supervision.

Besides my advisor and co-advisor, I would like to thank the rest of my thesis committee: Prof. Naseem Daher, and Prof. Elie Shamma, for their guidance and encouragement.

My recognition and gratitude are addressed to AUB University Research Board and the Lebanese National Council for Scientific Research for their financial support.

I would like to thank the members of the Vision and Robotics Lab at the American University of Beirut for their continuous support and efforts exerted in the preparation of the testing area. Also, I would like to show my gratitude to Aline Eid, Georges Younes, Noel Maalouf, Marie-Joe Najm and Ali Kanso for the nights we were working together, and for the research-related discussions that we held.

Last but not least, special thanks for every person that participated in the organized experiments.

AN ABSTRACT OF THE THESIS OF

Serge Ramzi Mghabghab

for

Master of Engineering

Major: Control Systems

Title: Personalized Teleoperation Via Intention Recognition

One of the major challenges in teleoperation is the recognition of a user's intended commands, particularly in the manning of highly dynamic systems such as drones. Since the introduction of unmanned aerial vehicles (UAVs), their teleoperation was and still is a challenging task. Significant research was conducted to facilitate and improve UAV maneuverability, and yet many accidents are attributed to human error. The main purpose of this thesis is to build a teleoperation system that allows users to teleoperate any UAV in an intuitive manner.

Remote controls (RCs) are commonly used to control UAVs, but with their constant input mapping parameters and fixed configuration, RCs do not suit all users and require significant training. In this work, we are addressing the issue faced with this type of controllers, where multiple new teleoperation techniques are investigated and tested to design a powerful teleoperation system that acts based on the intentions of pilots.

Haptic joysticks are usually used as a replacement for RCs in the field of teleoperation. With the conducted experiments, it has been shown that with the appropriate configuration, haptic joysticks are easy to use and help pilots to perform the tasks in an efficient and accurate manner. But studies were mainly focused on using one joystick and only few were interested in the use of multiple joysticks.

As the user performance was not improved by using two joysticks, a new approach was adopted to enhance the teleoperation system. We focused on modifying the mapping parameters of the joystick based on the pilot commands. An adaptive gain tuning algorithm is used to map the masters' inputs (RC inputs) to the slave (UAV) motions to enhance the pilot's performance, regardless of the subject's experience. Based on his/her commanding characteristics the input sensitivity and smoothing of the joystick is modified. Experimental results showed a significant improvement when adaptive gain tuning is employed.

Once the adaptive gain tuning algorithm proved to be efficient, it was time to apply a more generalized algorithm to modify the commands of the pilot. The adaptive gain

tuning algorithm was built by only for the teleoperation of UAVs using only RCs. Also, the proposed equations are purely based on observation. A machine learning algorithm should be applied to detect these hidden features and customize the joystick in a more personalized manner. We present a solution to this problem by relying on a Convolutional Neural Network (CNN) that is trained to recognize the user's intended commands, directed through a haptic device. Our proposed method allows for any human machine interface to be personalized to each user once the CNN is trained. Experiments were conducted using two haptic devices and classification results demonstrate that the proposed system outperforms ANN and geometric-based approaches. In fact, the average classification percentage for CNN is equal to 78.56% compared to 76.24% for ANN, and 67% for geometric-based classifier. Furthermore, our system lends itself to other human machine interfaces where intention recognition is required.

The proposed teleoperation system was compared to velocity input based teleoperation system, and to the conventional teleoperation system. CNN based teleoperation system proved to perform better objectively and subjectively than the other selected ones.

CONTENTS

ACKNOWLEDGMENTS	v
ABSTRACT.....	vi
LIST OF ILLUSTRATIONS	xi
LIST OF TABLES.....	xiii

Chapter

I. INTRODUCTION.....	1
II. PREVIOUS WORK.....	4
A. Types of UAV Controllers.....	5
B. Types of Haptic Feedback	5
C. Controller’s Mapping Algorithms	6
D. Mapping Parameters	7
E. Machine Learning and Intention Recognition	8
III. TELEOPERATION SYSTEM USING TWO HAPTIC JOYSTICKS	11
A. Gestures Interpretation Algorithm.....	11
B. Haptic Feedback	15
1. Coulomb Force Field:.....	16
2. Virtual Stick Force Field:.....	17
a. Condition 1: Translation Gestures.....	18
b. Condition 2: Rotation Gestures	19
3. Final Force Field	19

IV. ADAPTIVE GAIN TUNING	20
A. The Proposed Methodology	20
1. Adjusting sensitivity.....	21
2. Regulating the fluctuation of input commands	23
3. Reducing the roughness in input commands.....	23
4. Adaptive Algorithm.....	24
B. Experimental Setup and Results	26
1. Experimental Results and Setup.....	26
2. Detailed Procedure	27
3. Testing Results	28
V. INTENTION RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK.....	32
A. Dataset Collection.....	33
B. Intention Recognition Using Neural Networks	34
1. ANN Architecture	34
2. CNN Architecture	36
a. Temporal stratification representation	36
b. Representation by a 3-channel input	37
C. Dataset Collection and Classification Experimental Setup and Results. 41	
1. Format of the input feature vector.....	41
2. Experimental Design and Setup	42
3. Results	43
a. ANN vs Temporal Stratification Representation	43
b. Temporal Stratification Representation vs Representation by a 3-channel Input	44
D. Testing Experimental Setup and Results	50
1. Experimental Design and Setup	50
a. Modes of Teleoperation	50
i. Mode 1: CNN Based Teleoperation System	50
ii. Mode 2: Velocity Based Teleoperation System...	51
iii. Mode 3: Conventional Teleoperation System	52

b. Testing Area	53
c. Testing Design and Procedure.....	55
2. Testing Results	57
a. Objective Results.....	57
b. Subjective Results	60
VI. CONCLUSION AND FUTURE WORK	62

LIST OF ILLUSTRATIONS

Figure	Page
1. AscTec Pelican quadrotor	2
2. RC Inputs Configuration and related Quadrotor Dynamics	2
3. Haptic joysticks with the UAV axis	12
4. Thrust, pitch and 1st roll gestures	13
5. Yaw and 2nd roll gestures	14
6. UAV Teleoperation Diagram.....	15
7. Coulomb Force Field	17
8. Virtual Stick Force Field.....	18
9. Parameters tuning diagram	21
10. Sample of pitch inputs marked with the local maxima and the corrections done to the preceding inputs	22
11. Comparison between the original and modified RC inputs using the proposed algorithm.....	26
12. Testing trajectory	28
13. Objective results for all the trials showing the average and the std. dev. of the flight duration, distance to target and distance traveled	29
14. Objective results for the acceptable trials showing the average and the std. dev. of the flight duration, distance to target and distance traveled	30
15. Motivation for the problem: The user inputs a command to the joystick intending to direct it in a vertical direction; however, his commands are flawed and would result in forward motion V_x in addition to the intended V_y . A classifier is trained to decipher his true intention and command the drone with the correct motion, setting V_x to zero.....	33
16. ANN consecutive inputs	35
17. ANN architecture.....	35
18. CNN architecture for the stratified representation.....	37

19.	The three-dimensional input matrix with depth equal to 3.....	38
20.	Representation of one depth of the input matrix.....	39
21.	Sample of the RGB input matrix representation.....	40
22.	CNN architecture for 3-channel input representation.....	41
23.	Example of a motion in the right direction on the left haptic device.....	42
24.	First trial of each motion for subject 8.....	46
25.	first trial of each motion for subject 11.....	47
26.	Confusion matrices that describe the classification inputs of user 11 - The five classes in this figure represent the following motions and velocities: 1 - negative motion with high velocity, 2 - negative motion with low velocity, 3 - no motion, 4 - positive motion with low velocity, 5 - positive motion with high velocity.....	48
27.	CNNs used to map user inputs to the UAV	51
28.	Parrot AR drone 2.0 teleoperation application	53
29.	Flight testing trajectory	54
30.	Image showing the testing area.....	54
31.	NASA Task Load Index	56
32.	hardware connections	57
33.	Objective results of the three proposed modes	58

LIST OF TABLES

Table	Page
1. Difference between the two modes and ANOVA test results	31
2. Example input for the stratified representation.....	36
3. ANN vs CNN classification results for the participants	44
4. Classification results comparison for the two proposed CNN architectures	45
5. Classification results for the table based classifier	49
6. Comparison between the three modes	59
7. P-values of the ANOVA test results	60
8. Subjective results of the experiment on Parrot AR drone 2.0	60

CHAPTER I

INTRODUCTION

Teleoperation is the process of remote controlling a distant robot, and it is found in multiple areas. In the medical area, teleoperation is applied to the surgery of patients [1], the diagnosis of the patients [2], and many other applications. In the military area, the teleoperation of robot is present in many forms, aurally through drones and on ground through explosive detectors [3] and others. As well, teleoperation systems are present in other areas and are mostly common in the robotic arm systems [4], ground vehicles [3], underwater vehicles [5], and unmanned aerial vehicles (UAVs) [6-10].

UAVs are defined as flying vehicles that can be controlled and remotely operated without having a pilot on board. Typically, they are teleoperated by a pilot using RC, or controlled by an onboard computer [6].

Recently, UAVs grabbed the attention of many researchers for their capability to maneuver in harsh and dangerous areas. UAVs are now introduced to the surveillance field as they can maneuver in tight corridors, with low noise and high precision. Due to dynamic changes in the biological, physical, and chemical conditions in the environment, UAVs are used in the management and surveillance of oceanic and coastal areas [11]. Also, due to their agility and capability to carry sensory devices (like cameras), UAVs can be used in mapping targets [12] and in 3D reconstruction [13].

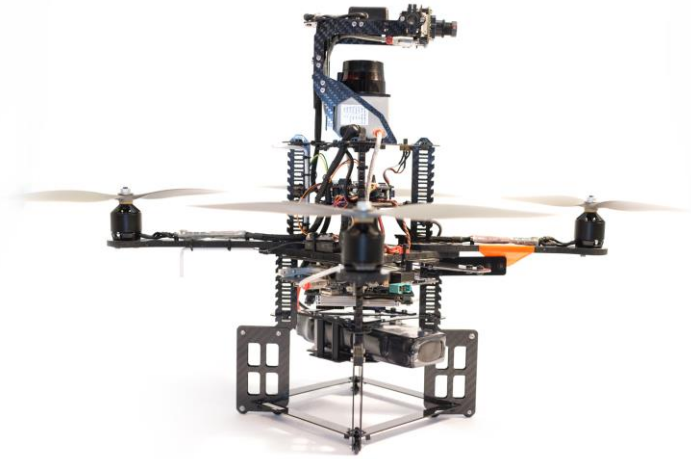


Figure 1: AscTec Pelican quadrotor

One of the most common and widely used types of UAVs are quadrotors (Figure 1), composed of four rotors positioned at a fixed length from the center. Quadrotors are under-actuated systems [7], thus controlling them is challenging, whether automatically or manually. Quadrotor dynamics is affected by three rotational angles about the vehicle body axes (roll, pitch and yaw), and a collective thrust applied in the perpendicular direction of the surface. In conventional RCs, the stick on the left is responsible for applying the thrust and yaw to the UAV as for the roll and pitch they are applied using the stick on the right side as shown in Figure 2.

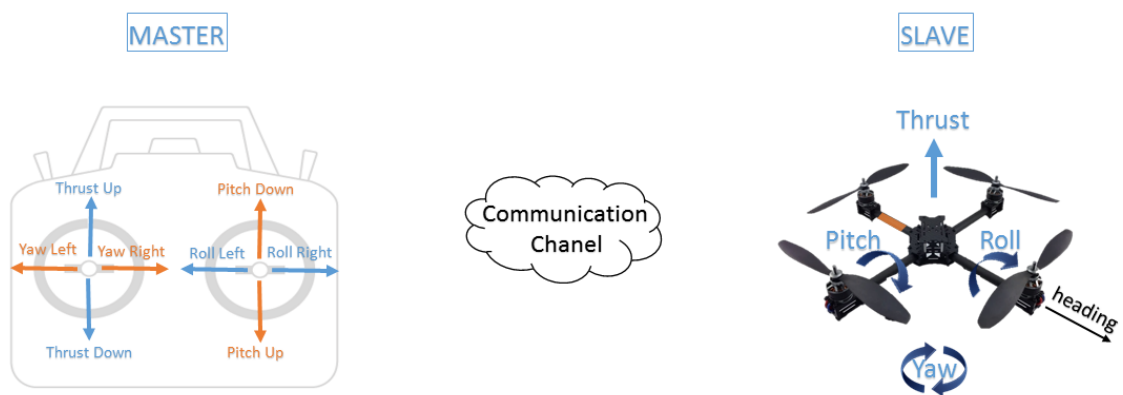


Figure 2: RC Inputs Configuration and related Quadrotor Dynamics

Teleoperation systems are generally formed of three main subsystems: the slave subsystem represented by the robot, the master subsystem represented by the controller, and the third subsystem consists of the communication channel that is present in between the master and the slave.

The development of a perfect controller is a hard task and hardly feasible for many applications. Teleoperating a robot is considered a challenging task especially when it is done by a novice operator. UAVs are extremely hard to command and many controllers have been designed in order to facilitate their teleoperation [14].

In the case where RC is used to command UAVs, inputs are mapped in linear form, where the full scale of RC inputs is transformed to the full scale of outputs to UAVs. Whereas, the input of joysticks could be mapped using multiple configurations. Some of the used joysticks can handle haptic feedback, allowing it to serve as a sensory device.

This thesis includes the following main parts: chapter II presents literature review on the various systems implemented to teleoperate UAVs. An overview of the first proposed UAV teleoperation technique that is achieved by using two haptic joysticks is presented in chapter III. Chapter IV expands the adaptive gain tuning algorithm, used to enhance the mapping parameters of any controller, regardless of the mapping algorithm used and to make them personalized for each user. Chapter V explains the newly proposed mapping algorithm that uses the CNN to recognize the intention of the pilot and drive the UAV in a personalized manner. At the end, Chapter VI concludes the thesis and proposes future work.

CHAPTER II

PREVIOUS WORK

Since the time UAVs were introduced, their teleoperation was and still is challenging. Significant research has been conducted in order to facilitate and improve UAVs maneuverability, and yet to this date, the solutions presented can be improved. Remotely teleoperating an aerial vehicle is an extremely difficult task, especially if the pilot has no prior experience in this field.

Different approaches were explored in order to make teleoperation more natural and intuitive. These approaches were mostly related to the design of new UAV controllers or related to the modification of the existing ones. The adjustment of the controllers could be achieved by adding a haptic force feedback that could help the subject while teleoperating a robot by adding a new sensory feedback. The modifications are also associated with the design of a new mapping algorithm where the user inputs are processed and transmitted using the same controller but with a different approach as shown in section C. Where the mapping parameters, are modified to make each controller subject-related.

The algorithms used in the literature for mapping approaches and mapping parameters are achieved through simple equations or through machine learning algorithms like hidden Markov models (HMMs) or Neural Networks.

A. Types of UAV Controllers

To facilitate the teleoperation task on inexperienced pilots, multiple control devices (ex: RC, haptic device, ...) were created and customized to ensure the best performance. In the literature, haptic joysticks were widely used to map the user's inputs to the quadrotor [8, 9, 15]. Others tackled this problem by designing a multimodal haptic interface, which consists of two input devices: a haptic joystick to control the position of the quadrotor, and a trackball to control its velocity [16]. The combination of two joysticks was introduced as an input device that controls multiple UAVs; one joystick was used to control the UAV flight direction, and the other was used to determine the distance between the multiple UAVs [17].

Besides the use of RCs, and conventional and custom-made joysticks, also gestural based input was used to control a UAV. In [18] the shape of the hands was analyzed, for the sake of teleoperating aerial vehicles, where it was possible to control one or multiple UAVs depending on the hand combinations.

B. Types of Haptic Feedback

As the user lacks sensory feedback from the quadrotor, it is crucial to provide haptic feedback to users for them to feel the UAV's dynamics and sense the mapping of the input commands. In [19] T. Lam et al. introduced artificial force feedback, in order to assist the pilot in controlling the UAV; hence, preventing future damages and crashes. In [20] investigation was made to determine whether the effect of force feedback is constructive or destructive. The authors used the average UAV speed and the average distance from target, to design the necessary haptic feedback. In fact, in degraded visual interfaces, using an artificial force field provided sufficient awareness

and reduced the number of collisions. In [21] haptic forces were exerted depending on the zone being categorized as safe or warning. In addition, a virtual spring was applied as a force, in the horizontal direction, to prevent collision with walls. Other forces were applied in the vertical plane to avoid floor and ceiling obstacles. All of these methods rely on the ability to sense the surrounding, prior to calculating force feedback. In places where it is impossible to map the surrounding, or when sensory equipment lacks, it is not reliable to use the previously mentioned types of force feedback to reduce the chances for crashing. In [15] repulsive forces were added based on the velocity of the quadrotor, which makes it harder for the pilot to control the quadrotor with undesired inputs; when sufficient input is given, repulsive force starts to contradict user's inputs. EMG signals were used in [22] to study the physiological signals and then this study was used to calibrate the muscles contraction subconsciously using an admittance force feedback model. As in the literature, haptic feedback helped in improving the quality of teleoperation, in chapter III, haptic feedback is based on the quadrotor velocity, which is measured using a Vicon System.

C. Controller's Mapping Algorithms

The performance of an operator commanding any machine is influenced by the design and the selection of the input mapping parameters, as described by Mitsuhiro Kamezaki et al. in [23]. Classical mapping algorithms use P2P (position to position) [8] and P2V (position to velocity) configurations; where in the first configuration, the position of the joystick tip determine the desired position of the UAV, and in the second configuration, the position of the joystick tip determine the velocity of the UAV. Farkhatdinov et al. [24] combined P2P and P2V configurations in order to control a

robot. In [15] the quadrotor was teleoperated by sensing the velocity of inputs and then it was transformed into UAV desired angles and thrust.

D. Mapping Parameters

Flying skills are one key for a successful flight, but the machine interface and the mapping of parameters play an essential role as well. Every operator perceives and deals with teleoperation differently. As seen in [26], each user forms his/her own internal model of the machine dynamics, and it is more beneficial to assist the user in forming more accurate internal model that describes the dynamics of the targeted robot.

Assistance should be applied without the awareness of the user.

In [23], the tuning of input mapping parameters was tackled by fitting the user's input into a normal distribution curve, which is achieved by changing the gain and the mapping variables. This configuration was tested on a hydraulic arm system and had a significant improvement on the user's workability and operability over the machine. Input gain tuning targeted different types of vehicles, extensive research was done on the teleoperation of wheelchairs. The change of input mapping parameters for wheelchair joystick in [27] was based on the states of the wheelchair and the parameters were tuned after evaluating the driver skills. The standard deviation of the wheelchair speed determined the driver skill, and the states were divided into three phases: start phase, transfer phase, and positioning phase. This tuning method showed a significant improvement in performance for novice drivers, whereas it had a minimal improvement for skilled drivers.

As shown, in the literature they modified the mapping parameters for multiple robot, but none focused on applying similar approaches to the teleoperation of UAVs. On the contrary, the adopted mapping techniques behave similarly for every operator.

E. Machine Learning and Intention Recognition

Intention recognition systems have been explored before in the literature in human-machine interaction. In the work of Khokar et al. [28], motion intention is recognized for the sake of teleoperating a robotic hand; Hidden Markov Models (HMMs) were used to recognize objects of interest, select the desired configuration, and aid the users in performing the required task. In an offline stage, an expert trained the HMMs, to recognize user intention using as input the motion of a robotic hand end-effector. Afterwards, during testing, the learned system adjusts the human-machine input-output gains to mitigate for human intention. The result was an increase in efficiency of the operators in performing a required task; however, the disadvantage of the proposed system is the requirement for a new learning stage for each new manipulated object of interest or new required motion.

In other work, intention recognition was used to select the proper control parameters for assisting an operator in commanding a robotic suitcase [29]. During a learning phase, different motion patterns and corresponding user inputs are synthesized, and stored in a database; then during operation, user inputs are assessed and matched to the most similar motion pattern in the database. The major limitation of this method is that it is not possible to input new commands that were not considered in the training set.

In the work of Rabhi et al. [30], the performance of physically challenged users driving their wheelchairs was improved with the aid of machine learning. During an offline stage, an Artificial Neural Network (ANN) is trained with user commands as inputs and intended controls as outputs. The ANN is then used during testing to aid in conveying the correct intention to the control of the wheelchair. But in this work, they were only interested in modifying the rotational movements along with their accelerations.

Gestural-based teleoperation of a mobile robot was investigated in Tzafestas et al. [31]; with the aid of multi-layer perceptron neural network, it was possible to recognize the shape of hands from images. Multiple shape descriptors were used to represent the segmented hand shapes and served as inputs to the neural network. With this architecture, they were able to control a mobile robot in real time using multiple static hand postures. However, this system uses one to one mapping between each gesture and its corresponding movement, and the limited number of gestures constrains the robot's maneuverability.

The modern CNNs was originally introduced in 1998 in the paper of LeCun and Bottou [32] but its application was constrained by the available processing power at the time. However, with the increase in processing power in our day and with the replacement of the traditional sigmoid activation function by the more successful rectified linear unit (ReLU), the applications of CNNs in the computer vision community has exploded.

In the paper of Simonyan and Zisserman work [33], a two-stream architecture was proposed to recognize the action of the targeted human, relying on both temporal and spatial components, each trained with its own CNN. Fusing the results of the CNN was

attempted in two different manners: one involving a simple average of the results, and a second relying on an SVM. Later, Park et al. [34] proposed a different approach for combining multiple CNN consensus. First, they rely on feature amplification optical flow to perform spatially varying soft gating on intermediate CNN feature maps. Second, they use spatially varying multiplicative fusion for combining multiple CNNs trained on different sources. This fusion results in robust prediction, by amplifying or suppressing the feature activations based on the features agreement. The disadvantage of both of these systems is that they require complex computations and thus can only be processed at the end of a given input video.

To the best of our knowledge, the personalized mapping of input commands to the UAV has not been addressed in the literature. The main focus of my work is to create a mapping architecture that enhances the performance of the subjects regardless the characteristics of each person.

My main contribution is to propose a new teleoperation system that permits all the users, regardless of their piloting skills, better to fly UAVs. The system should account for the behavior of all users and adapt to them in order to achieve the best performance.

CHAPTER III

TELEOPERATION SYSTEM USING TWO HAPTIC JOYSTICKS

In the previously proposed methods only one haptic joystick was used to control the quadrotor; as a result, the control degrees of freedom are limited. In contrast, we propose a new method to control the UAV using two haptic joysticks.

The mapping of inputs to the UAV is designed by mapping the two joysticks' inputs. Several gestures are applied to control the quadrotor, and every quadrotor's motion is triggered by a predefined combination of inputs. Knowing that novice pilots cannot control a UAV in a desired manner, without having any assistance or prior training, three adaptive mapping techniques are applied to facilitate the teleoperation procedure. These adaptive techniques modify the scaling factor for the parameters depending on each user's performance.

To compensate for the separation of the pilot from the quadrotor, a force feedback algorithm is implemented, where the user is able to sense the state of the quadrotor.

A. Gestures Interpretation Algorithm

In this thesis, we use a pair of Phantom Omni devices as haptic joysticks. Figure 3 shows the haptic joysticks (Master) used for teleoperation, along with the quadrotor (Slave) and its axes.

The proposed method consists of five gestures that map the thrust, roll, pitch and yaw motions. The roll motion can be triggered using two gestures, which give flexibility

to the user in choosing the desired gesture. Each of the other motions can be triggered by one gesture. The buttons on the right haptic joystick are used to hover and land the quadrotor; button 1 (blue button) is used to activate the hovering mode at a certain position, while button 2 (white button) activates the landing mode. The roll, pitch, yaw and thrust are mapped as functions of the following inputs:

$$\text{Roll: } \varphi = k_1 * \left(F(P_{x_r}, P_{x_l}) + F(P_{y_r}, P_{y_l}) \right) \quad (1)$$

$$\text{Pitch: } \theta = k_2 * F(P_{z_r}, P_{z_l}) \quad (2)$$

$$\text{Yaw: } \psi = k_3 * F(P_{z_r}, P_{z_l}) \quad (3)$$

$$\text{Thrust: } T = k_4 * F(P_{y_r}, P_{y_l}) + T_{UAV} \quad (4)$$

where $k_1 = F(P_{x_r}, P_{x_l}, P_{y_r}, P_{y_l})$, $k_2 = F(P_{z_r}, P_{z_l})$, $k_3 = F(P_{z_r}, P_{z_l})$ and $k_4 = F(P_{y_r}, P_{y_l})$. k_1, k_2, k_3 and k_4 are adaptive variables used to adjust the sensitivity of the commands. T_{UAV} is the hovering value of the UAV, and is added to the thrust equation in order to maintain the hovering state of the UAV when no input commands are applied. $P_{x_r}, P_{x_l}, P_{y_r}, P_{y_l}, P_{z_r}$ and P_{z_l} are the components of the joysticks' position vector at each instance of time. Subscripts “r” and “l” stand for “right” and “left” joysticks.



Figure 3: Haptic joysticks with the UAV axis

In order to decouple the different motions, there is a dead zone that allows the operator to perform part of his gesture before starting to analyze the given command. Thus, the system will only catch the desired motion and avoid the noisy ones. Even though each motion is decoupled, the user can perform a combination of desired motions (for example, the user can perform a thrust motion along with a roll motion). Once the sticks of the haptic joysticks are back to the center, all of the motions are canceled, and the quadrotor enters the hovering mode again (if not accounting for any external or internal disturbance). The mapping of these gestures to the UAV is performed by calculating the difference in position between the actual position of the stick and the origin (or the “zero” position). Figure 4 and Figure 5 show the gestures used to control the quadrotor.

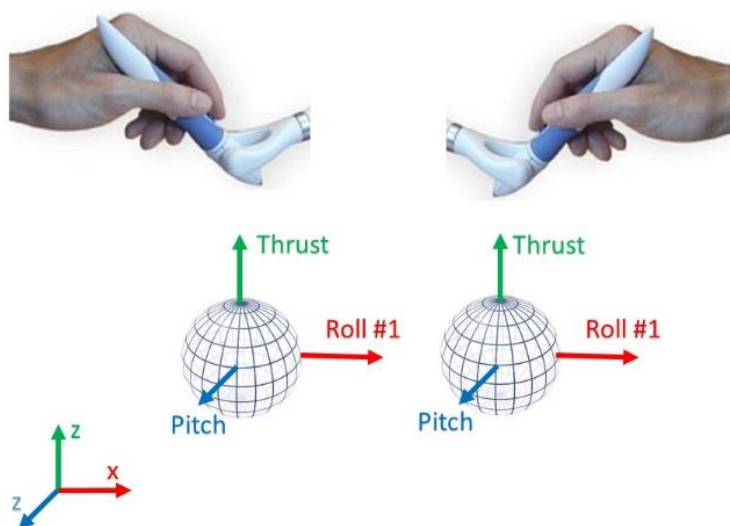


Figure 4: Thrust, pitch and 1st roll gestures

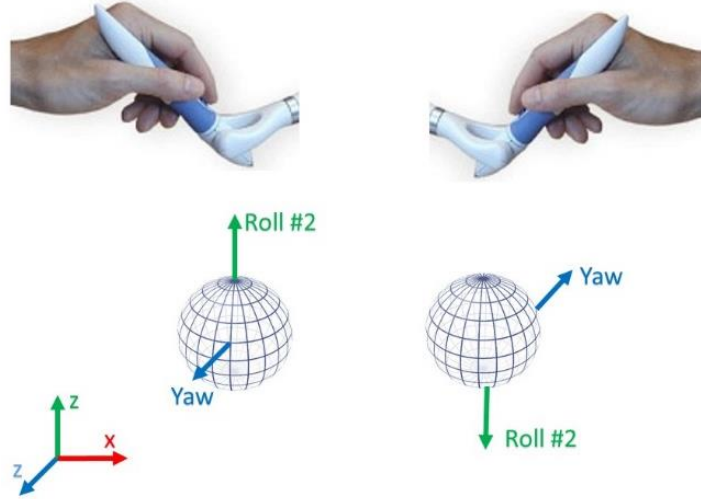


Figure 5: Yaw and 2nd roll gestures

Commanding gestures are mapped as follows:

$$\varphi_1(P_{x_r}, P_{x_l}) = \begin{cases} (P_{x_r} + P_{x_l} - 2R_s)/FS, & P_{x_r} > R_s \text{ and } P_{x_l} > R_s \\ (P_{x_r} + P_{x_l} + 2R_s)/FS, & P_{x_r} < -R_s \text{ and } P_{x_l} < -R_s \\ 0, & \text{else} \end{cases} \quad (5)$$

$$\varphi_2(P_{y_r}, P_{y_l}) = \begin{cases} (P_{y_l} - P_{y_r} - 2R_s)/FS, & P_{y_r} < -R_s \text{ and } P_{y_l} > R_s \\ (P_{y_l} - P_{y_r} + 2R_s)/FS, & P_{y_r} > R_s \text{ and } P_{y_l} < -R_s \\ 0, & \text{else} \end{cases} \quad (6)$$

$$\varphi(P_{x_r}, P_{x_l}, P_{y_r}, P_{y_l}) = k_1(\varphi_1(P_{x_r}, P_{x_l}) + \varphi_2(P_{y_r}, P_{y_l})) \quad (7)$$

$$\theta(P_{z_r}, P_{z_l}) = \begin{cases} -k_2(P_{z_r} + P_{z_l} - 2R_s)/FS, & P_{z_r} > R_s \text{ and } P_{z_l} > R_s \\ -k_2(P_{z_r} + P_{z_l} + 2R_s)/FS, & P_{z_r} < -R_s \text{ and } P_{z_l} < -R_s \\ 0, & \text{else} \end{cases} \quad (8)$$

$$\psi(P_{z_r}, P_{z_l}) = \begin{cases} k_3(P_{z_l} - P_{z_r} - 2R_s)/FS, & P_{z_r} < -R_s \text{ and } P_{z_l} > R_s \\ k_3(P_{z_l} - P_{z_r} + 2R_s)/FS, & P_{z_r} > R_s \text{ and } P_{z_l} < -R_s \\ 0, & \text{else} \end{cases} \quad (9)$$

$$T(P_{y_r}, P_{y_l}) = \begin{cases} k_4(P_{y_r} + P_{y_l} - 2R_s)/FS, & P_{y_r} > R_s \text{ and } P_{y_l} > R_s \\ k_4(P_{y_r} + P_{y_l} + 2R_s)/FS, & P_{y_r} < -R_s \text{ and } P_{y_l} < -R_s \\ 0, & \text{else} \end{cases} \quad (10)$$

where $R_s = \text{radius of dead - space}$, $FS = \text{Full scale}$ (used to scale the roll, pitch, yaw and thrust from 0 to 1).

B. Haptic Feedback

The relation between the haptic feedback and the position of the joysticks is poorly accounted for in the literature. The position of the Joysticks' sticks can serve as sensory devices. Based on the sticks positions, the haptic force can be increased, and hence the user will be able to sense the input levels sent to the quadrotor as shown in the diagram of Figure 6. This technique is usually used for gaming joysticks, and in this work, it is used for aerial vehicles joystick.

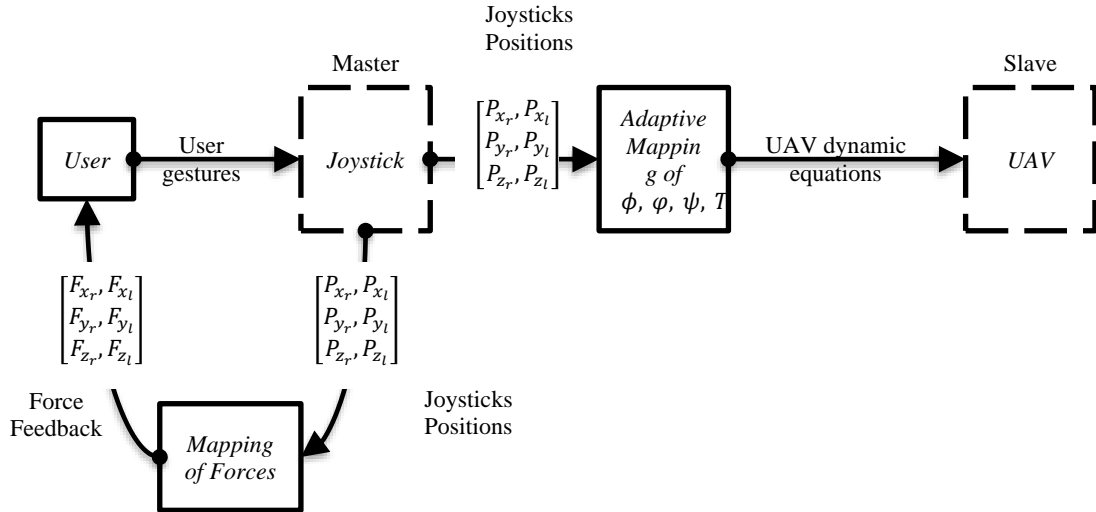


Figure 6: UAV Teleoperation Diagram

To let the user perceive the “zero” position of the haptic joystick, a “center of gravity” force feedback is introduced in the area of motion of the haptic joystick. This center of gravity attracts the haptic sticks to the center whenever the pilot wants to apply

a zero input. In [25] a mediator (object used to interconnect two joysticks' sticks together) is used to interconnect two joysticks, which limits the hand motions and ameliorates the user's performance. As such, in this system, a virtual stick connects the two haptic joysticks, allowing the pilot to operate in a predetermined workspace. Then, haptic feedback of the two joysticks will be a function of the following variables:

$$F_{x_r}, F_{x_l} = -k_5 * F(P_{x_r}, P_{x_l}) \quad (11)$$

$$F_{y_r}, F_{y_l} = -k_6 * F(P_{y_r}, P_{y_l}) \quad (12)$$

$$F_{z_r}, F_{z_l} = -k_7 * F(P_{z_r}, P_{z_l}) \quad (13)$$

where $k_5, k_6,$ and k_7 are parameters used to scale the intensity of the force feedback applied to the joystick. They are functions of joystick positions and each one of them is the combination of two types of forces: The Coulomb Force Field and the Virtual Stick Force Field. $P_{x_r}, P_{x_l}, P_{y_r}, P_{y_l}, P_{z_r}$ and P_{z_l} are the components of the joysticks' position vector at each instance of time.

1. Coulomb Force Field:

The gravitational center of each haptic joystick is placed on the center of the y-axis and z-axis but shifted from the center of the x-axis in order to reduce the distance separating the two joysticks: the right joystick has its center shifted to the left side and the left joystick has its center shifted to the right side.

The haptic forces are designed to direct the pilot's hand to the specified zero position of the haptic joysticks. But these haptic forces should not contradict the decision of the pilot.

The distance of the right haptic joystick's tip from its origin is calculated by the following formula:

$$D_r = \sqrt{(P_{x_r}^2 + P_{y_r}^2 + P_{z_r}^2)} \quad (14)$$

The Coulomb force field is obtained using the following formula:

$$F_{1_r}(D_r) = \begin{cases} -k_8 * P_r, & D_r < R_s \\ -k_9 * unitPos/D_r^2, & D_r \geq R_s \end{cases} \quad (15)$$

with $k_8 = 0.1, k_9 = 1200, unitPos = P_r/D_r, R_s$ is the dead-space radius and P_r is a 3-dimensional vector containing x, y and z coordinate of the right haptic joystick. F_{1_l} is calculated in the same way as F_{1_r} and the shape of this type of force is shown in Figure 7.

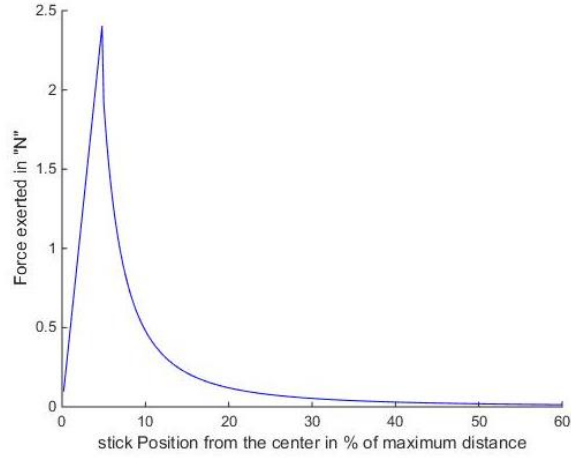


Figure 7: Coulomb Force Field

2. *Virtual Stick Force Field:*

The use of virtual stick is necessary for this configuration, as it will allow the pilot to only perform desired and predefined gestures. The purpose of this virtual stick is to preserve the initial distance between the two haptic joysticks regardless of the

performed gesture. This type of force allows the user to move freely in a dead zone. The virtual stick allows the user to move the two joysticks in a different orientation to let the quadrotor perform a roll (roll #2) or a yaw only if both joysticks are placed near the center position. This option makes it impossible for the pilot to perform undesired roll or yaw motions while controlling the quadrotor. The exerted forces intensify as long as the distance separating the two joysticks is being reduced or augmented.

a. Condition 1: Translation Gestures

The two equations stated below are responsible for keeping the two haptic joysticks at a given and a relatively constant distance. The forces produced from these equations are shown in Figure 8.

$$F_{2r} = (P_l - P_r)/k_{10} \quad (16)$$

$$F_{2l} = (P_r - P_l)/k_{10} \quad (17)$$

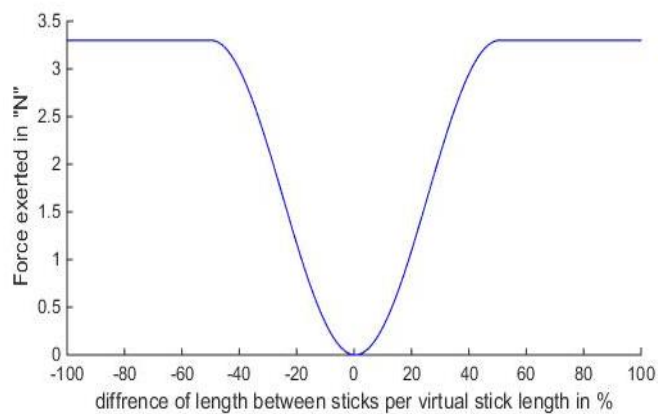


Figure 8: Virtual Stick Force Field

b. Condition 2: Rotation Gestures

When rotating the sticks to perform the yaw gesture, the effect of F_{2_r} and F_{2_l} fades away in the z-direction to allow soft hand motions. As such, the effect of F_{2_r} and F_{2_l} fades away in the y-direction when performing the second roll gesture.

3. Final Force Field

The final force field is the combination of the coulomb force field and the virtual stick force field as shown in the below equations:

$$F_r = F_{1_r} + F_{2_r} \quad (18)$$

$$F_l = F_{1_l} + F_{2_l} \quad (19)$$

Preliminary testing using the proposed teleoperation method in chapter III resulted in poor flight performance including multiple crashes. This proved that adding a second joystick and applying our proposed teleoperation system, increased the frustration level of the participants leading to worse performances. This directed our research focus to developing new approaches for a more user-friendly setup.

CHAPTER IV

ADAPTIVE GAIN TUNING

RCs are commonly used to control UAVs, but with their constant scale for inputs (input scale is the mapping variable used to map the amount of input applied by users to output), RCs do not suit all users and require significant training. In our work, we are addressing this issue, where adaptive gain tuning is used to map the masters' inputs (RC inputs) to the slave (UAV) motions in order to enhance the pilot's performance, regardless of the level of experience.

Our work focuses on tuning the parameters responsible for mapping the RC inputs to the quadrotor motions. The motivation behind this is that each user has their own technique in teleoperating UAVs, also each user adapts to the controllers in a different way, based on previous experience in the field, or based on other factors. Thus, the tuning algorithm should be able to help the pilot, by mapping the desired user commands to the UAV in an improved way; this should enhance the performance of each user in terms of accuracy and flight duration.

A. The Proposed Methodology

The quadrotor is commanded by four inputs which are the roll, pitch, yaw and thrust. Each one of these inputs is scaled linearly over the full range in the conventional RCs. In this paper, the proposed adaptive method for tuning the mapping parameters is formulated as follows: k_{roll} , k_{pitch} , k_{yaw} and k_{thrust} stand for the mapping parameters for roll, pitch, yaw, and thrust respectively. Thrust, roll, pitch and yaw are each the

combination of three gains k_{-1} , k_{-2} , k_{-3} . These gains will contribute to adjusting the sensitivity of the RC, regulating the fluctuation in inputs, and reducing the roughness of inputs. The following sections describe the tuning of k_{pitch} since k_{roll} , k_{yaw} and k_{thrust} are tuned similarly.

As shown in Figure 9, in order to apply the proposed algorithm and tune the mapping parameters, we extract from the RC the acceleration of each stick, the working range of the sticks, and the number of corrections done while teleoperating the quadrotor, as described in the next sections. This information is fed into an algorithm that tunes the RC inputs into the desired ones; then, these modified inputs are sent to the quadrotor.

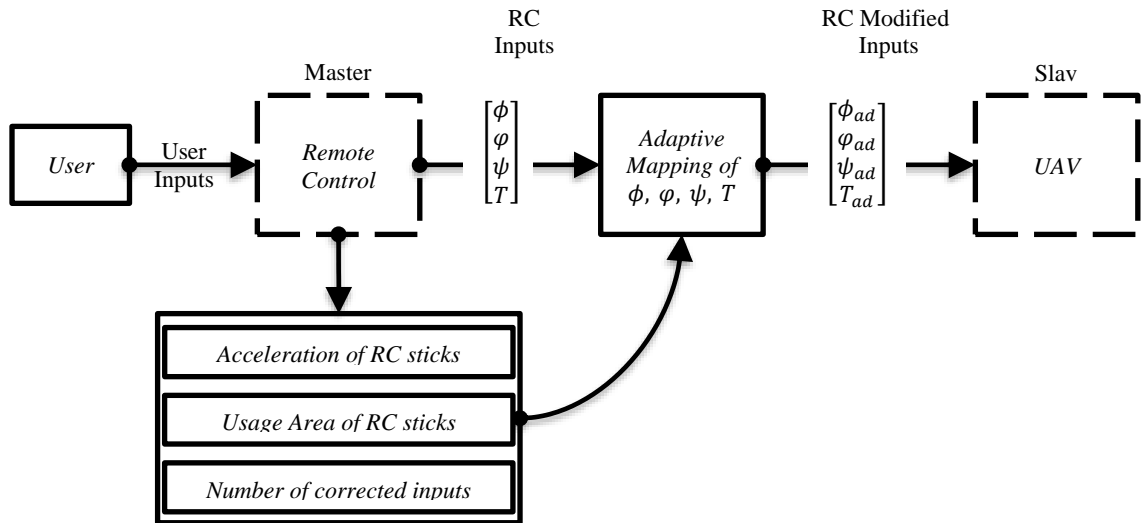


Figure 9: Parameters tuning diagram

1. Adjusting sensitivity

$k_{pitch.1}$ is responsible for adjusting the sensitivity of the RC, as it is used to modify the input range of the joystick. In other words, the variable $k_{pitch.1}$ is modified adaptively in order to limit the range of the joystick inputs. This mapping gain compares

the maximums of the pitch inputs for a given set of inputs (Figure 10), and the variable changes based on these inputs. If the maximum input is less than a preset proportion of the full range for the stick displacement, then $k_{pitch.1}$ increases in order to make the working range wider and in order to reduce the sensitivity of the commands. But for the case where the pilot uses the full range for the commands, $k_{pitch.1}$ increases to increase the sensitivity of commands and to reduce the working range. In other words, this parameter is used to tune the sensitivity of the input. The maximum joystick displacement for pitch motion is measured when there is an active input for the pitch motion. To ensure smooth transition of gain values, we are interested in the average of the maximum over the whole cycle. $k_{pitch.1}$ is calculated using the following formula:

$$k_{pitch.1} = \max \left(k_{pitch.1} min, k_{pitch} Factor * \text{average}(\text{Inputs local maxima}) \right) \quad (20)$$

With $k_{pitch.1} min$ being the minimum allowed value for $k_{pitch.1}$, and $k_{pitch} Factor$ is assigned in order to determine the reference range of interest.

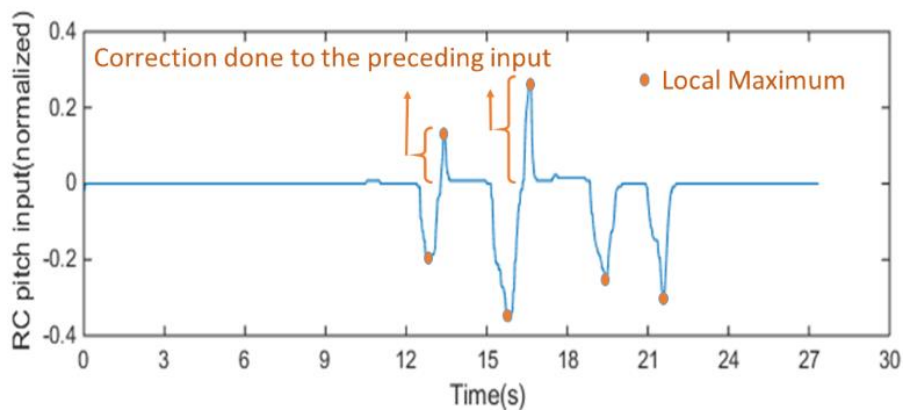


Figure 10: Sample of pitch inputs marked with the local maxima and the corrections done to the preceding inputs

2. *Regulating the fluctuation of input commands*

$k_{pitch.2}$ is affected by the consistency of the pitch commands. Usually, an unskilled pilot applies inputs larger than desired and then corrects them by other smaller and opposite ones (Figure 3). The corrected inputs are applied directly after the main input and tend to have high acceleration in the reverse direction. Parameter $k_{pitch.2}$ is used to reduce such fluctuations in inputs, where this gain is affected by the number of corrections that the users makes. It is scaled from $k_{pitch.2}min$ to 1. For a skilled pilot $k_{pitch.2}$ converges to the maximum and has no effect on the mapping variable k_{pitch} .

$$k_{pitch.2} = \max \left(k_{pitch.2}min, 1 - \left(\frac{corrected\ inputs}{total\ nbr.\ of\ inputs} \right) \right) \quad (21)$$

Where the “total nbr. of inputs” is obtained by summing the number of given inputs (excluding the corrections done to these inputs) over the duration of interest. The number of “corrected inputs” is obtained by summing the corrections in the same period of time.

3. *Reducing the roughness in input commands*

$k_{pitch.3}$ regulates the roughness of joystick commands since a newly introduced pilot to a teleoperation system would be unfamiliar with its mode of operation and would have sudden changes. Sudden input commands are captured by studying the acceleration of the given pitch input. In order to soften these commands, the parameter $k_{pitch.3}$ varies between $k_{pitch.3}min$ and 1. This gain converges to 1 for a trained and skilled pilot and thus there will be no effect from this gain of the input commands. The

value of this gain is based on the ratio between unacceptable accelerations captured over the total set of inputs over the given cycle given by:

$$k_{\text{pitch.3}} = \max\left(k_{\text{pitch.3}min}, 1 - \left(\frac{\text{inputs w/ unacceptable acc.}}{\text{number of inputs}}\right)\right) \quad (22)$$

4. Adaptive Algorithm

The three gains are combined to produce the final input command. $k_{\text{pitch.1}}$ is responsible for adjusting the sensitivity and $k_{\text{pitch.2}}$ is responsible for minimizing the undesired inputs to the system, these two gains are multiplied by the initial input value. Regarding $k_{\text{pitch.3}}$, its main purpose is to reduce sudden inputs, thus smoothing the input is done by applying moving average between current input and previous one based on this gain as follows:

$$\varphi(i) = k_{\text{pitch.3}} \left(k_{\text{pitch.1}} k_{\text{pitch.2}} \varphi_{(\text{unmod.})}(i) \right) + (1 - k_{\text{pitch.3}}) \varphi(i - 1) \quad (23)$$

Where $\varphi(i)$ is the current pitch command while $\varphi(i - 1)$ is the previously applied pitch command. And $\varphi_{(\text{unmod.})}(i)$ is the unmodified current RC pitch input. Similarly, roll (\emptyset), yaw (ψ) and thrust (T) inputs are formulated as:

$$\emptyset(i) = k_{\text{roll.3}} \left(k_{\text{roll.1}} k_{\text{roll.2}} \emptyset_{(\text{unmod.})}(i) \right) + (1 - k_{\text{roll.3}}) \emptyset(i - 1) \quad (24)$$

$$\psi(i) = k_{\text{yaw.3}} \left(k_{3.1} k_{\text{yaw.2}} \psi_{(\text{unmod.})}(i) \right) + (1 - k_{\text{yaw.3}}) \psi(i - 1) \quad (25)$$

$$T(i) = k_{\text{thrust.3}} \left(k_{\text{thrust.1}} k_{\text{thrust.2}} T_{(\text{unmod.})}(i) \right) + (1 - k_{\text{thrust.3}}) T(i - 1) \quad (26)$$

Human operators tend to teleoperate UAVs with low performance at the beginning, but with more practice and long flight duration, their performance improves. Thus, the joystick mapping parameters should adapt to this change in real-time. The three introduced gains ($k_{-.1}$, $k_{-.2}$, $k_{-.3}$) are evaluated over the set of inputs belonging to a moving window with a fixed period of time. Where only the inputs given in the last period of time are used to calculate the gains. Rapid changes of mapping parameters can frustrate the operator and handicap his/her leaning ability as described in [26]. With a large time period for the window, the frustration is minimized as the studied proportion of inputs is large and the transition between the values of mapping parameters is smoothed. Also, in order to have a responsive system that acts based on the latest commands, and this is accomplished by assigning a small-time period for the window. A balance should be achieved between smoothness and responsiveness when selecting the window duration. The gains start to change once this period of time has elapsed. At the beginning of each flight, the window time is bigger than the flight duration itself. As it is required to have enough inputs to cover the window period, the inputs given in the previous trial are treated as they belong to the beginning of the current one. Figure 11 shows an example of the modified RC inputs compared to the original ones. In this paper, the chosen parameters are shown in the experimental design and setup paragraph in section IV.

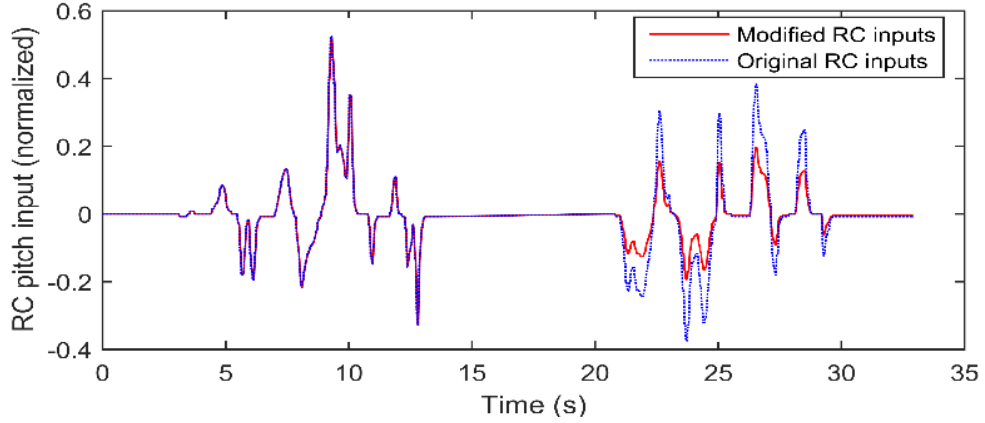


Figure 11: Comparison between the original and modified RC inputs using the proposed algorithm

B. Experimental Setup and Results

Human subject experiment is conducted to assess the proposed method. Test subjects control the UAV to achieve a task using RC with both traditional and adaptive techniques. Ten (10) participants took part in these experimentations distributed as 8 males and 2 females, with ages ranging from 19 to 24 years old. Five of the test subjects were video gamers and five of them were familiar with the concept of thrust, roll, pitch and yaw motions. While only two of them were amateur UAV operators. All the test subjects were selected randomly from the American University of Beirut (AUB) students.

1. Experimental Results and Setup

Experiments were held in a closed environment, and all subjects tested both the conventional mode of operation along with the proposed one. In this experiment, the AscTec Pelican quadrotor was used. In order to evaluate the performance of users, the trajectory of the quadrotor was captured using the Vicon system, where 6 cameras were

spread around the testing area as shown in Figure 12. Using this setup, we were able to capture the distance traveled of the quadrotor, the time of flight, and the accuracy in landing. To ensure the safety of participants, the quadrotor was tested inside a 5x3 m cage. The task of each user was to fly the quadrotor, take it off between two waypoints and then land it on a specified area. As the chosen quadrotor had 38cm of radius, the distance separating the two obstacles was set equal to twice its diameter, making it feasible for novice operators to fly the quadrotor on the desired path. The following values were selected in the experiments: minimum for each gain (k_{-1min} , k_{-2min} , k_{-3min}, \dots) equal to 0.5, window time = 20 seconds, $k_Factor = 2$, acceptable acceleration = $1.8 m/s^2$.

2. Detailed Procedure

Prior to testing, every participant was introduced to the experiment, then each operator practiced the quadrotor teleoperation for 5 minutes for the purpose of familiarizing with the required task. Then, each operator was asked to perform 12 trials (6 for each mode). The selected mode of the first trial was chosen at random, and the rest of the trials were selected by alternating between the two modes. This selection of trials minimizes the effects of training and fatigue, as well it removes any type of biasing. Trials with cage hit or crashes were repeated, and if the number of repetitions exceeded 10 times for the whole experiment, all the data was discarded. If the quadrotor hovered over a waypoint in an experiment, the trial was not discarded, but rather it was marked with a hit. On average, each testing session lasted between 30 to 40 minutes. Performance was evaluated quantitatively using time of flight, distance to target and distance traveled.

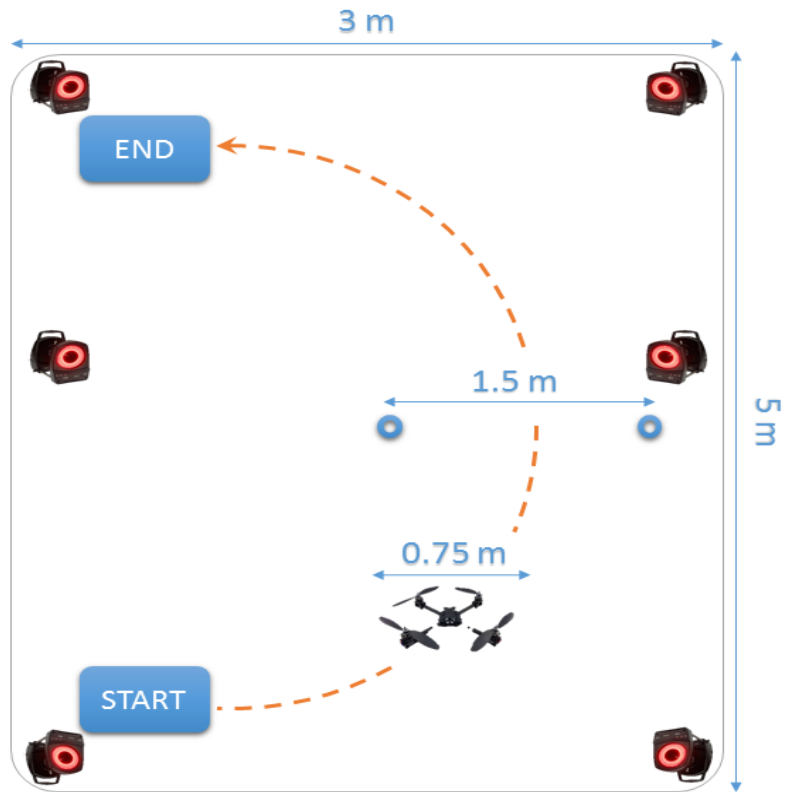


Figure 12: Testing trajectory

3. Testing Results

Figure 13 shows the mean and standard deviation for all the trials of the objective results (flight duration, distance to target, distance traveled) for the two modes of operation. The bars represent the mean value for all the trials, while the standard deviations are displayed as vertical lines on top of the bars.

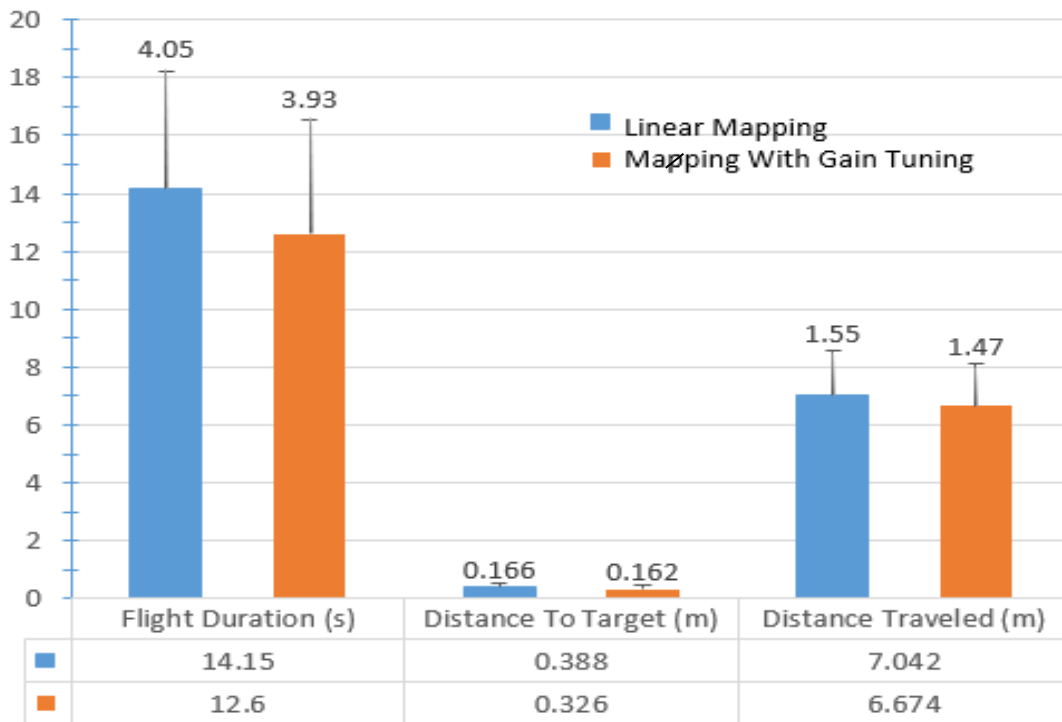


Figure 13: Objective results for all the trials showing the average and the std. dev. of the flight duration, distance to target and distance traveled

In the first mode, when users were teleoperating the quadrotor using the linear mapping for RC parameters, the mean of flight duration was equal to 14.15 seconds. while in the second mode, using the gain tuning to modify the RC inputs, users were able to finish the trials with a mean of 12.6 seconds. Thus, mode 2 reduced the flight duration by 1.55 seconds or 11.59%. As for the distance to target, the mean landing distance from the target position was equal to 0.388 meters in mode 1 compared to 0.326 meters in mode 2 with an 17.51% improvement. As well, the distance traveled was reduced from mode 1 to mode 2 where it dropped from 7.04 meters to 6.674 meters with a 5.37% improvement. Moreover, the standard deviations are smaller in mode 2 compared to mode 1 which suggests that more consistent flights were obtained by modifying the RC input parameters.

Combining the trials of 10 users, yield to 60 trials in each mode. But in both modes, we got 12 trials marked with a hit, which are considered as outliers as in all of them the quadrotor went off the intended trajectory and the user selected by mistake a shorter or a wrong path. Figure 14 shows the mean and standard deviation of the objective results related only to the 48 acceptable trials in each mode.

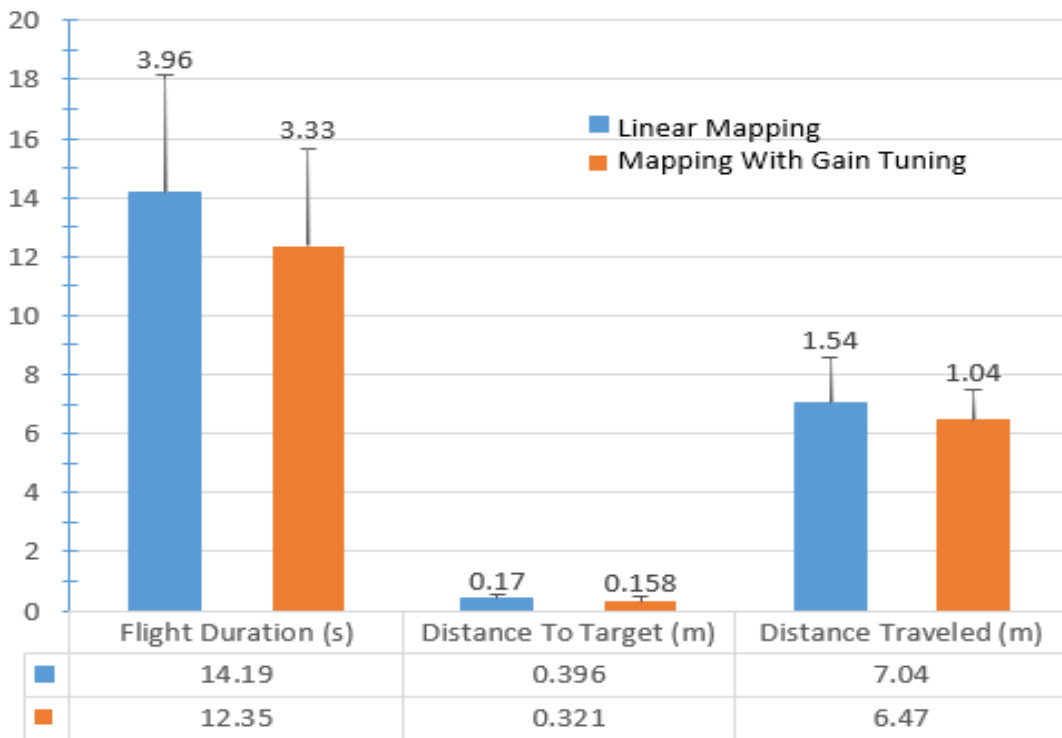


Figure 14: Objective results for the acceptable trials showing the average and the std. dev. of the flight duration, distance to target and distance traveled

By removing the unacceptable trials, the flight duration in mode 1 went from 14.15 meters to 14.19 meters while the standard deviation dropped from 4.05 meters to 3.96 meters. In mode 2 the flight duration was reduced from 12.6 meters to 12.35 meters and the standard deviation was reduced from 3.93 meters to 3.33 meters. This drop in standard deviation shows that by eliminating the trials with hits, more consistent results were obtained. As for the distance to target, distance in mode 1 became 0.396

meters and in mode 2 it became 0.321 meters. Regarding the distance traveled, in mode 1 the mean distance stayed 7.04 meters and in mode 2 it dropped to 6.47 meters. As for the standard deviations, in all the modes it dropped except for the first mode in distance to target it increased to 0.17 meters.

The one-way Analysis of Variance (ANOVA) statistical test is used to analyze the relevance between the obtained means of mode 1 and mode 2. In order to accept or reject the null hypothesis, the resultant p-value is compared to the reference value $\alpha = 0.05$. In other words, we can conclude that our results are statistically significant by obtaining a p-value lower than α .

The difference in percentage between the linear mapping of parameters and the mapping with gain tuning is displayed in Table I. As well Table I provides the p-values related to each characteristic extracted from the experiments. And it shows that commanding the quadrotor with the modified inputs for the RC gave better results in terms of flight duration, accuracy and distance traveled. We obtained 13.86% improvements in the flight duration when using the proposed method with a 0.0156 p-value. Distance to target dropped by 20.71% in the second mode with 0.0291 p-value. And the distance traveled was reduced by 8.44% in the second mode with a p-value equal to 0.0379.

Table 1: Difference between the two modes and ANOVA test results

		Flight Duration	Distance to Target	Distance Traveled
All Trials	difference	-11.59%	-17.51%	-5.37%
	p-values	0.0290	0.0399	0.1696
Acc. Trials	difference	-13.86%	-20.71%	-8.44%
	p-values	0.0156	0.0291	0.0379

CHAPTER V

INTENTION RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK

The adaptive gain tuning approach uses three subjectively tuned gains that must be tailored differently based on the design criteria. Therefore, there was a need for a more systematic approach. We chose to use a machine learning system which is robust to noise and fast in classification. In that the machine-learning algorithm learns for different operators their own weights, which correspond to their perceptual mapping of a desired motion to a given control. The training is done by prompting the user with known motions; then measuring the corresponding given commands by the user and using them as inputs to the classifier.

Our problem was solved using both ANN and CNN and the classifier that gave the higher classification accuracy was adopted. But first, in order to train both classifiers, we need a labeled dataset. Figure 15 describes the problem explicitly.

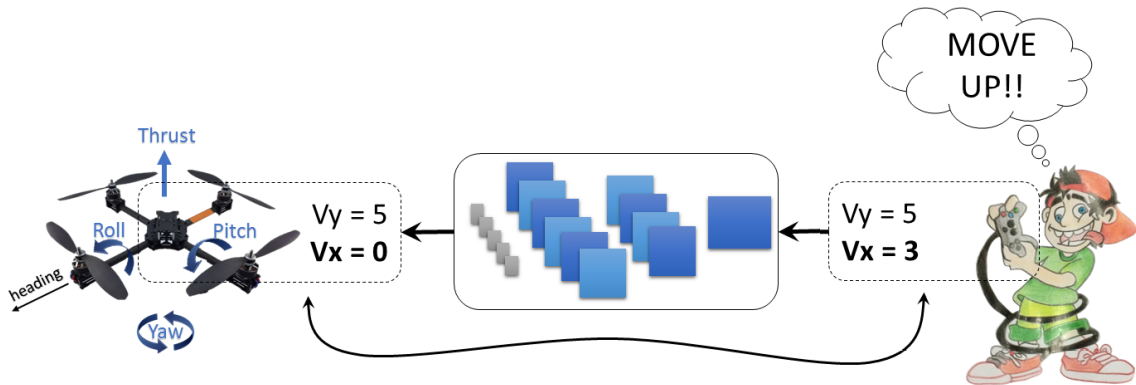


Figure 15: Motivation for the problem: The user inputs a command to the joystick intending to direct it in a vertical direction; however, his commands are flawed and would result in forward motion V_x in addition to the intended V_y . A classifier is trained to decipher his true intention and command the drone with the correct motion, setting V_x to zero

A. Dataset Collection

In order to build the personalized teleoperation system, it should be well designed to recognize the operator's motions. To do so and train it to adapt to each user, a dataset containing the operator motions should be built. Once all the data regarding the operator motions is collected, the dataset is used to train a classifier which is later used to classify any new inputs into their desired outputs.

The dataset is collected as follows:

- Every operator is asked to build his own dataset that is used to train the classifier responsible for recognizing the user gestures.
- The input devices are two 3D haptic devices. The user has the choice of performing the desired gesture using either one of the haptic devices or both.
- The collected data represents the position of the tip of the two haptic devices along with the time stamp of each collected point in 3D. The number of collected points is affected by the sampling period of the computer and the device itself. On average 500 3D points are captured from each haptic device for each performed gesture.

- The dataset contains the gestures used to perform X motions with Y velocities each, in our case X is 32 and Y is 2 (slow and fast motions) and in total there are 64 classes.
- Every user is required to perform 20 trials per class, in total every user performs 1280 gestures. The 20 trials of each gesture to be performed are divided into 4 rounds each containing 5 trials. When collecting the dataset, the order of the gestures to be performed was selected randomly in each round. This randomization in the selection of gestures is implemented to minimize the fatigue effect of the operators.

B. Intention Recognition Using Neural Networks

1. ANN Architecture

ANN is a powerful data-driven classifier that has the capability of studying and predicting the output of nonlinear and complex systems. such systems are trained form large set of data, and once trained, they can excel is areas where feature is hard to achieve. Neural networks have been used to solve a wide variety of tasks, tasks that are hard to solve using ordinary rule-based programing like computer vision.

In our system, the collected datasets contain thousands of points representing the 64 predefined classes. Classifying an input into 64 classes may result into poor classification results. In our case, instead of obtaining one output that describes the desired thrust, roll, pitch and yaw commands. It is possible to implement four networks that describes each one of the four UAV inputs. By implementing this design, each neural network output could fall into one of 5 classes. These classes represent the

magnitude of the output which is distributed between null, high and low in the positive and negative directions.

The used ANN consists of 4 layers each having 60 neurons. The final layer is composed of a regression layer mapping the input matrices to an output between -1 and +1 that represents the desired direction and magnitude. We used the hyperbolic tangent as an activation function between these layers. The input is a vector containing 30 variables representing 5 consecutive input velocities of the right and left joysticks. Figures 16 and 17 represent respectively the ANN inputs and the ANN architecture.

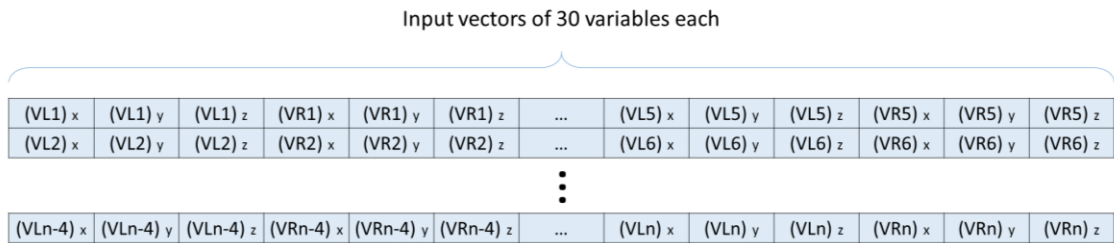


Figure 16: ANN consecutive inputs

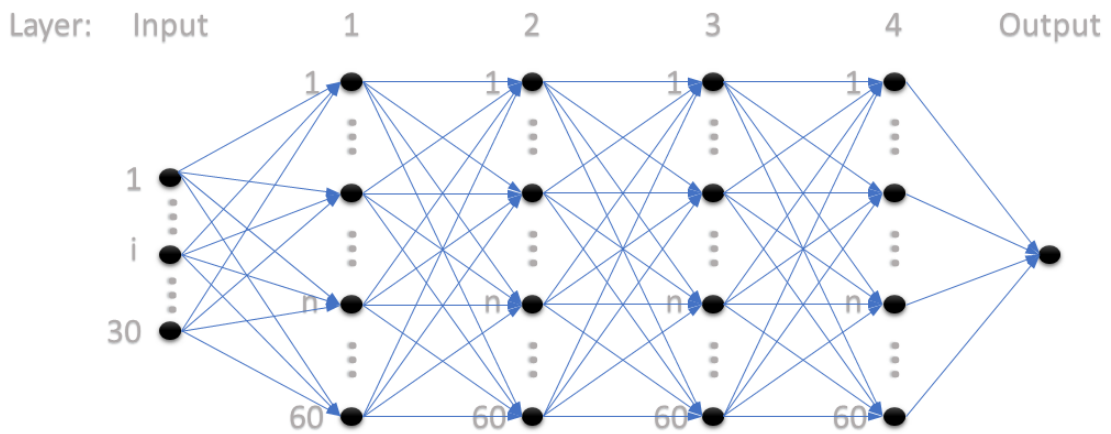


Figure 17: ANN architecture

2. CNN Architecture

CNNs are usually used to classify soundtracks, images, and videos; however, in this application we are interested in classifying joystick inputs. We architect and compare two different representations for the input feature vector: the first one is based on a temporal stratification of inputs, and the second is a mapping from joystick trajectory to a corresponding 2D image.

a. Temporal stratification representation

The first type of input to the CNN consists of layering, at each time stamp, five velocity vector inputs in chronological order—the current and past four. Table 2 shows an example of such an arrangement, where the first row displays the state of the current velocity, and each ensuing row represents the state immediately preceding the other in time. Each row represents the 3D velocities of the left and right haptic interfaces, for a total of six inputs per row.

Table 2: Example input for the stratified representation

$VL_x(n)$	$VL_y(n)$	$VL_z(n)$	$VR_x(n)$	$VR_y(n)$	$VR_z(n)$
$VL_x(n-1)$	$VL_y(n-1)$	$VL_z(n-1)$	$VR_x(n-1)$	$VR_y(n-1)$	$VR_z(n-1)$
$VL_x(n-2)$	$VL_y(n-2)$	$VL_z(n-2)$	$VR_x(n-2)$	$VR_y(n-2)$	$VR_z(n-2)$
$VL_x(n-3)$	$VL_y(n-3)$	$VL_z(n-3)$	$VR_x(n-3)$	$VR_y(n-3)$	$VR_z(n-3)$
$VL_x(n-4)$	$VL_y(n-4)$	$VL_z(n-4)$	$VR_x(n-4)$	$VR_y(n-4)$	$VR_z(n-4)$

The CNN architecture for the stratified representation consists of an input layer, two convolutional layers, two fully connected layers, and a classification layer (See Figure 18). The first convolution layer uses eighty filters, each having a dimension 1x3. The purpose of this filter is to relate the velocity inputs of the two joysticks. The second

convolutional layer, also uses eighty filters but with a dimension of 2×1 . The purpose of this filter is to relate the consecutive joystick inputs. The two fully connected layers consist of thirty neurons each. The final layer is composed of a regression layer mapping the input matrices to an output between -1 and +1 that represents the desired direction and magnitude.

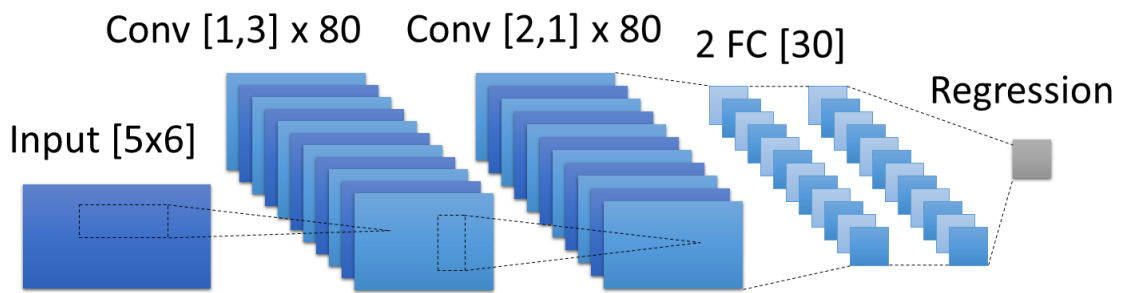


Figure 18: CNN architecture for the stratified representation

b. Representation by a 3-channel input

Mapping the user inputs could be also achieved by transforming the inputs of each joystick into a three-dimensional matrix analogous to an RGB image. Each of the three channels represents the projection of the 3D trajectory of the haptic device on one of the three 2D spaces of an orthonormal coordinate frame, including the x-y, x-z, and y-z spaces. To ensure a compact representation of the data, the right and left inputs are concatenated side by side in each space as shown in Figure 19. The total distance in the x, y, and z directions for the right joystick is represented respectively by $P_r x$, $P_r y$, and $P_r z$. Similarly, for the left joystick they are represented by $P_l x$, $P_l y$, and $P_l z$.

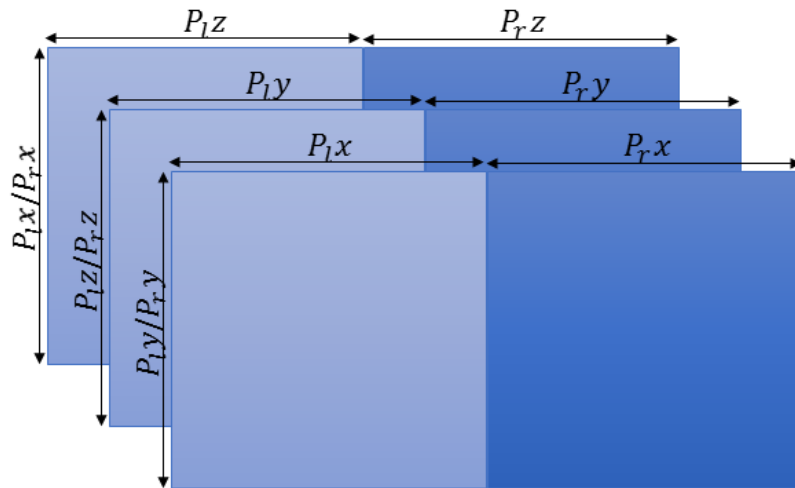


Figure 19: The three-dimensional input matrix with depth equal to 3

To reduce the computational requirements of the CNN we focus, for each motion, on a local neighborhood, rather than the entire space. The size of the neighborhood is dependent on the sought-after resolution. The proposed input matrix is composed of $2n + 1$ rows and $2 * (2n + 1)$ columns with the resolution of r mm. The input trajectory includes the displacement of the haptic device in the current time step, along with the previous displacements in a range of $r * n$ mm in the positive and negative directions.

The matrices are initialized such that they map part of the joysticks' workspace. At first the matrices are empty. As the operator moves the joysticks, the corresponding cell in the matrix is filled with the time elapsed to form this maneuver. An example of one depth of this matrix (with a smaller size) is shown in Figure 20.

Finally, each input is mapped to an 8-bit number, similar in spirit to the intensity value of a single image channel (Figure 21). This turns out to be helpful in using an off-the-shelf CNN, which takes images as inputs.

$$V_{new_{x,y,z}} = \text{round} \left(\max \left(0, \left(V_{old_{x,y,z}} - V_{minNZ} \right) \right) \right. \\ \left. * \left(255 / (V_{max} - V_{minNZ}) \right) \right), \quad (27)$$

where $V_{new_{x,y,z}}$ is the new value for a specific cell in the input matrix, and $V_{old_{x,y,z}}$ is the previous value of that cell. V_{max} indicates the maximum value of the cells in the 3D input matrix and V_{minNZ} represents the non-zero minimum value of these cells.

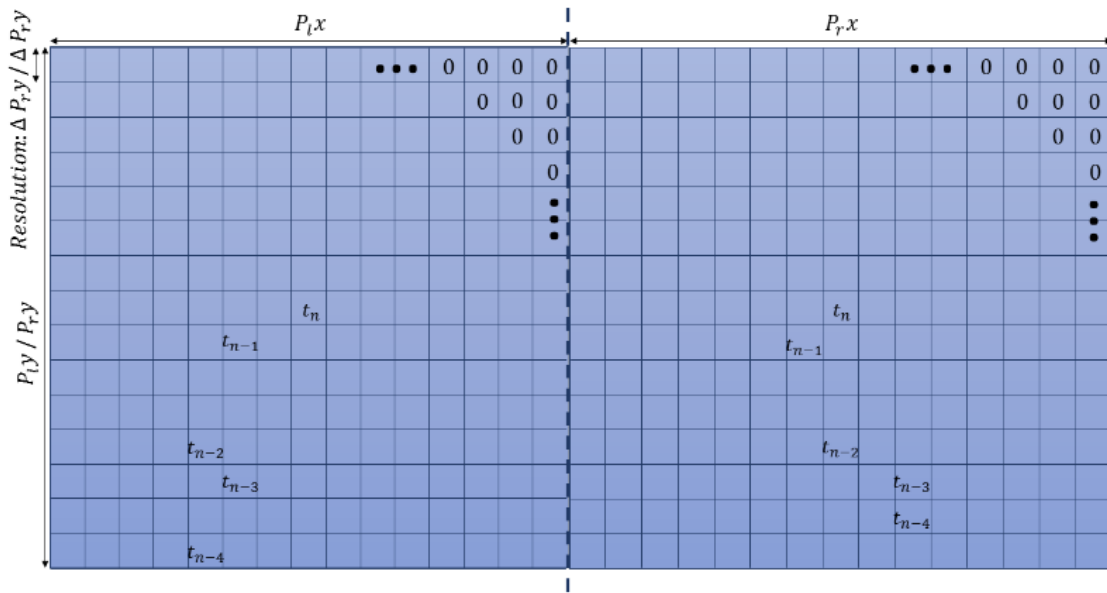


Figure 20: Representation of one depth of the input matrix

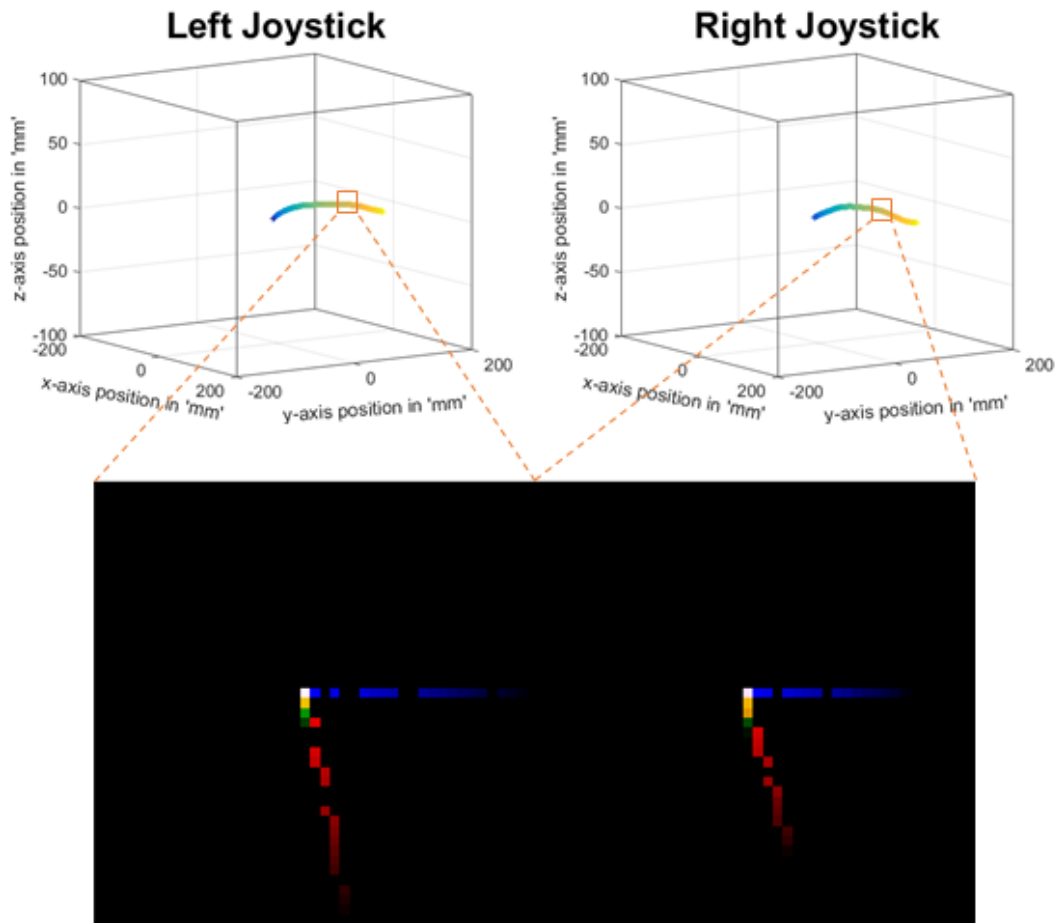


Figure 21: Sample of the RGB input matrix representation

The CNN for this representation is constructed of two convolutional layers, each followed by a max-pooling layer, as shown in Figure 22. The convolutional layers consist of forty and eighty filters, with 8-by-8 and 4-by-4 receptive fields, respectively. Each max-pooling layer pools 2-by-2 regions at strides of two pixels. The convolutional layers are followed by two fully connected hidden layers, having two hundred units each. The final layer is composed of a regression layer mapping the input matrices to an output between -1 and +1 that represents the desired direction and magnitude. Both convolutional layers use the rectified linear activation function.

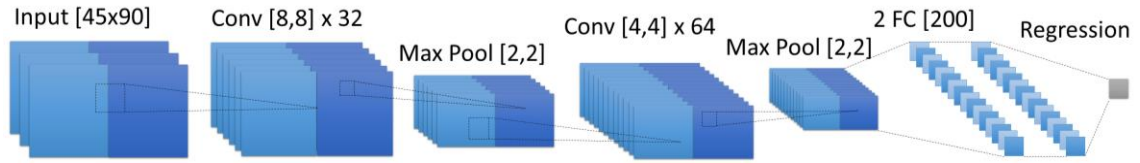


Figure 22: CNN architecture for 3-channel input representation

C. Dataset Collection and Classification Experimental Setup and Results

During the experiment, twelve test subjects used two haptic devices to perform their gestures. The subjects consisted of nine right-handed and three left-handed male persons, with ages ranging from twenty to twenty-seven years old. Four of the test subjects were video gamers and nine of them were familiar with the concept of thrust, roll, pitch and yaw motions; only three of them were amateur UAV operators. All the test subjects were selected randomly from the students at the American University of Beirut (AUB).

1. *Format of the input feature vector*

During the experiments, the user had the choice to use one or both haptic devices to perform any given input; the recorded motion is that of the tip of these devices, along with the time stamps of each collected point in 3D. The number of collected points is dependent on the sampling period of the computer, as well as that of the device itself. On average five hundred 3D points are captured from each haptic device for each performed gesture.

Figure 23 shows an example for a trajectory performed by a user on the right joystick for a 'right' motion command. Note how for the same person, the same intended motion is not a perfect replicate.

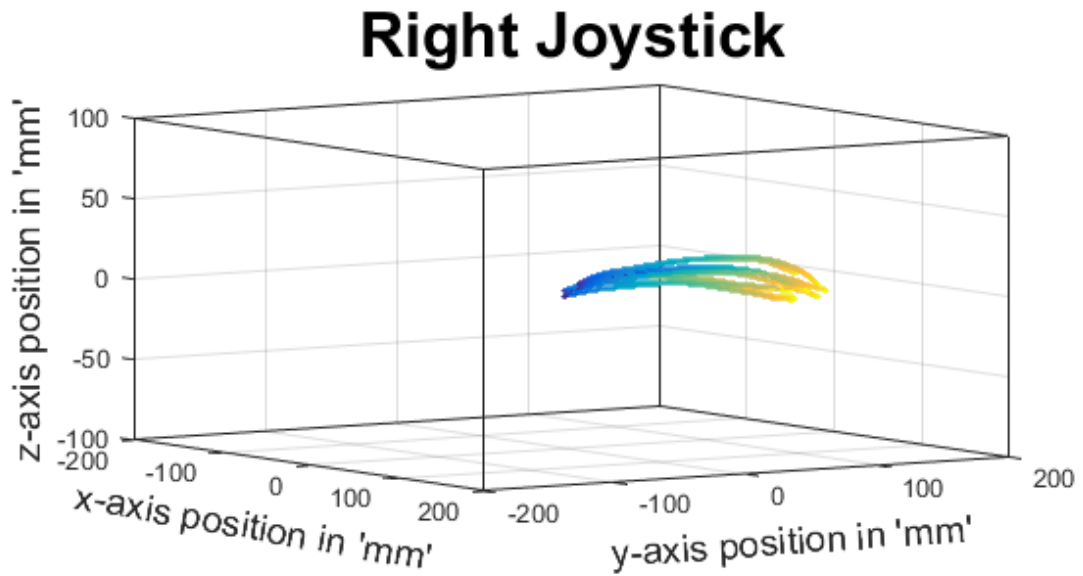


Figure 23: Example of a motion in the right direction on the left haptic device

2. *Experimental Design and Setup*

Prior to testing, every participant was introduced to the experiment, and then the operators took 5 minutes to familiarize themselves with the experiment. Every participant was free to apply any gesture on the left haptic device, the right one, or both. Experiments were held in a closed environment, and all the subjects were free to leave and stop the experiment, whenever they wanted to. The two haptic devices used in this experiment are the Sensable^{TD} Phantom.

Every user was required to perform twenty trials per class, for a total of 1280 gestures. The twenty trials were applied in four rounds, each containing five trials. When collecting the dataset, the order of the applied gestures was selected in a random manner; this was done to minimize the fatigue effect on the test subjects.

On average, each session of dataset collection lasted between 120 to 150 minutes. During each session, the participants were asked to take a five-minute break between

each continuous thirty minutes of data collection. Once the dataset was acquired, it was preprocessed (as shown in section B) and then fed to the CNN. Eighty percent of the data was used for training and validation, and twenty was used for testing the CNN. In other words, for each specific motion and velocity, sixteen gestures were used for training and validation, and four gestures were used for testing.

While preprocessing the data, "n" and "r" were selected equal to 20 and 0.3 respectively. With these values, the dimension of the input matrix became equal to 45x90x3, as for the resolution between one cell and the other it is equal to 0.35 mm. With these values, the neighborhood of interest in one matrix is equal to 7 mm. These values could be selected differently based on the input and application. By increasing "r", the resolution increases, but it would require a larger matrix to describe the same neighborhood area in one matrix. As well by increasing n, the system would account for a larger neighborhood, but it would be harder to track sudden changes in the input.

3. Results

Three classifiers are proposed in our work, 2 CNNs and 1 ANN. Our best candidate is obtained based on the higher classification accuracy.

a. ANN vs Temporal Stratification Representation

ANN is compared to the temporal stratification representation architecture to assess the classification percentages obtained. Only the dataset of the first two subjects were selected for assessment. Table 3 displays the comparison between the two architectures, where the classification results for the up/down, left/right, forward/backward, and clockwise/counterclockwise rotations are displayed in 4 separate

columns. The final column in the table above displays the classification percentages for the four movements to be classified correctly at the same time. CNN proves to be better than ANN in all scenarios. That's why CNN is adopted for classification, but we have to decide which architecture suits our system best.

Table 3: ANN vs CNN classification results for the participants

Subjects	Classifier	Up / Down	Left / Right	Forward / Backward	CW / CCW Rotations	All the inputs
1	ANN	93.2%	96.1%	94.6%	93.9%	86.0%
	CNN	94.3%	97.3%	95.5%	94.0%	86.9%
2	ANN	84.3%	89.4%	81.4%	98.3%	73.9%
	CNN	85.7%	90.4%	82.3%	98.5%	74.8%

b. Temporal Stratification Representation vs Representation by a 3-channel Input

Table 4 displays the classification results for the two different input representations described above. Columns 3 to 6 display the classification results for the four CNNs representing the principal motions. The last column of Table 4 represents the percentage classification for the case where the same input is classified correctly by the four CNNs.

It is shown that the average classification accuracy is higher for the second CNN architecture, where nine out of the twelve subjects scored higher classification accuracy. On average, the accuracy in classification results increased from 76.24% to 78.56%. For Subjects 5, and 8, the first CNN architecture gave better results, which can be attributed to the selection of the specific values of "n" and "r". As each subject has his own system and his own dataset, tuning the resolution "r" of the input images could improve the classification percentages.

Table 4: Classification results comparison for the two proposed CNN architectures

Subjects	CNN arch.	Up / Down	Left / Right	Forward / Backward	CW / CCW Rotations	All the inputs
1	1	94.3%	97.3%	95.5%	94.0%	86.9%
	2	96.5%	97.9%	96.8%	97.3%	90.7%
2	1	85.7%	90.4%	82.3%	98.5%	74.8%
	2	86.4%	91.7%	86.1%	98.9%	79.3%
3	1	80.4%	87.6%	84.3%	97.1%	68.7%
	2	84.5%	90.0%	86.2%	97.0%	72.0%
4	1	88.6%	93.5%	91.9%	97.4%	82.7%
	2	91.8%	96.8%	93.5%	97.7%	86.1%
5	1	94.2%	96.7%	94.6%	95.4%	85.0%
	2	92.9%	96.0%	93.3%	95.0%	82.7%
6	1	88.1%	87.6%	78.7%	98.6%	69.3%
	2	88.2%	90.6%	79.2%	98.3%	70.8%
7	1	84.5%	92.0%	81.5%	98.4%	69.2%
	2	84.7%	92.8%	81.6%	98.9%	69.5%
8	1	86.2%	90.2%	77.3%	97.0%	65.4%
	2	85.4%	87.7%	75.4%	97.8%	65.1%
9	1	88.2%	88.1%	85.1%	97.9%	73.7%
	2	91.8%	91.8%	89.1%	97.7%	78.5%
10	1	94.0%	93.2%	89.6%	98.8%	81.5%
	2	95.0%	94.1%	91.0%	99.1%	84.1%
11	1	99.3%	98.7%	99.0%	99.9%	97.8%
	2	99.5%	98.9%	98.7%	99.9%	97.9%
12	1	93.0%	96.8%	90.0%	96.3%	81.4%
	2	95.0%	97.3%	91.1%	97.8%	85.4%
Average	1	88.84%	92.13%	86.44%	97.22%	76.24%
	2	90.20%	93.33%	87.58%	97.77%	78.56%

As it is visible from Table 4, User 8 has a 65.1% of accuracy for all the inputs in the second CNN architecture, while User 11 has a 97.9% of accuracy for all the inputs. It was observed that low accuracies were associated with users who performed similar motions for different desired outputs and therefore made it difficult to disambiguate between them. It was also observed that better results were obtained when the input motions were close to straight lines, reducing ambiguity in the classification of an intention. As we are taking a small portion of the input at each instance to classify it, if

the total input is curvy (for example forming a circular shape), each portion of the input may be pointing to a different direction.

The first trial of each motion from the dataset is displayed in Figures 24 and 25 representing respectively the inputs of Users 8 and 11. The inputs of User 11 are spread all over the area, and are as well straight lines. These characteristics of the dataset made it easier for the CNNs to classify the inputs and consequently we obtained 97.9% accuracy. The inputs of User 8 are close to each other and appear to be similar, although each one belongs to different class. Also, the inputs are curvy and far from representing straight lines. The combination of the curved and grouped inputs lead to the poor accuracy in classification that was equal to 65.1 %.

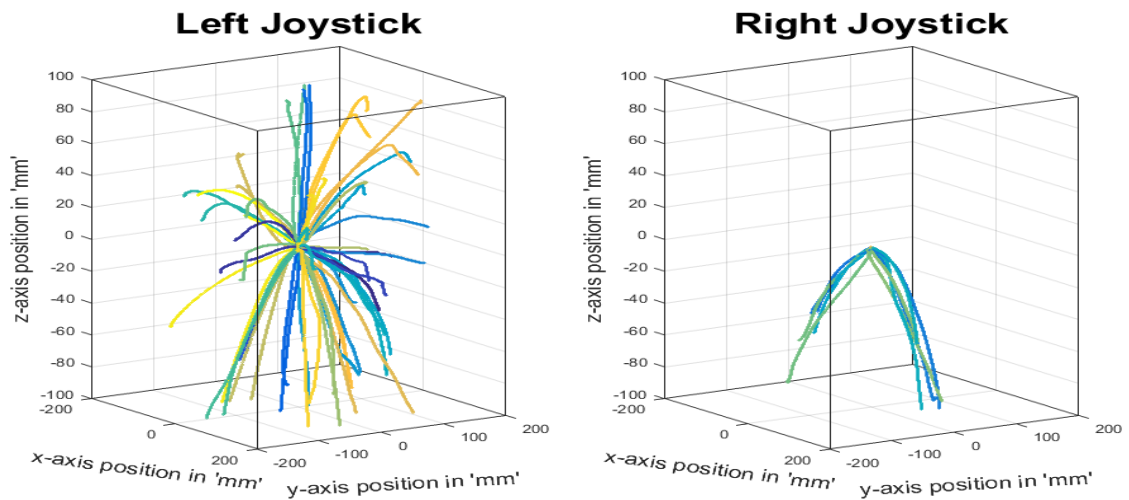


Figure 24 : First trial of each motion for subject 8

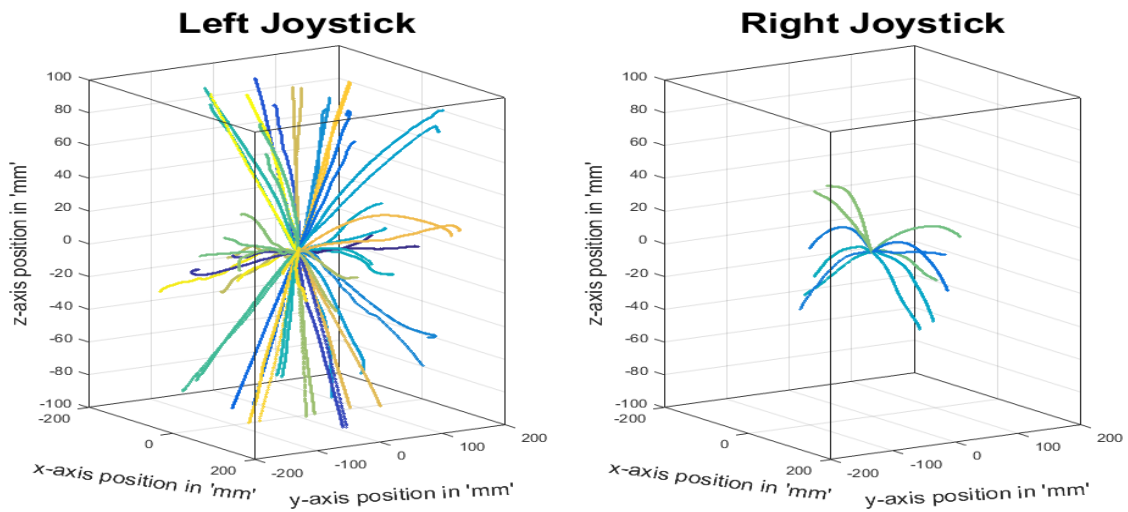


Figure 25: first trial of each motion for subject 11

The confusion matrices for the classification results of user 11 are displayed in Figure 26. The results displayed in this figure belong to 20% of the dataset, and they represent 176414 input matrices. As shown, most of the misclassified inputs lie in the neighbor classes. This means that a misclassified input will not have a significant effect, due to the small classification error. To do further investigations on the confusion matrix, we calculated the distance of the misclassified inputs and we compared this distance to the total number of classified inputs; if an input is classified with the desired class, the error distance becomes equal to 0, whereas the error distance of an input classified with class 1 instead of class 3, the error distance becomes 2. In our case, the average of the error distance of the four CNNs' outputs are equal to 0.0064, 0.0121, 0.0146, and 0.0003.

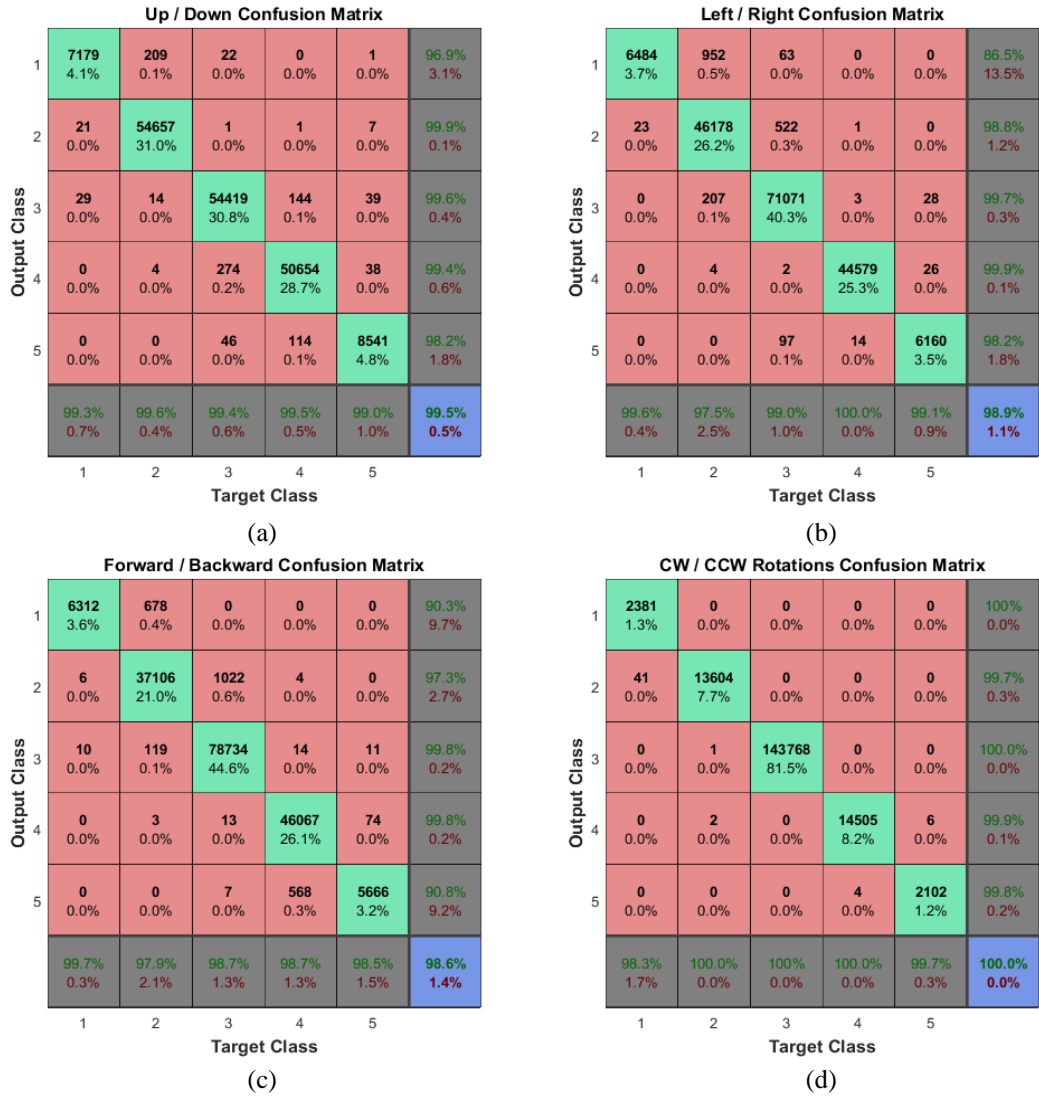


Figure 26: Confusion matrices that describe the classification inputs of user 11 - The five classes in this figure represent the following motions and velocities: 1 - negative motion with high velocity, 2 - negative motion with low velocity, 3 - no motion, 4 - positive motion with low velocity, 5 - positive motion with high velocity

A geometry based classifier was proposed in order to assess the classification percentages in Table 4. This classifier is trained using the same input architecture as the first proposed CNN architecture. The input is a vector containing thirty variables, representing five consecutive input velocities containing six variables each. Each six variables hold the linear velocities of the two haptic devices. Also, the output of this classifier represents the states of the principal direction.

This classifier is trained by building a table containing rows of 30 variables. These variables represent the last five velocity inputs in the x, y, and z directions for the left and right joysticks.

Therefore, we have in our table 30 columns containing the variables that are going to be compared with each input, and we have 64 columns symbolizing the possible motions and velocities.

In order to obtain the desired values for each row, all the training inputs are grouped by motions and velocities. Then, a vector containing the average values is calculated for each group and stored in the table and labeled with the desired classification classes. Afterwards, the classification of the new inputs is calculated by comparing the input vector to each row of this table. The row giving the lowest RMS error between all the table rows is chosen and the label of this row is selected as the desired output. The classification results obtained using this method are displayed in Table 5.

Table 5: Classification results for the table based classifier

Subjects	Up / Down	Left / Right	Forward / Backward	CW / CCW Rotations	All the inputs
1	89.4%	92.45%	85.2%	97%	79.1%
2	79.2%	85.2%	78.3%	95.7%	66.7%
3	78%	84.2%	80.9%	94.3%	62.9%
4	91.2%	92.8%	93.1%	97%	83.6%
5	85%	97.7%	87.6%	88.8%	65.4%
6	77.8%	83.5%	77.1%	96.6%	59.5%
7	80.2%	88.6%	77.3%	97.7%	62.2%
8	80%	87%	70.8%	95.3%	58.7%
9	82.3%	86%	77.8%	91.1%	61.5%
10	75.1%	84%	73%	97.1%	55.9%
11	94.7%	91.7%	94%	98%	83.7%
12	85.89%	89.4%	79.1%	88%	64.8%
Average	83.97%	88.5%	81.18%	94.72%	67%

The average accuracy obtained using the geometry based classifier is equal to 67%. The average accuracy of the CNN using both architectures is equal to 76.24% for the first architecture, and 78.56% for the second one.

D. Testing Experimental Setup and Results

A statistical study of the participant's flight performance is presented in this section. Objective and subjective assessment is studied and the significance of the CNN based teleoperation system is assessed.

1. Experimental Design and Setup

To assess the proposed teleoperation method every participant had to perform 6 trials in 3 modes. Testing was done using the Parrot AR drone 2.0 quadrotor. The three modes are formed by the CNN based teleoperation system, the velocity based teleoperation system, and the conventional teleoperation. The test subject is required to teleoperate the UAV using these 3 modes in a predefined trajectory using a testing strategy that minimizes the biasing effect. And at the end of each trial, each subject is required to fill a small questionnaire that describes his opinion regarding his performance in each trial.

a. Modes of Teleoperation

i. Mode 1: CNN Based Teleoperation System

In this mode, the UAV is teleoperated using the trained CNNs. As the representation by a 3-channel input gave the higher accuracy, it was used to map the

subject inputs to the UAV using their personal networks. While teleoperating the UAV, four networks are operating in the regression mode as shown in Figure 27. The magnitude of the output level is personalized based on the training data.

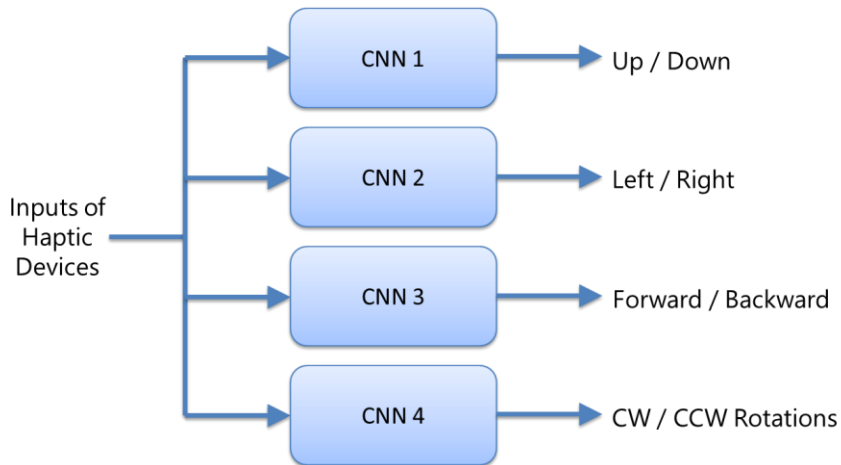


Figure 27 : CNNs used to map user inputs to the UAV

The user commands are applied using two haptic joysticks. The motions are assigned based on each subject dataset. The commands are activated by pressing a button on the joystick end-effector to allow the hand to move freely between consecutive inputs. Another button is used to allow the UAV to take off and land.

ii. Mode 2: Velocity Based Teleoperation System

In the literature, the performance in teleoperating a UAV was highly improved while using haptic joysticks [8-10]. Multiple modes were implemented, and the mode with the highest success rate was the one with velocity based commands. Based on that, we wanted to compare our proposed method to this special one as it proved to be one of the best proposed teleoperation systems.

In the previously proposed velocity based teleoperation system, the joystick movements along the x, y, and z axes directly control the UAV velocity in the x-axis, y-axis, and z-axis directions respectively as described in (28) to (30). In our work, we added an extra joystick responsible of the yaw movement as described in (31)

$$V_x = k_x * J_r V_x \quad (28)$$

$$T_y = T_{UAV} + k_y * J_r V_y \quad (29)$$

$$V_z = k_z * J_r V_z \quad (30)$$

$$\psi = k_{yaw} * J_l V_z \quad (31)$$

Where k_x , k_y , k_z , and k_{yaw} are constants used to adjust the sensitivity of the velocity commands. T_{UAV} is the current thrust value that is used to maintain the UAV at its current altitude. $J_r V_x$, $J_r V_y$, and $J_r V_z$ are the components right joystick's velocity vectors. While $J_l V_z$ is the left joystick velocity in the z direction. V_x and V_z are the UAV velocities in the x and z directions while T is the UAV collective thrust and ψ is the yaw command.

Additionally, to detect only the required commands, a button on the end effector should be always pressed whenever the operator wants to send a command to the UAV. This allows the users to move their hand freely between consecutive commands. The takeoff and landing were activated using another button on the haptic joystick.

iii. Mode 3: Conventional Teleoperation System

In this mode, the UAV is commanded using the conventional method used to control the Parrot AR drone 2.0 which is the phone. The application showed in Figure 28 is available on android and IOS and it is built by the parrot team [35]. This mode is

used to assess the obtained results in the first two modes and to show which mode is better in teleoperating the UAV.



Figure 28: Parrot AR drone 2.0 teleoperation application

b. Testing Area

A testing area is built with a well-defined trajectory to assess the performance of each operator in all the modes as shown in Figures 29 and 30. In this experiment, every user had to takeoff, move the UAV through two wooden frames, and at the end land on a targeted area.

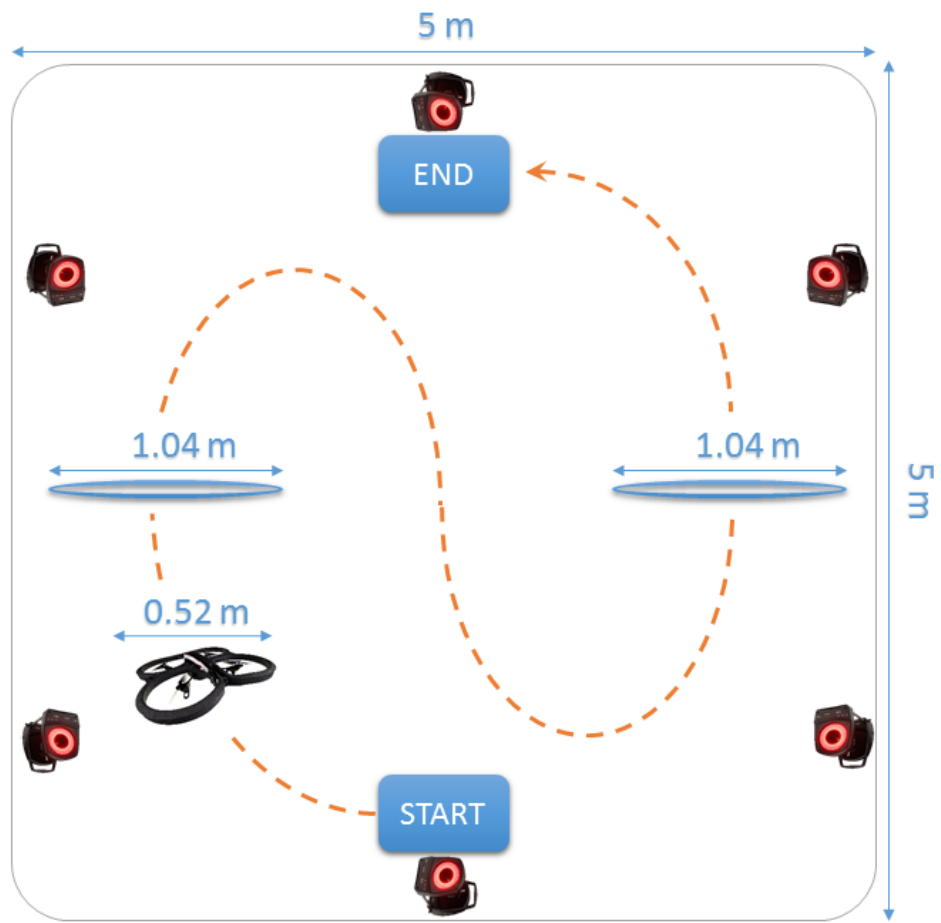


Figure 29: Flight testing trajectory



Figure 30: Image showing the testing area

c. Testing Design and Procedure

Prior to testing, every participant was introduced to the experiment, and then the operators took five minutes to familiarize with each mode of the experiment.

Experiments were held in a closed environment, and all the subjects were free to leave and stop the experiment, whenever they wanted to. The two haptic devices used in this experiment are the Sensable^{TD} Phantom Omni.

Every user is required to perform 6 trials per mode. Modes first trial sequence is selected randomly, and then the following trials were chosen with the same order. For every trial, we are interested in objective and subjective metrics. The objective metrics are the time elapsed in each trial, the distance traveled, and the accuracy in landing. Subjective metrics are obtained at the end of each trial and are taken using a survey based on the NASA Task Load Index (NASA-TLX). This self-assessment score based survey is widely used in human factors research [10] (see Figure 31). The workload perceived by the test subject is spread over 6 subscales dimensions: Mental Demands, Physical Demands, Temporal Demands, Own Performance, Effort and Frustration [36].

Regarding the hardware, as previously mentioned, in this experiment we used the AR drone 2.0 quadrotor along with the Sensable^{TD} Phantom Omni haptic joysticks. VICON is used to track the quadrotor in the held experiments, its data are analyzed to obtain the time elapsed, the distance traveled, and the accuracy in landing. As for the connections, the joysticks are connected to PC1 using firewire connection, VICON is connected to PC1 using Ethernet cable. PC1 is connected to PC2 through an Ethernet cable and the connection is established using TCP/IP protocol. PC1 is responsible for collecting the input data from the joysticks and the VICON system while PC2 is responsible for mapping the inputs using the built CNN networks and transferring the

obtained command to the UAV using a WIFI connection. Figure 32 shows the connections made. The tasks were divided between two computers in order to increase the frame rate.

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.



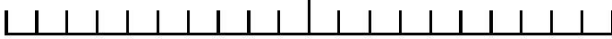



Name	Task	Date
<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;">Mental Demand</div> <div style="width: 70%;">How mentally demanding was the task?</div> </div>  <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Very Low Very High </div>		
<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;">Physical Demand</div> <div style="width: 70%;">How physically demanding was the task?</div> </div>  <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Very Low Very High </div>		
<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;">Temporal Demand</div> <div style="width: 70%;">How hurried or rushed was the pace of the task?</div> </div>  <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Very Low Very High </div>		
<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;">Performance</div> <div style="width: 70%;">How successful were you in accomplishing what you were asked to do?</div> </div>  <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Perfect Failure </div>		
<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;">Effort</div> <div style="width: 70%;">How hard did you have to work to accomplish your level of performance?</div> </div>  <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Very Low Very High </div>		
<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;">Frustration</div> <div style="width: 70%;">How insecure, discouraged, irritated, stressed, and annoyed were you?</div> </div>  <div style="display: flex; justify-content: space-between; margin-top: 5px;"> Very Low Very High </div>		

Figure 31: NASA Task Load Index

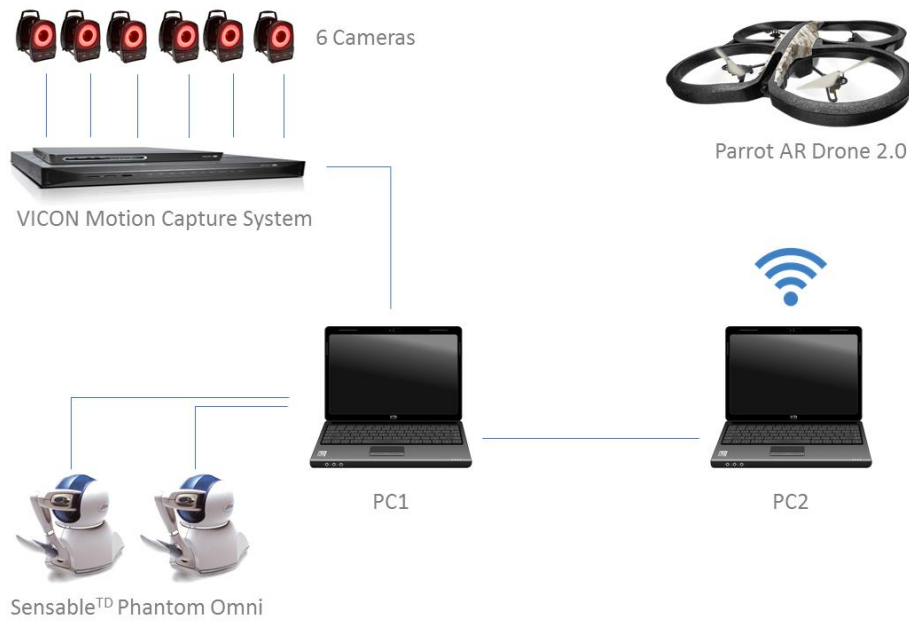


Figure 32: hardware connections

2. Testing Results

Ten out of the twelve subjects that participated in the dataset collection were able to perform this experiment. Subjective and objective results are recorded and studied.

a. Objective Results

Figure 33 shows the mean and standard deviation for all the trials of the objective results (flight duration, distance to target, distance traveled) for the three modes of operation. The bars represent the mean value for all the trials, while the standard deviations are displayed as vertical lines on top of the bars.

In the 1st mode, when users were teleoperating the quadrotor using the CNN based teleoperation system, the mean of flight duration was equal to 35.4 seconds. In the 2nd mode, using the velocity input based teleoperation system, users finished the trials with a mean of 36.1 seconds. And in the 3rd mode, using the conventional controllers,

average trajectory time is equal to 47.5 seconds. Thus, 1st mode reduced the flight duration by 0.683 seconds or 1.91% compared to 2nd mode and by 12.082 seconds or 28.94 % compared to 3rd mode (as shown in Table 6). As for the distance to target, the mean landing distance from the target position was equal to 0.308 meters in 1st mode compared to 0.356 meters in 2nd mode with a 14.2% improvement, as for the 3rd mode the mean distance is equal to 0.376 meters making 1st mode better than mode 3 with 18.36%. As well, the distance traveled was reduced from 2nd mode to 1st mode where it dropped from 16.3 meters to 15.7 meters with a 4.07 % improvement. Also, the distance traveled was reduced from 3rd mode to 1st mode where it dropped from 18. meters to 15.7 meters with a 13.83% improvement. Moreover, the standard deviations are smaller in 1st mode compared to the other 2 modes which suggests that more consistent flights were obtained by introducing the CNN based teleoperation system.

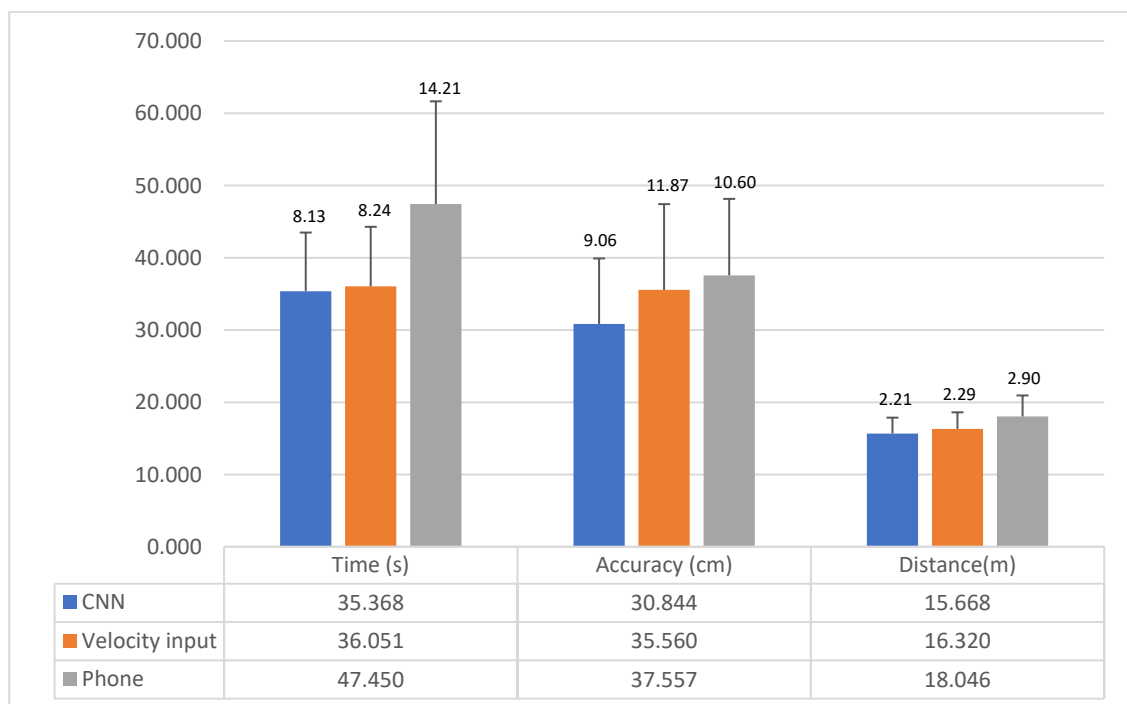


Figure 33: Objective results of the three proposed modes

Table 6: Comparison between the three modes

	Time (s)	Accuracy (cm)	Distance(m)
CNN vs Velocity Input	-1.91%	-14.20%	-4.07%
CNN vs Phone	-28.94%	-18.36%	-13.83%
Velocity Input vs Phone	-27.30%	-5.46%	-10.04%

The one-way Analysis of Variance (ANOVA) statistical test is used to analyze the relevance between the obtained means of the 3 modes. In order to accept or reject the null hypothesis, the resultant p-value is compared to the reference value $\alpha = 0.05$.

Table 7 displays the p-values related to each metric studied in the experiments. For all the comparisons, the p-values were less than 0.05 except for two cases; the flight duration for the CNN vs Velocity based inputs with a p-value of 0.6511 and the accuracy in landing for the Velocity based inputs vs Phone with a p-value of 0.3371. For the two cases where the p-value is high, we do not have sufficient statistical significance for the obtained results.

In our case, we are sure that using the 1st mode in teleoperation would improve the distance traveled and the accuracy in landing compared to the 2nd mode, but we are not sure for the future trials whether the flight duration is going to be reduced with 1.92%. The same case applies when comparing the 2nd mode to the 3rd one, there is no evidence against the null hypothesis when comparing the accuracy in landing for the 2 modes. But when comparing the 1st teleoperation mode to the 3rd one, it is clear that the 1st mode is going to beat the 3rd one in all the stated metrics with the displayed percentages of improvements.

Table 7 : P-values of the ANOVA test results

	Time (s)	Accuracy (cm)	Distance(m)
CNN vs Velocity Input	0.6511	0.0168	0.0488
CNN vs Phone	0.00003	0.0278	0.00002
Velocity Input vs Phone	0.0001	0.3371	0.0006

b. Subjective Results

Fig.17 shows the individual averaging of each one of the NASA TLX subscales. As shown, the load index for the 1st mode is always lower than the other two modes, also the load index is lower for the 2nd mode compared to the 3rd one. This proves that the use of velocity based inputs does not only enhance the performance of operators, it also reduces their average load index. But this improvement is increased even more by using the CNN teleoperation system.

Table 8 : Subjective results of the experiment on Parrot AR drone 2.0

	CNN	Velocity Input	Phone
Mental Demand	29 %	38 %	45 %
Physical Demand	23 %	32 %	41 %
Temporal Demand	27 %	36 %	44 %
Performance	26 %	33 %	39 %
Effort	30 %	35 %	48 %
Frustration	20 %	29 %	36 %
6 subscales equally weighted	25.7 %	34 %	42.2 %

The total scores of the 6 subscales equally weighted gave a score of 25.7% for the CNN-based teleoperation system, compared to 34% for the velocity-based teleoperation system, and 42.2% for the conventional teleoperation system. Which means that the load index was reduced by 8.3% by switching form the 3rd teleoperation mode to the 2nd one. And the load index was reduced by 16.5% when the user switched from the conventional teleoperation system to the proposed mode

ANOVA test was used to assess the obtained results. The p-value obtained for comparing the subjective results obtained from the first 2 teleoperation modes is equal to 0.015. By comparing 1st and 3rd modes together, the p-value obtained is equal to $2.8e-6$. The p-value relating the 2nd mode to the 3rd one is equal to 0.032. The results of the ANOVA test show a strong evidence against the null hypothesis (p-values $< \alpha = 0.05$).

CHAPTER VI

CONCLUSION AND FUTURE WORK

This thesis highlighted the background of UAV teleoperation system, the controllers used to teleoperate a quadrotor with or without haptic feedback. As well we showed how the input gains affected the operators during any type of teleoperation. Algorithms used in previous work to recognize intention in various systems were presented. We showed the importance of developing a personalized teleoperation system that works based on the user intention and uses personalized input mapping gains.

In this thesis, we explored the effect of adding another joystick to the teleoperation system using teleoperation similar to those in the literature. But doing so, did not enhance the performance of teleoperation; on the contrary, participants were frustrated.

The path of the thesis shifted to enhancing the mapping parameters of the UAVs mostly used conventional controller which is the RC. An adaptive gain tuning algorithm was developed to modify the mapping parameters of the RC based on the skill of every user. The testing was successful, and the algorithm was usable on any type of controller.

Then we were interested in developing an intention recognition tool that takes the joysticks commands as an input and transforms them to an output to control the UAV. This was done using CNN where the input was transformed to an image and studied in order to obtain the desired output. The performance of this method was compared to the velocity based commands and to the conventional teleoperation device. Our method gave better results both subjectively and objectively.

The proposed teleoperation algorithm was created and tested on the Phantom omni devices, as a future work it could be implemented to conventional teleoperating systems. Also, as a future work, the CNN based teleoperation system could be implemented using a different CNN architecture, and a different input preprocessing technique.

REFERENCES

- [1] F. Ferraguti, N. Preda, M. Bonfè and C. Secchi, "Bilateral teleoperation of a dual arms surgical robot with passive virtual fixtures generation," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.
- [2] J. Jakubiak, M. Drwięga and A. Kurnicki, "Development of a mobile platform for a remote medical teleoperation robot," 21st International Conference on Methods and Models in Automation and Robotics (MMAR), 2016.
- [3] S. Nahavandi, J. Mullins, M. Fielding, H. Abdi and Z. Najdovski, "Countering Improvised Explosive Devices through a Multi-Point Haptic Teleoperation System," IEEE International Symposium on Systems Engineering (ISSE), 2015.
- [4] M. Kamezaki, H. Iwata and S. Sugano, "An adaptive basic I/O gain tuning method based on leveling control input histogram for human-machine systems," in IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014.
- [5] M. Serna, L. G. Garcia-Valdovinos, T. Salgado-Jimenez and M. Bandala-Sanchez "Bilateral teleoperation of a commercial small-sized underwater vehicle for academic purposes," in OCEANS'15 MTS/IEEE Washington, 2015.
- [6] R. Martin, F. Antonio, R. G. Paolo, B. Heinrich H. and H. I. Son, "Experiments on Intercontinental Haptic Control of Multiple UAVs," in Intelligent Autonomous Systems 12, Jeju Island, Springer, 2012, pp. 227-238.
- [7] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," The International Journal of Robotics Research, vol. 31, no. 11, pp. 1279–1291, 2012.
- [8] A. Ruesch, A. Mersha, S. Stramigioli and R. Carloni, "Kinetic scrolling-based position mapping for haptic teleoperation of unmanned aerial vehicles," in IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, 2012.
- [9] R. Hala, H. Minh-Duc, H. T. and M. P., "Haptic-based bilateral teleoperation of underactuated Unmanned Aerial Vehicles," in 18th IFAC World Congress, Sophia Antipolis, France, 2011.
- [10] A. Kanso, I. Elhaji, E. Shamma and D. Asmar, "Enhanced teleoperation of UAVs with haptic feedback," in IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Busan, 2015.
- [11] A. S. Lomax, W. Corso, J. F. Ebro, "Employing Unmanned Aerial Vehicles (UAVs) as an Element of the Integrated Ocean Observing System," in Proceedings of MTS/IEEE OCEANS, 2005.
- [12] A. M. Samad, N. Kamarulzaman, M. Asyraf Hamdani, T. Aslamiah Mastor, K. A. Hashim, "The Potential of Unmanned Aerial Vehicle (UAV) for Civilian and Mapping Application," IEEE 3rd International Conference on System Engineering and Technology (ICSET), Shah Alam, 2013.

- [13] F. Fraundorfer, "Building and site reconstruction from small scale unmanned aerial vehicles (UAV's)," Urban Remote Sensing Event (JURSE), 2015 Joint, Lausanne, 2015.
- [14] S. Mghabghab, I. Elhadj and D. Asmar, "I. Adaptive Gain Tuning for Teleoperation of Quadrotors," in IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), 2016.
- [15] A. Kanso, I. Elhadj, E. Shammas and D. Asmar, "Enhanced teleoperation of UAVs with haptic feedback," in IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Busan, 2015.
- [16] X. Hou and R. Mahony, "An intuitive multimodal haptic interface for teleoperation of aerial robots," in IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014.
- [17] A. Franchi, C. Masone, H. Bulthoff and P. Giordano, "Bilateral teleoperation of multiple UAVs with decentralized bearing-only formation control," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, 2011.
- [18] J. Nagi, A. Giusti, L. Gambardella and G. Di Caro, "Human-swarm interaction using spatial gestures," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, IL, 2014.
- [19] T. Lam, H. Boschloo, M. Mulder and M. van Paassen, "Artificial Force Field for Haptic Feedback in UAV Teleoperation," Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 39, no. 6, pp. 1316 - 1330, 2009.
- [20] T. Lam, V. D'Amelio, M. Mulder and M. van Paassen, "UAV Tele-operation using Haptics with a Degraded Visual Interface," in IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06. (Volume:3), Taipei, 2006.
- [21] A. Brandt and M. Colton, "Haptic collision avoidance for a remotely operated quadrotor UAV in indoor environments," in IEEE International Conference on Systems Man and Cybernetics (SMC), Istanbul, 2010.
- [22] Y. Chenguang, C. Junshen, L. Zhijun, H. Wei and S. Chun-Yi, "Development of a physiological signals enhanced teleoperation strategy," in IEEE International Conference on Information and Automation, Lijiang, 2015.
- [23] M. Kamezaki, H. Iwata and S. Sugano, "An adaptive basic I/O gain tuning method based on leveling control input histogram for human-machine systems," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago, IL, 2014.
- [24] I. Farkhatdinov and J.-H. Ryu, "Hybrid position-position and position-speed command strategy for the bilateral teleoperation of a mobile robot," in International Conference on Control, Automation and Systems, ICCAS '07. , Seoul, 2007.
- [25] Q. Ang, B. Horan ; S. Nahavandi, "Multipoint Haptic Mediator Interface for Robotic Teleoperation," IEEE Systems Journal Vol. 9, no. 1 pp. 86 – 97, 2015
- [26] H. Igarashi, "Subliminal calibration for machine operation with prediction based filtering," in 12th IEEE International Workshop on Advanced Motion Control (AMC), Sarajevo, 2012.
- [27] K. Terashima, K. Watanabe, Y. Ueno and Y. Masui, "Auto-tuning control of power assist system based on the estimation of operator's skill level for forward and

- backward driving of omni-directional wheelchair," in IEEE International Conference on Intelligent Robots and Systems (IROS), Taipei, 2010.
- [28] K. Khokar, R. Alqasemi, S. Sarkar, K. Reed and R. Dubey, "A Novel Telerobotic Method for Human-in-the-Loop Assisted Grasping based on Intention Recognition*," in IEEE International Conference on Robotics and Automation (ICRA), 2014.
- [29] S. Nomura, T. Inoue, Y. Takahashi and T. Nakamura, "Learning Control based on Intention Recognition by Inverted Two-Wheeled Mobile Robot through Interactive Operation," in Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on Soft Computing and Intelligent Systems (SCIS), 2014.
- [30] Y. Rabhi, M. Mrabet, F. Fnaiech, P. Gorce, "A feedforward neural network wheelchair driving joystick," in IEEE International Electrical Engineering and Software Applications (ICEESA), Hammamet, 2013
- [31] C. Tzafestas, N. Mitsou, N. Georgakarakos, O. Diamanti, P. Maragos, S.-E. Fotinea, E. Efthimiou, "Gestural teleoperation of a mobile robot based on visual recognition of sign language static handshapes," in the 18th IEEE International Symposium on Robot and Human Interactive Communication. RO-MAN 2009.
- [32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no.11, pp. 2278-2324, November 1998.
- [33] K. Simonyan, and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in *Advances in Neural Information Processing Systems 27*, Pages 568-576, 2014.
- [34] E. Park, X. Han, T. L. Berg, and A. C. Berg, "Combining multiple sources of knowledge in deep CNNs for action recognition," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.
- [35] Parrot AR.Drone 2.0 Elite Edition," Parrot Store Official, 23-Mar-2017. [Online]. Available: <https://www.parrot.com/us/drones/parrot-ardrone-20-elite-edition>. [Accessed: 02-May-2017].
- [36] NASA Task Load Index (TLX) Instruction Manual, v. 1.0, Moffett Field. California: Human Performance Research Group, NASA Ames Research Center.