Controlling a drone via gestures

BSc (Hons) Computer science

Third year CS project Report at University of Manchester

Abdullah Alsaati

April - May 2016

Supervisor: Dr. Paul Nutter
Abstract

Commercial drones are becoming increasingly popular with new ones being introduced regularly as well as new methods and devices to control them. This report details an attempt to utilise gestures to control drones without using other means concurrently such as a joystick. During the length of the project several approaches were assessed in terms of suitability for the aims of the project. Consequently the most appropriate techniques and technology was chosen from the available options. This led to the attempt being mostly successful in meeting the initial aim although with some problems and unachieved objectives.

Acknowledgements

I would like to thank my supervisor Dr Paul Nutter for his guidance and valuable input throughout the duration of this project, my family for their continuous support and encouragement, and finally my friends for their encouragement and taking the time to try the program.
# Table of Contents

Chapter 1: Introduction ........................................................................................................4  
Section 1.1: Drones ...............................................................................................................4  
Section 1.2: Why ..................................................................................................................4  
Section 1.3: Aims and objectives ..........................................................................................5  
  Section 1.3.1: Aim ..............................................................................................................5  
  Section 1.3.2: Objectives ....................................................................................................5  
Section 1.4: Requirements ...................................................................................................6  
Section 1.5: Similar applications .........................................................................................7  
Section 1.6: Report Overview ..............................................................................................7  

Chapter 2: Development .....................................................................................................8  
Section 2.1: Methodology ...................................................................................................8  
Section 2.2: Hardware .........................................................................................................9  
  Section 2.2.1: Gestures capturing device .........................................................................10  
  Section 2.2.2: The drone ....................................................................................................12  
Section 2.3: Platform ..........................................................................................................14  
Section 2.4: The final product .............................................................................................15  
  Section 2.4.1: How it works ...............................................................................................15  

Chapter 3: Testing and Evaluation .......................................................................................17  
Section 3.1: Testing ..............................................................................................................17  
Section 3.2: Evaluation .......................................................................................................17  

Chapter 4: Reflection and Summary ..................................................................................20  
Section 4.1: Reflection .........................................................................................................20  
  Section 4.1.1: what went well ............................................................................................20  
  Section 4.1.2: Problems and what I would do differently ................................................20  
Section 4.2: Future project improvement ............................................................................21  
Section 4.3: Conclusions .....................................................................................................21
Chapter 1

Introduction

The project topic as the title indicates is a program to control a drone via gestures. That means the user will be able to control a consumer grade drone using arm and hand gestures and nothing else. The program will act as a medium between the gestures capturing device and the drone. It receives the gesture information from the sensing device and processes it and translates it into flying commands for the drone. This chapter contains a background of drones, details about why this project was chosen, what were the aims and objectives for it and finally similar applications from other sources.

Section 1.1: Drones

Drones are (UAV) Unmanned aerial vehicles. As the case with many technologies, they started as a military weapon then made their way to commercial and civil use. UAVs take many forms, commercial UAVs started out as (RC) radio controlled aircrafts. At the beginning it was a specialised hobby that needed an experienced user to fly a model plane effectively. However as technological advancements were made, RC aircrafts are now easy to use. Furthermore the introduction of various user friendly quadcopters that do not need expert manoeuvring of controls and throttle, led to people nowadays using quadcopters drones in everyday life. Some for entertainment others to capture video from a bird’s eye view using a drone fitted with a camera.

Section 1.2: Why

The reasons this project was chosen were: personal interest, gestures are more intuitive and more entertaining than conventional controls and finally because of the rarity of previous implementations. I have a personal interest in developing applications to control machines and by choosing a project that consists of this interest I hoped to achieve a higher quality product. Furthermore gestures in general if done properly are more intuitive than joysticks and other alternatives. By developing this application the aim was to make an intuitive and easy to use alternative that is different and more entertaining than conventional controls. Finally previous implementations of controlling a drone via gestures are not very common hence it’s beneficial to take advantage of a growing field.
Section 1.3: Aims and objectives
The project’s aims and objectives will be delineated in this section.

Section 1.3.1: Aim
Develop an application to control a drone that is an easy to use and intuitive alternative to conventional controlling methods.

Section 1.3.2: Objectives
1- Evaluate options
Evaluate the options available in terms of hardware and developing platforms and make decisions on what to use.

2- Gaining familiarity with SDKs
Upon making the decisions read the documentation of SDKs and come up with a draft for a program design.

3- Investigate sample drone programs and explore possibilities
To gain a better understanding of how the SDKs work and explore the possibility of repurposing some of the code to work in this project.

4- Identify possible gestures
Identify all possible gestures that can be detected by the gesture capturing device, then assign them to drone’s functions.

5- Develop a basic working example
The reason is to have something presentable as soon as possible.

6- Develop a portable version of the project
This version will be developed on a mobile platform (Android or iOS). The drone will most likely be used outdoors; likewise the user might want to move with the drone. In that use case the user will benefit greatly from mobility, making the project more useful in real world usage.

7- Explore ways to advance the project
Come up with advanced functionality beyond the scope of the original project to implement as a stretch goal
Section 1.4: Requirements

Functional Requirements:

1- **Hovering mode**
   The user is able to let the drone hover automatically without further commands to maintain altitude and position.

2- **Not piloting mode**
   The user can switch between piloting and not piloting the drone without having to land the drone or close the program.

3- **Emergency landing (fail-safe)**
   In the event of any emergency (e.g. loss of connection to the drone or gesture capturing device, or drone’s battery is low on power) there is a fail-safe in the form of the drone landing or another solution.

4- **Controlling mode**
   This mode will be the main functionality of the program. In this mode the user is able to pilot the drone in every way the drone is capable of using the most intuitive gestures to do so.

5- **Smart functions**
   As an extended goal, smart functions will be implemented. Smart functions such as using a gesture to command the drone to follow you automatically, another one is pointing at a place for the drone to go there automatically. These functions will be implemented if there is enough time for them.

Non-functional requirements:

1- **Intuitive and accurate drone movements**
   To ease the use of the program and lower the skill required to use it

2- **Easy gestures**
   Activating drone functions using easy gestures.

3- **Automatic prevention of bumping**
   This will also help with ease of use; furthermore it will help when flying the drone indoors or in cramped areas.

4- **Works on the most popular platform**
   The program will be usable by more potential users

5- **A mobile version**
   For portability and ease of use when moving, also the program will be usable by even more potential users.
Section 1.5: Similar applications

There are numerous programs to control a drone via joysticks or touch screen controls these are referred to in this report as conventional or regular controls. They are well established and supported as such there is no need to list them or analyse them in this report.

Officially supported programs to control a drone via gestures are very few; at the time of writing this report only one can be found that uses the same devices. However there are examples made by individuals, although when reading their instructions some of them warn that the program might be unstable in some cases, to be expected from unofficial individual work.

Section 1.6: Report Overview

This Report is divided into four chapters:

1- The introduction
   It is the chapter this section is in, it introduces the project, the motivation for it, the aims and objectives and similar applications.

2- Development
   This chapter consists of a description of the development process and all the choices that have been made regarding methodology, hardware and software used during development.

3- Evaluation
   In this chapter the program and choices made during development will be evaluated for success or failure.
Chapter 2

Development

During development not everything happened as planned. In this chapter there are details about the initial plan as well as the actual final result. Furthermore this part of the report will detail the decisions that have been made regarding which developing methodology, hardware and programming language to use. The first section will explain the development methodology that was used and a comparison with alternatives. The second section will describe the hardware that is used by the program then it will be compared with alternatives. Next the platform and programming language choice is described in the third section. Finally the fourth section describes what the final result is and how it works. To help illustrate the functional blocks of the project and how they are connected, figure 1 demonstrates those aspects of the project in the form of a diagram. The diagram will be gradually populated whenever a functional block is described.

Figure 1: Initial diagram of functional blocks

Section 2.1: Methodology

The initial plan for the project was to be divided into 4 parts using an iterative process those four parts were:

1- A controlling program for the drone on Windows
This iteration consisted of developing a program for windows desktops and laptops to control the drone via the gestures capturing device, provided they had the proper connectivity hardware to connect with the drone and the gestures capturing device. This iteration was to be the base for the more advanced functionality that was planned in the coming iterations.

2- **An obstacles avoidance system**

The drone will be able to automatically avoid crashing into obstacles using the system from this iteration. More specifically the drone will stop whenever it is about to crash into an obstacle. Furthermore it will not respond to incoming user commands if the result was for it to crash into that obstacle.

3- **An android app**

For added portability, the previous functionality was to be converted to an android app. This will benefit the user in the sense that the user won’t need to carry a laptop plus the drone and the gestures capturing device if he/she wants to use the program outdoors, which is the most likely use case for the program and the drone.

4- **Smart functionality**

Finally smart functionality was planned as an extended goal if there was enough time. There were two initial functions planned; following the user automatically without the user sending piloting commands and pointing at a location for the drone to go automatically to it and hover there.

This iterative process was chosen because when compared to the regular, develop everything at the same time approach, the iterative process has these benefits:

1- Complete focus will be on a single part of the functionality which will lead to a more refined program
2- Separate research, requirement gathering and testing for each iteration, this will help achieve the aims and objectives detailed previously.
3- Having something presentable at any giving time as opposed to not being able to demonstrate any working functionality using the regular approach.

**Section 2.2: Hardware**

Some of the hardware choices were already made, that is due to the university having only one kind of drone. However there was an ability to choose from several types of gestures capturing devices. The features and abilities of the hardware will be described below and alternatives will be evaluated.
Section 2.2.1: Gestures capturing device

There were two choices; the first was the leap motion and the second was the Myo armband. For a clearer and easier comparison their features are arranged in the table below:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Leap motion</th>
<th>Myo armband</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported platforms</td>
<td>C++, C#, Java, JavaScript, python</td>
<td>C++, Android, iOS</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Wired - USB</td>
<td>Wireless - Bluetooth</td>
</tr>
<tr>
<td>Sensing Technology</td>
<td>Cameras and infrared</td>
<td>Electrical muscle activity, accelerometer, gyroscope</td>
</tr>
<tr>
<td>Hands/arms placement</td>
<td>Have to place hands directly above leap motion</td>
<td>Device is wearable on arm</td>
</tr>
<tr>
<td>Ability to use outdoors</td>
<td>Limited – affected by direct light</td>
<td>Limited by Bluetooth range</td>
</tr>
</tbody>
</table>

The Myo armband was chosen for this project. It had many advantages over the leap motion, they are:

1- Supports the targeted platforms
The Myo armband has an SDK for Windows, Mac, Android and iOS while the leap motion does not support mobile platforms. A mobile platform is the final target for this project hence why the Myo was chosen.

2- More mobile

Thanks to the Myo wireless connectivity it allows for free movement. Although the Myo is bounded by Bluetooth range, the leap motion is a wired device which restricts mobility more so than the range of Bluetooth. Furthermore if the user of leap motion desires to move around while using it he/she has to hold the device with one hand and use it with the other hand. The Myo however only needs one arm to function leaving the second arm free for other uses.

3- Bigger functional area

The user wears the Myo on their arm therefore all of their arm’s range of motion is the functional area. In contrast the leap motion functional area is limited to the area above it; it also requires the hands not to be very close or very far.

4- Outdoor usage

The ability to use the program outdoors is crucial; in fact it is the most likely use case. The leap motion performance is affected by direct light more so by sunlight. Thus the Myo is a more suitable device for the purposes of this project.

Figure 2: partially populated diagram of functional blocks[5]
Section 2.2.2: The drone

The university had only one type of drone, the Parrot AR.Drone 2.0. Since there is no alternative to compare it against, its features will be listed in this section with an added focus on the ones that were useful in this project.

1- **It is a quadrotor UAV**
   It uses four rotors to move in the air, furthermore since it is a quadrotor it is capable of vertical lift unlike model planes and other commercial UAVs.

2- **Indoors and outdoors modes (figure 3)**
   The drone comes with external bumpers to protect it and nearby people when using it indoors. This feature made testing the program much safer and less stressful because the university regulations require the drone to be used only inside their facilities.

3- **Blockage/Bumping detection**
   The drone is capable of detecting if one of the rotors cannot rotate due to blockage or the drone has bumped into something, preventing further damage to the drone and rotors. Much like the previous feature it made testing safer.

4- **Battery level sensor**
   When the on-board battery is low on power, a warning is sent to the connected program. Moreover if the level gets lower still, the drone will automatically land to prevent a rough landing in the case of an empty battery.

5- **Automatic pitch, roll and yaw stabilisation**
   It has sensors to detect pitch, roll and yaw; it uses the data to automatically stabilise itself. This feature makes the drone easier to fly; it also makes this drone a very convenient fit for this project. The user would not have to learn how to balance these three, not to mention it will be difficult balancing those three effectively using gestures.
6- **Automatic hovering mode**
   When the drone is in the air and does not receive piloting commands or a command to land, it hovers automatically over a certain point. This is achieved using a camera that is pointed downwards to the ground. One of the objectives of this project is a hovering mode, therefore this feature saved the time that would have been spent developing this mode and figuring out how to stabilise the drone and process the measurements correctly to achieve automatic hovering. Thus to achieve this objective there is only a need to figure out how to toggle this mode on/off and when is the most appropriate time during program usage to do so.

7- **Assistance of basic manoeuvres**
   Where other UAVs require hours of training to perform these basic manoeuvres successfully, this particular drone uses its various sensors to simplify it to a push of a button. As such basic manoeuvres such as taking off, landing and trimming (calibrating) are easily performed from the very first use of the drone. Again this makes this drone a convenient fit for this project, since if these manoeuvres were not automated and simplified they would have been hard to translate to gestures for the user to perform.

8- **Wi-Fi connection**
   A Wi-Fi connection is used to communicate with the controlling device. Most devices that will use the project have only one Wi-Fi connection and one Bluetooth connection. Since the Myo uses Bluetooth connection, this left only the Wi-Fi connection to be used for the drone, which once more makes this drone fit for this project.

9- **Forward facing camera**
   It is used to stream to the controlling device (laptop/mobile phone) a video of what is in front of the drone. Certainly that video stream can be captured in a file but also that video stream can be used as a navigation aid in the case where the drone is far away from the user.

10- **Tags detection**
    Provided with the drone are tags that can be put on any object and the drone can detect it via the forward facing camera.
Section 2.3: Platform

The platform choice was decided by the non functional requirement of the project, that is for the program to work on the most popular platform, to be usable by more users. Also the choice was decided based on SDK support from both the drone and the Myo. In the case of the desktop program both SDKs support only Windows in common and Windows is the most popular used OS therefore the choice was straightforward. However when it came to the mobile version both SDKs support iOS and android. Nevertheless android is the more popular mobile OS and it is easier to develop for, since iOS app development requires XCode toolset which is designed only for Mac OS X.
Section 2.4: The final product

The first iteration of the project was a Windows desktop program that all future work will be built upon. Therefore naturally it was the first aspect of the project to be worked on. However there was a problem, the released Windows SDK for the drone does not work despite all the time and effort spent trying to make it work. Thus the third iteration (android app) became the first iteration and the previously planned Windows desktop program was cancelled. Furthermore while developing the android app there were other deadlines for other courses. For these reasons there was not enough time for the second iteration (obsticales avoiding system) or the final one (smart funcionality). Therefore the best course of action was to focus on developing the android app then test and refine it as much as possible before the deadline.

Section 2.4.1: How it works

The program runs a device listener to detect changes in the orientation of the Myo also to detect hand gestures. The orientation of the Myo is used for drone navigation, drone functions such as taking off and landing are assigned a hand gesture. A full list of detectable hand gestures is detailed below:

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fist</td>
<td>Landing/Taking off</td>
</tr>
<tr>
<td>Double finger tap</td>
<td>Toggling piloting/not piloting(hovering)</td>
</tr>
<tr>
<td>Wave in</td>
<td>Rotate drone in hands direction</td>
</tr>
<tr>
<td>Wave out</td>
<td>Rotate drone in hands direction</td>
</tr>
<tr>
<td>Fingers spread</td>
<td>Currently unassigned</td>
</tr>
</tbody>
</table>

As for Myo’s orientation data they are processed and converted to piloting commands as such:

<table>
<thead>
<tr>
<th>Angle</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yaw</td>
<td>Drone moves right or left</td>
</tr>
<tr>
<td>Roll</td>
<td>Drone moves forward or backward</td>
</tr>
</tbody>
</table>
The assignment of orientation angles to piloting commands were straightforward (Figure 7), except for moving forward or backward. Since the roll angle was the only one left, the gesture is less than ideal to be assigned to forward and backward movement. However it can be analogous to an accelerator in a motor vehicle.

When taking off or toggling piloting mode from hovering mode, the program saves the initial roll and yaw degrees of the Myo. Upon arm movement and Myo’s orientation change from these initial values, the program calculates the difference between current values and the initial ones. Then it converts the difference into a value of how fast the drone should move in each direction based on how many degrees is the difference of the corresponding angle. The pitch initial value is not saved, because it is assumed that when the user starts to use the program he/she will extend their hands intuitively to have a longer range of arm motion in every direction.

As a result of testing it was found that the drone is too fast to control indoors via gestures. Therefore the drone maximum speed was limited to a manageable speed while still being fast enough to entertain.

Finally to implement a fail-safe, if the connection to the Myo is lost or it was unpaired from the user’s arm, the program automatically sends a command to the drone to perform an emergency landing.
Chapter 3

Testing and Evaluation

Now that the final version of the program has been described, the next step is to detail the method of testing, then evaluate the final results based on the objectives and requirements that were planned at the start and listed in chapter 1.

Section 3.1: Testing

It is difficult to show testing results due to the nature of this project, therefore this section will focus on how the testing was done rather than showing results. Testing was done by making use of the varying perspectives of different individuals. By letting as many people as possible use the program, valuable feedback was produced. Among the results of this feedback the most notable one is the previously mentioned maximum speed limit. Although some of the users wanted a higher speed limit, others could not control the drone indoors effectively with a higher limit. Thus the limited speed is an average that suit all the users that used the program.

Section 3.2: Evaluation

To evaluate the outcome of this project first the results of the requirements will be assessed then objectives’ results will be assessed as well.

Firstly an assessment of functional requirements:

1- Hovering mode
2- Not piloting mode
   Were achieved by assigning them to the double tap gesture. At anytime while the user is piloting the drone he/she can double tap to toggle on the hovering mode(not piloting). While in hovering mode the drone will automatically hover at its current altitude and position using the downward facing cameras and its sensors. Furthermore the drone will not respond to the change in arm’s orientation or respond to hand gestures. Then the user can at anytime while the drone is hovering go into controlling mode by double tapping again.

3- Emergency landing (fail-safe)
   There are four cases where the drone will receive an emergency command to land. The first instance is built in the drone, when the drone bumps into something or the rotors cannot rotate, it automatically shuts down its rotors to prevent further damage. The second and third instances regard the connection to the Myo; if the
connection is lost or the Myo is unpaired the program automatically sends an emergency landing command to the drone.

4- **Controlling mode**
The requirement description was that the user will be able to control the drone in every way the drone is capable of. The drone is capable of moving forward, backward, upward, downward, right and left. All these types of movement are available by moving the arm that is wearing the Myo in the desired direction, except for forward and backward drone movement where the user has to rotate his/her arm.

5- **Smart functions**
Unfortunately this requirement was not implemented as a result of running out of time.

Next, an assessment of non-functional requirements:

1- **Intuitive and accurate drone movements**
   This was mostly accomplished, drone movement directly follow arm movement except for forward and backwards movements. However there is a problem with rotating the drone whenever the user does the “wave out” gesture. The Myo sometimes misinterprets the gesture as making a fist or another pose. This problem was not solved in time.

2- **Easy gestures**
   Due to the Myo’s ability to detect hand poses, drone functions are assigned an easy to perform gesture. A difficult alternative would have been compound gestures where the user has to perform more than one gesture or movement (e.g. drawing a circle etc.). This alternative would have been challenging to implement and less than ideal for the user.

3- **Automatic prevention of bumping**
   Unfortunately this requirement was not implemented as a result of running out of time. It would have been a helpful feature for some of the potential users as well as it would have allowed for unlocking the maximum speed.

4- **Works on the most popular platform**
5- **A mobile version**
   The program was developed on Android which is the most popular mobile OS.

Finally the results of the objectives are assessed:

1- **Evaluate options**
   In the course of this project there were a lot of decisions to be made, the most suitable hardware and software for the requirements were chosen. The choices were
made based on the advantages they offer to achieve the requirements, these advantages were described in chapter 2.

2- **Gaining familiarity with SDKs**
3- **Investigate sample programs and explore possibilities**
   Upon reading the documentation of the SDKs and reading accompanied sample programs, I gained a better understanding of how the SDKs work which helped greatly in easing development of this project.

4- **Identify possible gestures**
   All of the detectable hand gestures were used in the project and assigned to a drone function except for one, which was saved for advanced features later in the project. Also arm movement was detectable and made use of in drone piloting.

5- **Develop a basic working example**
   The example was meant to be developed as soon as possible on Windows. Unfortunately because the drone’s Windows SDK did not work despite attempting to make it work, this objective was not met. Furthermore this problem caused the first presentable working example on android to be developed later than planned.

6- **Develop a portable version of the project**
   The final developed version of the project is on Android thus this objective was achieved.

7- **Explore ways to advance the project**
   Although several advanced functionality was planned such as the obstacles avoidance system and smart functions, they were not implemented due to running out of time. However possible solutions for them were investigated, they will be discussed in chapter 4.

The aim for this project was to make an easy to use and intuitive program to control a drone via gestures, as an alternative to conventional controls. Based on the completed and uncompleted requirements, objectives and the feedback from testers, it can be argued that this project has mostly reached its goal.
Chapter 4

Reflection and Summary

After evaluating the results of this project comes personal reflection and a summary of all the aspects of this project. A Reflection not only on what went well during the project, but also on the problems that prevented a full completion of everything that was planned for this project at the beginning.

Section 4.1: Reflection

To reflect is to learn from one’s mistakes. Reflection is not only crucial but most of the time it is necessary. Reflecting let the person recognise their strengths and their weaknesses, so they are better equipped in the future to make decisions that fit their strengths and better equipped to make decisions that improve their weaknesses or avoid them.

Section 4.1.1: what went well

Choosing a project that interested me instead of one that seemed easy to do was a good idea. I enjoyed working on the project because it interested me, that led to me being more active in development than I usually be. Hence in the future, although it is not always possible, I will try to choose a project that interests me regardless of the perceived difficulty.

Another aspect that went well was researching and evaluating all the available options and then deciding which hardware/software was more suitable. As a result of choosing the most suitable Hardware and investigating its full capabilities before development, developing some of the requirements and objectives was straightforward as well as not taking a lot of time.

Section 4.1.2: Problems and what I would do differently

The first problem that I faced was the drone’s Windows SDK not working. Although it is not related to my performance that it did not work, I should have not spent as much time trying to make it work and instead developed on a different platform sooner. Consequently I might have had time for more refining or adding more functionality to the program. In the future a possible solution might be is a weekly reflection period to assess current progress and decide if the work should continue as it is at the time or change the approach taken.

Another problem was underestimating the workload of other courses, which led to being overly ambitious when planning functionality and not delivering them at
Section 4.2: Future project improvement

The project can be improved in many ways if there was enough time, below is a list of possible improvements:

1- **Outdoors and indoors mode**
   The user can toggle an outdoors mode that unlimits the maximum speed or toggles the indoors mode to limit the maximum speed to a manageable limit.

2- **The ability to set a custom speed limit**
   This can be a global one or can be implemented as the ability to set a custom speed limit for each mode mentioned previously. This will give more control to the user.

3- **The ability to reassign gestures to different drone functions**
   For increased customisability and for the user to have more control.

4- **Obsticales avoidance system**
   This feature was detailed previously, it will be redundant if it is described again. This feature can help the least skilled users, for example children, to pilot the drone safely. A possible solution is to install Infrared sensors in each direction and connect them to a device, which will be set up onboard the drone to send the sensors’ output data to the program for processing.

5- **Smart functionality**
   Theoratically the drone can be programmed to follow the user, since it is capable of detecting certain tags that come with it, The user can put the tags on him/her and gesture the drone to follow.
   As for pointing at a place for the drone to go there, the direction of the myo and the orientation relative to the drone can be calculated alongside data from the obsticales avoidance system. After that the appropriate commands are sent to the drone, the drone then flies to the location and hovers there.

Section 4.3: Conclusions

To summarise, first the project was introduced, then the development process was described, after that came an evaluation of the development’s results. It was mentioned in the introduction that drones are becoming increasingly popular in commercial use; for that reason it is a growing and exciting field to get into, therefore precedents are not very common. Thus choosing this project was somewhat risky, but it turned out to be an educational and entertaining experience. The objectives and requirements planned were a little ambitious, however that provided a valuable learning opportunity for future work. In addition, assessing the available hardware options to determine which is more suitable benefited the project and reduced the complexity of development. In conclusion despite the
unfulfilled objectives and requirements and all the problems faced during this project, it mostly met its aim, the development of an easy to use and intuitive program to pilot a drone via gestures as an alternative to programs that use conventional controls such as a joystick or touch controls.

References


