JANUARY 2018 M.Sc. in Aircraft and Aerospace Engineering MESUT OĞUZ

UNIVERSITY OF GAZIANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES

DESIGN AND CONSTRUCTION OF AN EXPERIMENTAL HUMAN GYROSCOPE

M.Sc. THESIS
IN
AIRCRAFT AND AEROSPACE ENGINEERING DEPARTMENT

MESUT OĞUZ

JANUARY 2018

Design and Construction of an Experimental Human Gyroscope

M.Sc. Thesis
in

Aircraft and Aerospace Engineering

University of Gaziantep

Supervisor
Asst. Prof. Dr.Kürşad GÖV

Co-Supervisor

Asst. Prof. Dr.İbrahim GÖV

By

Mesut OĞUZ

January 2018

REPUBLIC OF TURKEY
UNIVERSITY OF GAZIANTEP
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
AIRCRAFT AND AEROSPACE ENGINEERING

Name of the thesis: Design and Construction of an Experimental Human Gyroscope

Name of Student : Mesut OĞUZ

Exam Date : 23.01.2018

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Ahmet Necmettin YAZICI
Director

I certify that his thesis satisfies all the requirements as a thesis for the degree of Master of Science

Asst. Prof. Dr. Eyüp YETER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İbrahim GÖV
Co-Supervisor

Asst. Prof. Dr. Kürşad GÖV
Supervisor

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Examining Committee Members:	Signature
Prof. Dr. Cengiz DOĞAN	……………………
Asst. Prof. Dr. Eyüp YETER	……………………
Asst. Prof. Dr. Kürşad GÖV	……………………

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have cited and referenced all material and results that are not original to this work.

Mesut OĞUZ

ABSTRACT

DESIGN AND CONSTRUCTION OF AN EXPERIMENTAL HUMAN GYROSCOPE

OĞUZ, Mesut
M.Sc. in Aircraft and Aerospace Engineering Department
Supervisor: Asst. Prof. Dr.Kürşad GÖV
Co-Supervisor Asst. Prof. Dr.İbrahim GÖV
January 2018 ,60Page

Flight simulations were developed during the 20th Century to use in flight training, research and crash investigations in aviation fields. The importance of simulation has been realized since man first took to the air. Flight simulations have developed over the last century as a means of reducing the cost of aviation, structure losses, and time spent flying in expensive machines for the purposes of training. Flight simulators are mainly developed for commercial aircrafts by using Stewart platforms, which can perform limited manoeuvres. These simulators are not suitable for military and acrobatic aircrafts, which can perform very fast and complex manoeuvres in every axis.

In this thesis, a new type of flight simulator has been designed and a prototype manufactured. The Human Gyroscope type structure flight simulator can perform fast and very complex manoeuvres in every axis and 360 -degree rotational movements. For this purpose, a new control software has been developed that allows for random and user-defined simulations for the use of the Human Gyroscope type flight simulator. By using this software, simultaneous data is taken while the system is running, and the manoeuvres of the aircraft are visually simulated in the computer environment.

Keywords; Human gyroscope, Motor control, Flight simulator

ÖZET

DENEYSEL BİR İNSAN DENGE ÇARKI TASARIMI VE İMALATI

OĞUZ, Mesut
Yüksek Lisans Tezi, Uçak ve Uzay Mühendisliği
Tez Yöneticisi: Yrd. Doç. Dr. Kürşad GÖV
İkinci Tez yöneticisi: Yrd. Doç.Dr. İbrahim GÖV
Ocak 2018 , 60 sayfa

Uçuş simülasyonları 20. yüzyılda havacılık alanlarında uçuş eğitimi, araştırma ve çarpışma araştırmalarında kullanılmak üzere geliştirilmiştir. Simülasyonun önemi insanlar ilk havalandığında fark edilmiştir. Uçuş simülasyonları, havacılık maliyetini, yapı kayıplarını ve eğitim amaçlı pahalı makinelerde uçuş süresini azaltmak için geçen yüzyılda geliştirilmiştir.Uçuş simülatörleri, sınırlı manevraları gerçekleştirebilen Stewart platformlarını kullanarak ticari uçaklar için geliştirilmiştir. Bu simülatörler her eksende çok hızlı ve karmaşık manevralar yapabilen askeri ve akrobatik uçaklar için uygun değildir.

Bu tezde, yeni bir uçuş simülatörü tasarlanmış ve bir prototipi imal edilmiştir. İnsan Denge Çarkı (Human Gyroscope)yapısal tipine sahipuçuş simülatörü, her eksende ve 360 derecelik dönme hareketlerinde hızlı ve çok karmaşık manevraları gerçekleştirebilmektedir. Bu amaçla, İnsan Denge Çarkıyapısında uçuş simülatörünün kullanımı için rastgele ve kullanıcı tanımlı simülasyonlara olanak veren yeni bir kontrol yazılımı geliştirilmiştir. Bu yazılım kullanılarak, sistem çalışırken eşzamanlı veriler alınıp ve uçağın manevraları bilgisayar ortamında görsel olarak simüle edilebilmektedir.

Anahtar Kelimeler: İnsan denge çarkı, motor kontrolü, uçuş simülatörü,

To my family…

ACKNOWLEDGEMENTS

I would like to express my gratitude to my Supervisor Asst. Prof. Dr.Kürşad GÖV and Co-Supervisor Asst. Prof. Dr. İbrahim GÖV for his guidance, supporting, suggestions and preparation of this thesis.

I want to thank my family and my friend for supporting and helping me everything.

CONTENT

LIST OF FIGURES

Pages

LIST OF TABLES

Page

CHAPTER 1

INTRODUCTION

1.1 Introduction

Simulation is a technique in which a physical system for solving a problem can be represented mathematically by a computer program [1]. This technique of problem solving is used when it is not feasible to conduct specific tests using the actual physical system, such as the aircraft, due to time, cost, or safety. There are various kinds of simulators, such as; for cars, motorbikes, trucks, and aircraft. With these simulators, users can learn to use these vehicles. This thesis includes information about flight simulators.

1.2 Flight Simulator and Human Gyroscope

Since the humans came out into the sky, the importance of simulators has increased by its very nature, taking to the air in a machine is a costly venture. Aircraft cost money, fuel costs money, and lives are irreplaceable. Flight simulation has been developed over the last century as a means of reducing the cost of aviation, both in terms of the lives of pilots and passengers, as well as reducing hull losses and time spent airborne in expensive machines for the purposes of training. First simulators were designed to give novice pilots a feel for the air. While today a student can study various subjects such as Aerodynamics and Flight Mechanics to prepare him for flight, first flight instruction was not so scientific [2].

Human Gyroscope is defined as an aerotrim that allows the user to control whole body orientation in any direction. The rider makes it possible to control their speed and movements directly by simply shifting their body weight. Since there are no motors involved, all of the movement is initiated by the rider. Therefore, each individual can tune their workout to their own specific needs through dynamic interaction.

A human gyroscope consists of a frame for supporting the gyroscope, a ring assembly and a riding station to support a human rider on the ring assembly. The ring assembly includes an inner ring, an intermediate ring, and an outer ring concentrically disposed with each other. The outer ring is mounted on the frame for rotation around a first axis extending generally diametrically from the outer ring. The intermediate ring is mounted on the outer ring for rotation relative to the outer ring about a second axis extending generally diametrically from the intermediate ring. The inner ring is mounted on the intermediate ring for rotation relative to the intermediate ring about a third axis of rotation extending generally diametrically from the inner ring. Two of the axes of rotation of the rings are oriented obliquely with respect to the other of these axes of rotation [2]. A conventional Human Gyroscope is given in Figure1.1.

Figure1.1 Human gyroscope[3]

Today, the use of simulators in the gaming industry is widespread. In arcades around the world, it is seen moving flight, car, and virtual reality simulators. These types of devices are being used purely for entertainment applications. Therefore, there are other kinds of moving simulators that are being used to study reactions of the human body in preparation for space travel or flight missions, when the human body is subjected to high G-forces. The advantage of using moving simulators is clear, as simulations take place in a safe, controlled environment to test the human body without risking the person's health or, as stated in the beginning, they can simulate driving or flying, just for fun[4].

In this thesis, the motor - driven human gyroscope is examined. The driven systems were studied and compered to each other. The established gimbal mounting of the frame elements permit a person occupying the innermost frame element to assume any angular position in space solely with the aid of the motors. It can be used for a variety of purposes, ranging from aiding pilots in overcoming airsickness and disorientation to therapeutic treatment for neurological disorders. For this purpose, a prototype human gyroscope device has been designed and manufactured. The prototype Human Gyroscopeis given in Figure 1.2.

Figure 1.2 Prototype ofHuman Gyroscope

1.3 Research Objectives and Task

The main goals of this thesis are todevelop new type of flight simulator that name isHuman Gyroscope. This simulator includes new design, new control unit and new software.

The research tasks can be shown as follows;

1. Development of new frame design
2. Make stable control unit
3. Development of a driven software
4. Visualize the flight path

1.4 Purpose of Thesis

In this thesis,the explanation of the structure of the flight simulator, the type of motor, controlling type and software was given. These topics are examined in detail in different sections.

The first section describes the types of motor to be used and the characteristics of the specified motor.After the comparison, the appropriate motor and driver must be selected. This choice was noted that attention should be paid to what is done. The suitable driver for this selected motor has been identified. Then the selection of mechanical parts in the flight simulator and information about the electronic parts were given.Finally, information about the developed software was given.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

The literature includes many types of flight simulators. Researchers have been working on simulators for a very long time. These simulators are not just aircraft simulators. There are also simulators for cars, motorbikes, and games. Car or motorbike simulators should simulate driving on asphalt or a footpath and should be designed based on car and motor equipment. This is similar on airplanes. Each simulator has its own unique design. However, aircraft simulators have the most complex structure, because aircraft can make many different moves.

In this section, studies on simulators in the literature up to now are examined. The aim and results of the studies are compared.

Erçin, [5], worked on the development of a nonlinear simulation model of a helicopter and the design of an automatic flight control system. A nonlinear dynamic model for a full-size helicopter was developed using a MATLAB / SIMULINK environment. In order to design an automatic flight control system, a classical internal stability loop and an external flight sequence mode approach are applied. Both the conventional speed feedback and the intermittent system state feedback control approaches are used for the internal stability loop. The applied outer loop modes are hover state and heading, pitch, roll, elevation acquisition and hold mode, and heading, pitch, roll, elevation acquisition and hold modes for air speed retention for forward flight status.

According to Lancaster [6], empirical work on general aviation pilot performance is lacking, especially in view of the anticipated future requirements. Two research studies have been conducted to evaluate human performance using new technologies. In the first study, ten participants completed the Modified Rhyme Test to compare the intelligibility of the two speech texts presented as 85 dB (A) aircraft cockpit motor noises. In a second study, pilot data was evaluated in terms of link performance using a flight simulator equipped with a mixed method of simulating flight conditions 16 data link for one of the two visual flight rules.

Root, [7] studied the development of a fixed-base flight simulator with avionics or avionics loop testing equipment. The system consists of a pilot-controlled 6-degree-of-freedom aircraft model, a signalling display, and a hardware interface for an arinc 429 avionic databus. This databus can be used for testing avionics. This system has been combined in a modular form that allows easy reconfiguration for use with various aircraft models and testing of avionics using databases.

According to Kaith [8], The Flight Simulator serves as a tool for serving and designing and evaluating structures and controls. The primary purpose is to function as a programmed Flight Simulator. For this purpose, the Flight Simulator's command repeatability and response accuracy should be as good as the analogue computer running it. The development of flight control systems requires knowledge of trunk dynamics. The purpose is to design a flight simulator that follows the roll, step, and yaw movement of a simulated g body. The flight simulator follows the solution of differential equations with small error. Then the active elements of the system can react to their natural surroundings.

Abbasi, [9], has worked to develop flight simulations on flexible aircraft focusing on the comparison of two flexible symmetry models. Simulation models have been developed and tested using a range of control inputs, both longitudinal and lateral dimensions. Based on the results of the study, a number of suggestions were made for further testing and research.

Stueck, [10], investigated the feasibility of simulating the Landing Signal Officer pilot interaction during the approach to a landing on an aircraft carrier. A simulator has been created that replicates the Landing Signal Officer's working environment through visual images generated by the computer. The test results show the basic feasibility of simulating the Landing Signal Officer pilot interaction and its application as a research tool in Landing Signal Officer models, wave-off techniques, and landing techniques.

Sherman, [11], the hang glider is probably the closest thing a person can experience to bird-like flight. Unfortunately, it is difficult for a hang glider to be taught through demonstrations or verbal instructions to acquire the delicate skills required to fly. As a result, many people's first flights are strange and sometimes dangerous. This need has led to the development of the Gooney Bird Trainer, a glider simulator that imitates the pitch, roll, and yaw characteristics of a planer in response to a student's control movements. The simulator will also allow students to practice the launch and descent sequences.

Albayrak, [12], studied the modelling and real-time control of a motion simulator for the dynamic testing of a biaxial gyro-balanced head mirror used in modern tanks. For this purpose, a six-stage Stewart Platform is used. The mathematical models of the Stewart Platform were built using MATLAB and ADAMS. The control unit setting is accomplished using mathematical models developed in MATLAB®. These parameters are applied in a real - time control system and fine adjustments are made. The accuracy of the motion simulator is tested by mounting an Inertial Measurement Unit on the Stewart Platform. More control system strategies are handled through simulation.

Cristofaro, [13], has shown the process of introducing high accuracy methods into the aircraft design process. The simulation tools evaluate the static and dynamic stability and manoeuvrability of aircraft based on a generally pre-calculated aerodynamic model. Starting with the first steps in aircraft design, the adoption of highly accurate, physically - based aerodynamic models will reduce the uncertainty of existing design procedures and prevent costly aircraft from being designed inefficiently. In Marco Cristofaro, steps are taken towards an autonomous high-accuracy dynamic analysis.

Çoban, [14],analogue devices are replaced with more capable digital devices by developing technology. The mechanical indicators related to aircraft technology are replaced by computer-aided imaging units. Mechanical indicators need to be independent from each other, so they need more space and a heavier aircraft. Computer - aided Main Flight Display Units solve the problem. The Windows Presentation Framework, which is included with Microsoft.NET 3.5 technologies, is used to provide high graphics performance. Communication is carried out with User Datagram Protocolpackets. This unit can be easily integrated with simulation and real systems. For the final conditions, an average CPU and an average GPU can provide the required performance. The Synthetic Data Generator, which can provide data for both this unit and external simulator systems, is implemented. The Synthetic Data Generator publishesUser Datagram Protocolpackets for basic indications. The system is then integrated into the network.

Bouthillier, [1], a low-cost flight simulator was designed and implemented using a low-cost, 3-degree freedom movement platform with components located on the market. The purpose of this study was to explore the advantages and drawbacks of such a system. This document describes the design, implementation and analysis of this system. At the heart of this work was the classic washout filter algorithm, a widely - adopted motion detection algorithm used to generate platform motion. The inertial measurement unit was used to examine and optimize the classical algorithm. Finally, the system was evaluated by collecting pilot grades with pilot validation exercise. The results showed that the majority of the pilots were satisfied by the system and adding motion to the flight simulation greatly improved it. The pilots also developed and identified areas for potential applications for the flight simulator. Empirical analysis of the system using an inertial measurement unit and grades obtained with a pilot validation exercise show that the low cost 3-degree flight simulator has potential for flight training.

Kesserwan, [15], simulation is the technique by which a physical system can be represented mathematically by a computer program for the solution of a problem. A mathematical model is developed for the physical system using knowledge of the physical laws describing the problem. This model is programmed on the computer to generate the problem's solution.

Bauer, [16], investigates the Educational Effectiveness of a low-cost, PC-based training system when compared to two modes (movement and inactivity) of a large-screen cabin training system for various aviation flight missions. This research considers the use of such training systems in the education process for a highly complex and dynamic task situation: a search and rescue mission. Maria C. Bauer's research demonstrates that a "low cost" simulation provides the experience necessary to learn highly complex missions related to search and rescue, as well as a platform of action in flight training.

Flight simulators are a common and useful tool for verification as well as control system design and are widely used in the aviation industry to evaluate system performance. The Illinois Simulator Lab uses a modular architecture that includes a simulation environment, a Simulink dynamics model, X-Plane visualization, a Frasca 142 cockpit, a customizable digital cockpit dashboard, and communicates between the elements running on the local area network. Ackerman, [17], shows that the design of a control was quickly evaluated, and the Simulink and flight simulator test can be tested to obtain pilot performance back. Thus, the creation of the Simulink library provides effective switching of simulation modes and control configurations.

The purpose of the Martinez [18], project is to carry out a security analysis for the future of flight simulators. One of the most crucial aspects of these devices is safety, because they have direct contact with the individual. By analysing the university simulator, the project not only provides background information on the topic, but also allows comparison of results from similar devices in the future.

Lawn, [19], created the flight simulator to give an idea of aircraft characteristics. A program was developed to estimate the stability of a plane based on the USAF DATCOM method. This program includes solutions that include all necessary graphical data. Then the project describes the development of dynamic models of flight dynamics and kinematics including ground and hybrid models. In his work, Peter Lawn describes the implementation of the models as a work simulation program. Then, he explains in detail the operation of the flight and navigational equipment typically found in general aviation aircraft simulation.

Shi, [20],was interested in the movement and control of a flight simulator with an electro-hydraulic servo valve. The basis of electro-hydraulic servo and hydraulic control systems are discussed. An electro-hydraulic servo-valve was mathematically modelled and simulated using a linear hydraulic double-ended cylinder, which is applied by Simulink. Finally, motion and control loading systems of a full flight simulator are examined. Motion simulation algorithms and their application are discussed.

Plumpton, [21], presented the development of a simulator prototype, built upon a Diamond DA20-A1 fuselage and developed through the effective use of original aircraft components and commercial off-the-shelf components. This development focused on the integration of flight controls and electrical components and their interface and virtual environment. Following the completion of the prototype, the evaluation of the simulator found that, despite some shortcomings, a low-cost, type-specific, high-performance aircraft simulator for flight training was developed.

A Project byZeitoun, [22], was related to flight simulation software and hardware cockpit instruments agreement on a distributed system. The project used the TRC472 Flight Cockpit with PresacisFlightSIM to fully mimic a Cessna 172 Skyhawk aircraft.The final product is to be a full -fledged flight experience with the complete environment and physics of the Cessna 172.

Dibello, [23], easy-to-use interactive simulator is developed to help students and others learn about the impact of time delays in complex systems.Users on the computer screen, at different times between different patterns and control actions on the screen with the effect they are trying to follow the target location for a destination with delays. The results of these experiments were used to determine the learning effect of the simulator. As expected, performance and consistency improved as the number of trials increased for each user.

Brownridge, [24], was described the methods of motion capture by an application for real-time recording of extreme human motion. A wireless gyroscopic sensor - based system was used to record and evaluate the misalignments in ankle position of ballet dancers in a performance environment. The process of providing a visually acceptable representation of an overall performance, in addition to the numerical evaluation of a specific joint lead, is an important contribution to knowledge.

The goal of Homan, [25], was to compare pre-training through the use of a cheap multimedia computer program with an ELITE ™, FAA - approved flight training device. DME ARC was chosen as a standard instrument flight manoeuvre. In an

FAA - approved simulator, a computer - scoring program was used to evaluate the psychomotor skills of the pilots.

A flight simulator for the ornithopter has been developed to evaluate design changes, try out take-off and flight strategies, and train the pilot. It is written in C ++ using an object-oriented paradigm. The simulator takes the keyboard input and the output is displayed as an ornithopterine line drawing. It solves motion equations with a time-based integration method. Many of the forces on the ornithopter are calculated analytically at all times. The simulator gives satisfactory results even though it can be improved by increasing the accuracy of the input data. Robinson, [26],usedit to assess the optimal design for an additional wing, examine possible landing gear changes, and analyse potential take-off strategies.

Sagoo, [27], presents the pilot in the loop tests carried out in a Six-Degree of Freedom motion flight simulator to evaluate failure detection, isolation, and identification schemes for an advanced F-15 aircraft. Sagoo, is an excellent example of how to effectively use auditory, visual, and motion hints to create a high level of simulation complexity in the design of control algorithms.

Rekleitis, [28], a new 6 degree-of-freedom parallel mechanism is considered as a flight simulator base. The mechanism consists of three fixed lengths supported by a three-wheeled mobile robot. It supports a platform that acts as a tip effect of a three-legged mechanism. The mechanism is considered as a base for a flight simulator using typical aircraft manoeuvres. The mechanism shows potential as a motion basis for a driving simulator.

Lee, [29], conducted a full factorial experiment with nine subjects in the UTIAS flight research simulator to investigate the effect of four factors on the phase difference between motion and visual cues. Movement gain, amplitude, frequency, and visual scene complexity were four factors. Two levels were tested for each factor and sixteen sessions were done in each subject. A Kennedy SSQ was applied to the topics after each session. The movement of the simulator was limited to the sinusoidal pitch. The results obtained in this study indicate an average phase difference threshold of 56.3 ° and a standard deviation of 5.67.

Li and Salcudean, [30], explain modelling, simulation, and control of the inverted, tumbler-mounted Stewart platform designed as a single-action simulator. The dynamic equations of the Stewart platform are derived using the virtual working principle. Leg dynamics are shown in negligible simulations. A model of the electrohydraulic actuator is derived and then verified using the experimental data.A link- space pressure-feedback controller is proposed for high performance with good stability robustness.

Thöndel, [4], looks at the design and optimum control of an electromechanical linear actuator to be used in six-degree-of-freedom motion platform design for simulation technology. Then, while reacting to the final call in the simulation industry, the hydraulic cylinders with electromechanical actuators are replaced, but kinematic and dynamic parameters are not affected. In Thöndel, the comparison of the two system types and the optimal control design for electromechanical actuators are explained.

In Runevad and Kjellin, [31], the simulator will be implemented as part of virtual reality systems, computer games, flight simulators, and training platforms. The motors on this platform will be controlled by the user's movements of the joystick, which will provide continuous and unlimited rotation. A very simple way of understanding the potential of this platform can be seen in a practical application such as flight simulator, where the flight requires continuous and unrestricted yaw, pitch, and roll motion types. A motorized gyroscope is a new and practical solution for such applications.Scenarios can besimulated.

2.2 Conclusions on Literature Survey

Many studies have been done on flight simulators. For this purpose the type flight simulators have been studied. The manoeuvre capability ofSteaward platform is limited compared to the HumanGyroscope movement. The Human Gyroscopeis mostly used in entertainment industry. However, from the literature survey, the use of HumanGyroscope as flight simulator has not been seen. In this study, the Human Gyroscope structure was modified to a flight simulator.

CHAPTER 3

FLIGHT SIMULATOR

3.1 Introduction

Flight simulator is a device that artificially re-creates aircraft flight and the environment in which it flies, for pilot training, design, or other purposes. It includes replicating the equations that govern how aircraft fly, how they react to applications of flight controls, the effects of other aircraft systems, and how the aircraft reacts to external factors such as air density, turbulence, wind shear, cloud, precipitation, etc. Flight simulation is used for a variety of reasons, including flight training (mainly of pilots), the design and development of the aircraft itself, and research into aircraft characteristics and control handling qualities [32].

The control of Human Gyroscopes can be classified in three main categories. The first is the user - controlled human gyroscope; movement is actuated by human body orientation in any direction. It allows the person riding to directly control the speed and movement by simply shifting their body weight. Since there are no motors involved, all of the movement is initiated by the rider. The next is the joystick-controlled Human Gyroscope. All movements are actuated by the joystick capability and no motion type is defined. The last is a fully - controlled type control for a Human Gyroscope. With this type, velocity, acceleration and path can be defined and actuated by the user. The important component of the human gyroscope is the motors that are used for actuating the motions.

The structure must be rigid and stable. For this reason, the main frame of the system is made of steel St-37 and moving part made from 6061 aluminium. All parts are manufactured with laser cutting machine and CNC lathe machine.

3.2 Motion Control Devices

- Stepper Motor
- Servo Motor
- DC Motor
- AC Motor

3.2.1 Stepper Motor

The stepper motor (Figure 3.1) is an electric motor, which converts electronic pulses to mechanical rotation. The speed of the stepper motor is determined by the time delay between steps. Generally, a Stepper motor has 0.9 degree or 1.8 degree and 200 steps or 400 steps per revolution. These steps define the sensitivity of the stepper motor.

Figure 3.1 Stepper Motor

Advantages of Stepper Motors

- Low cost
- Stepper motors work in an open loop (no feedback system required)
- Excellent torque at low speeds
- Low maintenance cost (brushless)
- Precise positioning control is good

Disadvantages of Stepper Motors

- Torque decrease when speed increase
- Noisy at moderate to high speeds

- Stepper motors can lose position running without a control loop
- Low efficiency

3.2.2 Servo Motor

The servo motor (Figure 3.2) is basically the same as DC and AC motors but with a servo configuration. The motors have very precise position feedback. The servo motor is controlled by the driver, so the driver controls the mechanical position, velocity, and acceleration. The servo motor contains the motor driver and control board.

Figure 3.2 Servo Motor

Advantages of Servo Motors

- High torque
- High speeds
- Work well for velocity control
- All size is available
- Its work quiet

Disadvantages of Servo Motors

- More expensive than stepper motors
- Cannot work open loop - feedback is required
- Require tuning of control loop parameters

3.2.3 DC Motor

The DCmotor (Figure 3.3) is an electric motor that runs on direct current (DC) electricity.The very basic construction of a DC motor contains a current - carrying armature, which is connected to the supply end through commutator segments and brushes and is placed within the north / south poles of a permanent magnet or an electro-magnet as shown in the diagram below.

Figure 3.3DC motor

Advantages of DC motors:

- Speed of DC motor controlled easily
- Starting torque is high
- Quick starting, stopping, reversing and acceleration

Disadvantages of DC motors:

- High initial cost
- Increased operation and maintenance cost due to presence of commutator and brush gear
- Cannot operate in explosive and hazard conditions due to sparking occur at brush (risk in commutation failure)

3.2.4 AC Motor

An Asynchronous ACmotor (Figure 3.4) is an electric motor driven by alternating current.It commonly consists of two basic parts: an outside stationary stator with coils supplied with alternating current to produce a rotating magnetic field, and an inside rotor attached to the output shaft that is given torque by the rotating field.

Figure 3.4AC Motor

Advantages of AC motors

- Low cost
- Simple design
- Generally high efficiency

Disadvantages of AC motors

- Speed control in induction motors is difficult
- Poor starting torque
- Efficiency drops at low loads.

3.3 Motor Selectionfor HumanGyroscope

To control the human gyroscope, the motor selection stage is prepared by using the design table. The design table is created according to the major and minor requirements of the control system. The requirements are selected according to the control system's needs. The requirements are graded from 1 to 10 to define the necessities of the system.

3.3.1 Major Requirement

Table 3.1 Major Requirement

Properties	DC Motor	AC Motor	Stepper Motor	Servo Motor
Major Requirement(%65)				
Torque	7	2	8	7
Weight	3	3	5	8
Size	4	2	7	8
Controlling	8	1	8	10
Precision	8	1	7	10
Repeatability	7	1	7	10
Sub Total	37	10	42	53

3.3.2 Minor Requirement

Table3.2 Minor Requirement

Properties	DC Motor	AC Motor	Stepper Motor	Servo Motor
Minor Requirement (%35)				
Tuning	3	2	9	4
Maintenance	8	9	9	10
Price	4	2	7	8
Mounting	8	1	8	10
Sub Total	15	23	34	27
Total	29	15	39	44

Looking at the different characteristics and rating given; the most suitable type would be a servo motor or stepper motor according to the working conditions. And the fact is that the best motor for controlling the human gyroscope is a servo motor, because the grade of the major requirements is very important. The design table shows that stepper motors and DC motors may be used for any prototype in Human Gyroscope applications. But AC motors are not suitable for this purpose because of the worst control characteristic.So, the stepper motor was chosen for its easy to control and low cost properties in this prototype Human Gyroscope flight simulator.

3.4 Construction

The humangyroscope has unique design (Figure 3.5). The design has 3 concentric rings. Each of these rings represents an axis, and these three axes can act independently from one another. Almost all moving parts are made of aluminium because the system needs to be light and durable. The system's motion components are specially designed. There are two special ball bearings in this part. Stepper motors connect these special parts, and these parts transmit power to the rings.

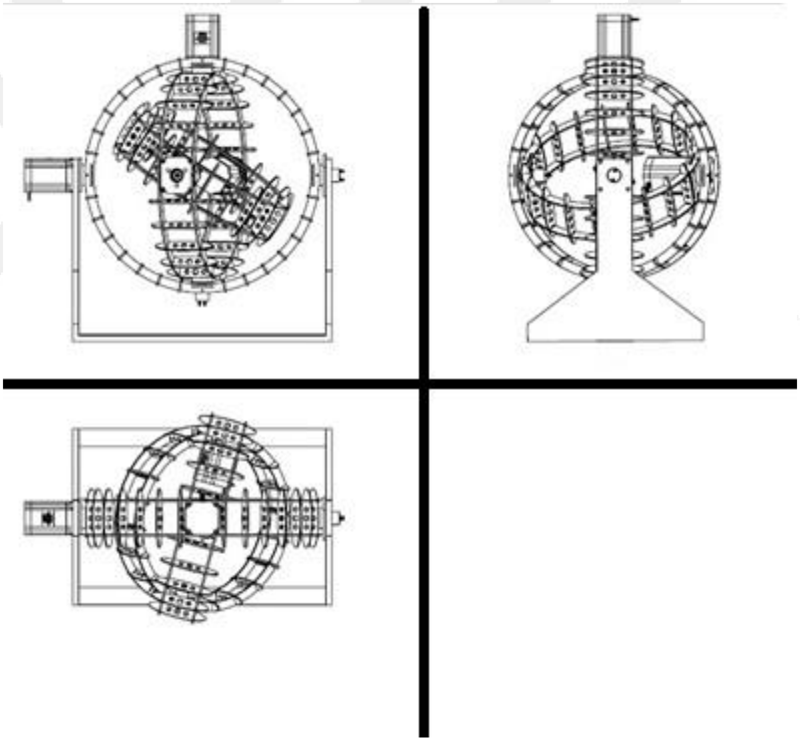

Figure 3.53D View of HumanGyroscope

3.5 Materials and Properties

3.5.1 Aluminium

6061 aluminium is highly versatile and can be used for almost any structural component. Commercial and personal use vehicles utilize 6061 for truck frames and running boards, infrastructural uses include mass transit and subway platforms, steps, flooring, walkways and cover plates, and consumer products such as bicycle frames and components, SCUBA tanks, fishing reels and small utility boats all benefit from the increased weldable nature and its ability to undergo hot forging.

3.5.2 Steel

ST-37 Steel is non-alloy steel which hasyield strengthof235 MPa, tensile strength of 360-510 MPa.St-37 have very good welding properties, good mechanical properties and easy to machine properties. This St-37 steel is used as astructureofHuman Gyroscope main frame and some connection parts.

3.5.3 Construction Parts and Properties

3.5.3.1 Support

The structure includes lots of components. One of these parts is support (Figure 3.6). Support connect the rings to each other. It has some holes which are used forthe cable channels the cables pass through these holes to the stepper motors. This part is made from aluminium and manufactured by laser cutting machine.

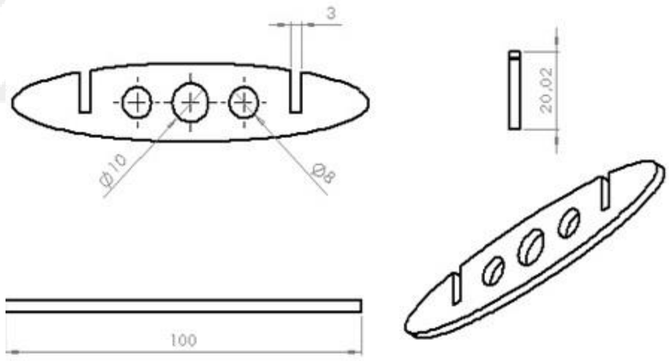

Figure 3.6 Support

3.5.3.2 Slipring Connector

The slipring connector (Figure 3.7) is manufactured by steel and consists of two parts. The flat part is manufactured by sheet metal by using laser cutting machine and the second cylindrical part is manufactured in CNC lathe. Then these two parts are welded to obtain connector.

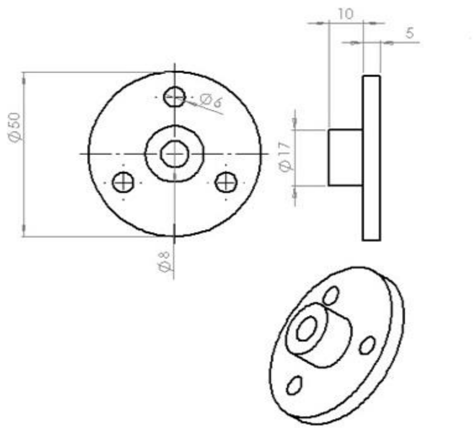

Figure 3.7 Slipring connector

3.5.3.3 Motor Connector

It was designed for the Nema 17 stepper motor. The motor connector (Figure 3.8) is similar to the slipring connector. It’s made by the same process, and the only difference is its shape and connection type.

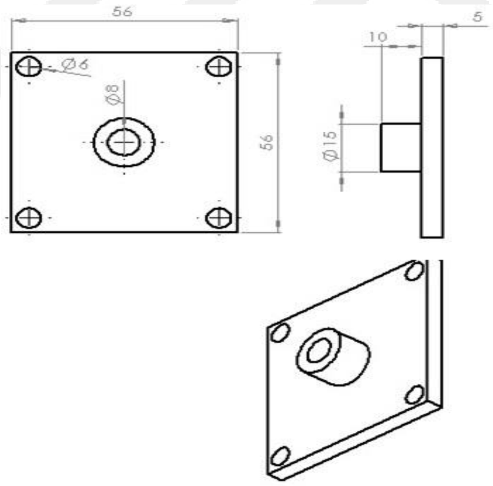

Figure 3.8 Motor Connector

3.5.3.4 Bearing House

The bearing house (Figure 3.9) is used as coupling with the motor and the rings. It’s made from the same process as the connectors. The bearing house is used for mounting the two-ball bearing and it is coupled with the connectors to obtain coining of the motor and rings.

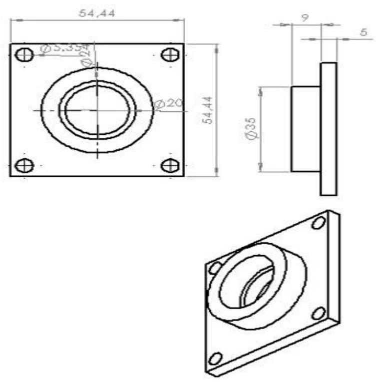

Figure 3.9 Bearing House

3.5.3.5 Rings

The rings (Figure 3.10) are manufactured byaluminium alloy 6061. The two rings are connected with supports to form a circle. The structure includes six rings, and these gimbalsare divided into three groups such as outer gimbal, middle gimbal and inner gimbal.

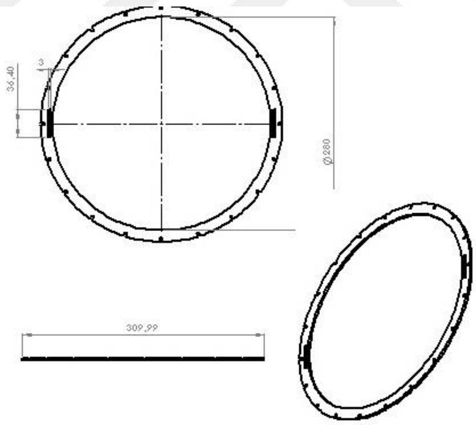

Figure 3.10 Rings

3.6 Assembly of Human gyroscope

The Human Gyroscope type flight simulator have been manufactured by many parts. These parts are assembled to each other to obtain simulator. The components and assembling procedure are explained in the following sessions.

3.6.1 Slip Ring Coupling

In the assembly process, the first part is the slipring coupling (Figure 3.11). This coupling design consists of three parts. The first part is connected to the slip rings;

the other one is connected to the gimbals. Then these two parts are connected by using ball bearings.

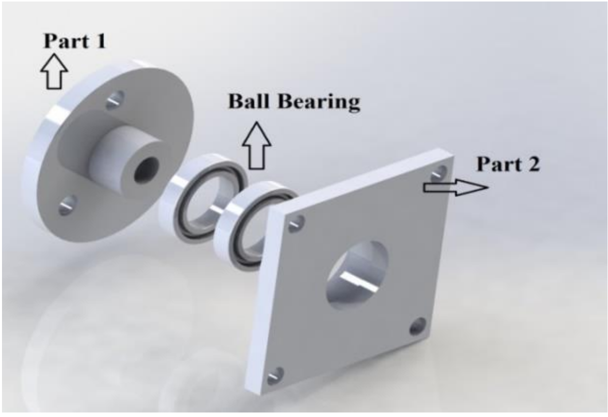

Figure 3.11 Slipring Coupling

3.6.2 Motor Coupling

After assembly of the slipring coupling (Figure 3.12), the motor coupling process is operated. It is similar as the other slip ring coupling process, but Part 1 has a rectangular shape, Part 1 is designed to connect for the Nema17 stepper motor.

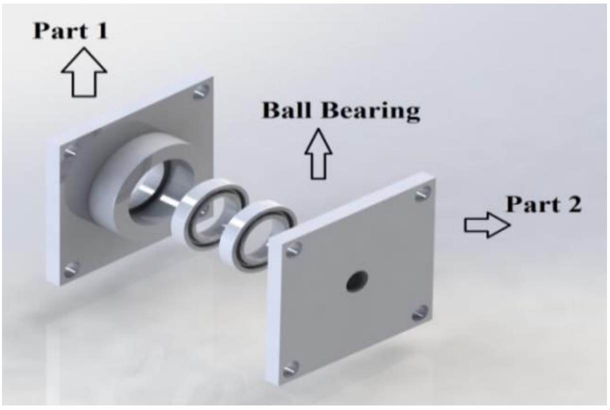

Figure 3.12 Motor Coupling

3.6.3 Main Chassis

Main chassis is designed and manufactured for base structure of the Human Gyroscope. All parts of the main chassis (Figure 3.13) are produced by laser cutting machine and then the parts are welded to each other to obtain the main chassis. The main chassis is manufactured by 10 mm steel sheet metal by using the laser cutting machine because it carries the full weight of the system and should be strong and rigid. The main stepper motor and slip ring are mounted on the main chassis.

Figure 3.13 Main Chassis

3.6.4 Inner Gimbal

The inner gimbal (Figure 3.14)includes a lot of parts. Supports, motor coupling, slip ring coupling, slip ring, stepper motor, and rings. First, 2 rings are connected by supports. Then the stepper motor and slip ring are connected to the couplings. These couplingsare connected to the inner gimbal by bolts.

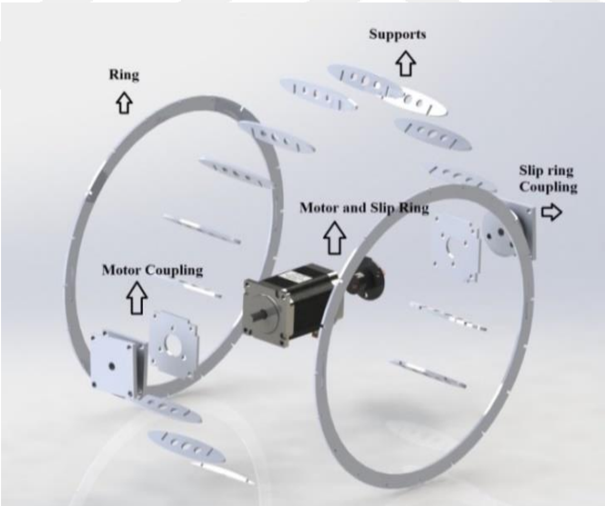

Figure 3.14 Inner Gimbal

3.6.5 Middle Gimbal

This gimbal (Figure 3.15)is the simplest one because it doesn’t include any stepper motor or slip ring. The stepper motor and slip ring are mounted on the outer ring, so it is only used for assembling the inner and outer gimbals to each other.

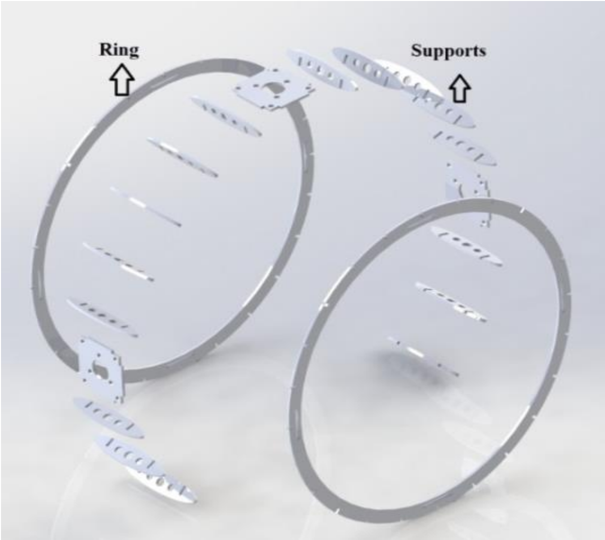

Figure 3.15 Middle Gimbal

3.6.6 Outer Gimbal

This gimbal (Figure 3.16)is the most complex part of the Human Gyroscope. The outer gimbal is assembled by two rings whichincludes 28 supports and 3 couplings. Assembly of the outer gimbal looks like the inner gimbal. But the outer gimbal caries all other gimbals and components in the system.

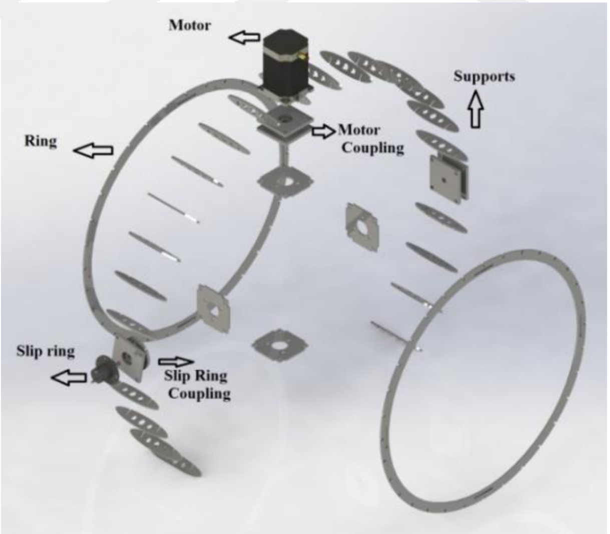

Figure 3.16 Outer Gimbal

3.7 Hardware

All systems need some hardware to be able to move. The flight simulator needs a motor, motor driver, and controller. Suitable equipment is selected based on some criteria. After suitable hardware is selected, the control panel (Figure 3.17) can be made.

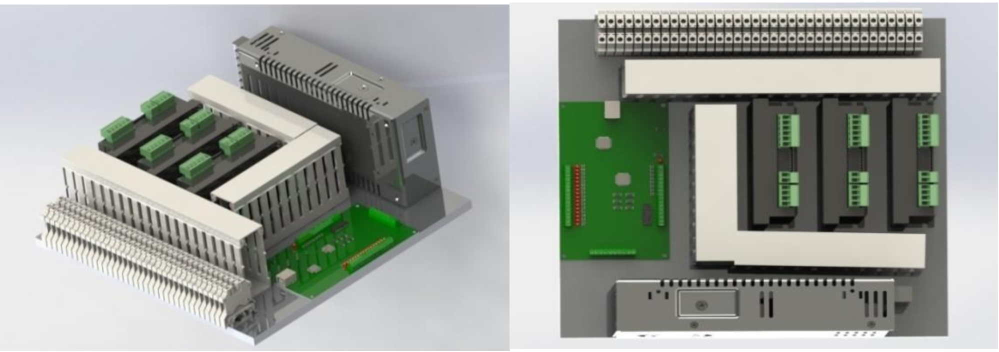

Figure 3.17 Control Box

3.7.1 Stepper Motor

The stepper (Figure 3.18) motor is an electric motor which converts electronic pulses to mechanical rotation. The stepper motor is selected for a prototype Human Gyroscope type flight simulator. The 57HS22 stepper motor is used for this purpose which is the Nema 23 and has the properties of bipolar, 200 steps/revolution, and a step angle of 1.8 degrees. The stepper motor is suitable for controlling the X, Y, Z axis movement of the Human Gyroscope type flight simulator.

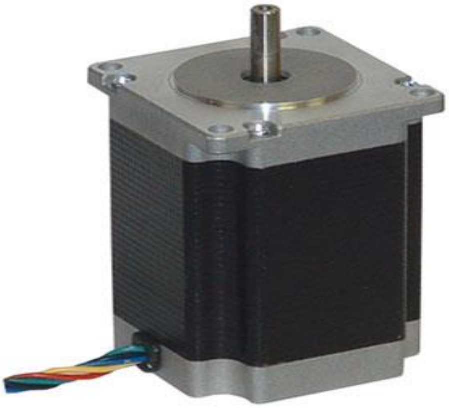

Figure 3.18 Stepper motor

3.7.2 Slip Ring

The important point of the study is transferring the electrical data and power to the stepper motors which are mounted on the gimbals. This is very crucial for a rotating system. This problem is solved by slip rings (Figure 3.19). Slip rings enable the transfer of electric power or signal circuits across a rotating surface, such as those found in radar antennae, gun turrets, periscopes, electro-optic sensor gimbals, and space satellites. So, slip rings are suitable for transferring the electrical signals and power to the stepper motors which are used in Human Gyroscope type flight simulator.

Figure 3.19 Slip Ring

3.7.3 Stepper Motor Driver

The stepper motors are controlled by the CWD 556 (Figure 3.20)stepper motor driver. This driver is economical and suitable for stepper motors. The selection of the motor driver is very important, because the motor and driver must be compatible. For example, the motor driver must accept the current of motor voltage of motor, etc.This motor driver is suitable for a variety of large-scale automation equipment and instruments, such as a cutting machine, packaging machine, plotter, CNC machine, etc. It always performs well when applied for equipment which requires low vibration, low noise, high precision, and high velocity.

Some advantages of motor driver:

- High performance
- Micro step
- Maximum response frequency up to 200 Kpps
- Low temperature rise
- Smooth motion

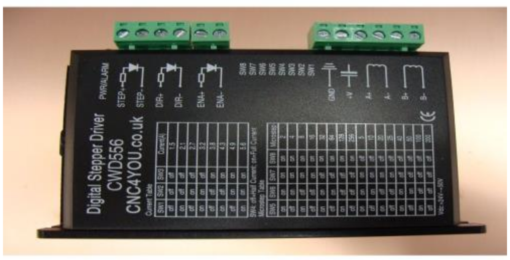

Figure 3.20 Motor Driver

3.7.4 Power Supply

All systems require power, which is provided in different ways. Some systems need

AC 220 V or AC 380 V, DC 9-12 -24-36. Therefore, it is necessary to define what type of energy the system requires. Motor drivers work between 20 to 50 V, so the power supply is selected 24V. Another criteria is power. Power is formulated by P=V*I, where V is voltage. A 57-BHH-82 motor requires 3A, and the system will have 3 stepper motors. So, a 9A power supply(Figure 3.21) is defined for a stepper motor.

Figure 3.21 Power Supply

3.7.5 Control Card

For controlling the Human Gyroscope, the AKZ250 (Figure 3.22) 3 Axis motion control card is used. All connection terminals are clearly explained and made according to the user manual. It has a stable card and is easy to use with a USB communication with the computer.

Connection: PUL+, DIR+ connected to DC5V. PUL- connected to XS and DIR- connected to XD.

Some properties of AKZ250:

- Supports all Mach3 versions
- Supporting Windows series, including Windows2000/XP/Vista/Windows7.
- No need to install any USB drivers, it can be used after plugging in the computer.
- Full support for USB hot-swappable, the card is Monitoring USB connection status at any time.
- Maximum step-pulse frequency is 200KHz, which is suitable for the servo or stepping motor.
- Status indicator LED can be useful to show the USB connection, and working status by flashing.

- 16 general-purpose inputs, with indicators.
- Feed rate, spindle speed rate, or jog rate can by controlled by the adjustment-knob.
- Real-time speed chart and spindle speed changes can be observed

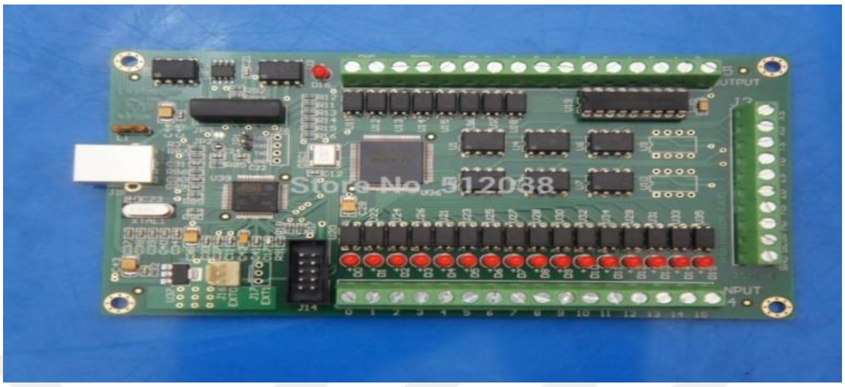

Figure 3.22AKZ 250

3.8 Wiring

This title explains the wiring of Human Gyroscope electronics. First, the wiring of the AKZ250 and CWD 556 step motor driver (Figure 3.23) was completed. Thesetting the wiring of the labelling of the step motor driver and AKZ250, AKZ250 data sheet should be followed.The connections ofthe XS terminal is pulse and XD terminal aredirection must be defined. Then, the motor driver needs to be selected. That is represented by mean + terminal, or – terminal AKZ250, which is need. The PUL- connects the XS, DIR- connects the XD, and the other PUL+, DIR+, ENA+ connect with each other to complete the proper wiring of the control car and stepper motor driver.

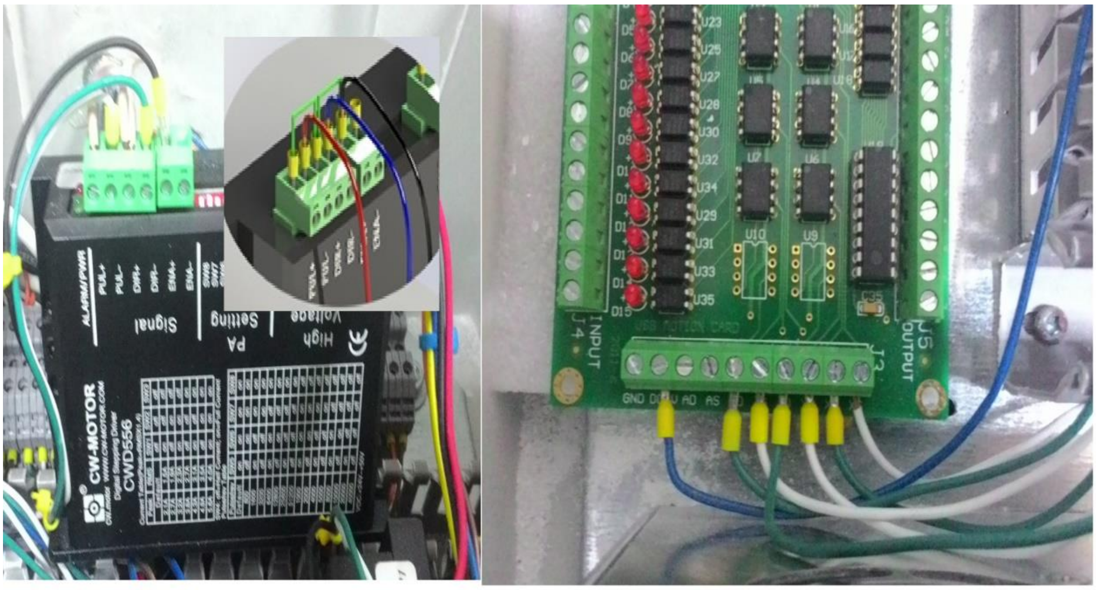

Figure 3.23 Connection Between Driver and Controller

After connection of the motor driver and AKZ250 have been completed, the stepper motor and motor driver(Figure 3.24) connections have been completed according to the data sheet of the motor and motor driver reference book. In the motor data sheet, the name of the wire Red and Blue is A+, Black and Yellow is A-, Brown and White is B+, and Orange and Green is B-. The motor has some type of connection. The selection of the connections is defined according to the types based on the system requirement.

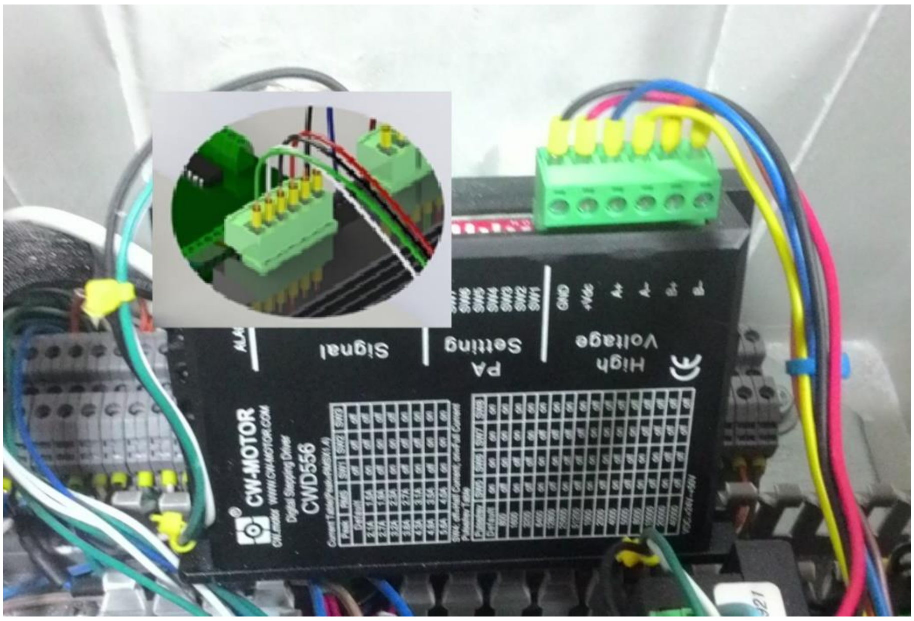

Figure 3.24 Connection Between Motor and Driver

The last connection is power and motor driver. CWD 556 can work with a 20-50 V DC. So, a 24 V power supply is suitable for the motor driver. In there, connect the + and – terminal. After all connection control box (Figure 3.25) was finished.

Figure 3.25 Full Connection of Control Box

3.9 Visual Simulation System

Visual simulation is the last part of this study. The aim of the VSS is to take data from the Human Gyroscope and visualize the motion in a computer program. So, this system needs some hardware and software. For the hardware, the electronic components of Arduino, gyro, and telemetry is used.

3.9.1 Arduino

Arduino(Figure 3.26) is an open-source electronics platform based on easy-to-use hardware and software [33].Arduino is the main part of the system. This board controls gyro and telemetry. Arduino takes data from gyro and sends with telemetry.

Figure 3.26 Arduino Control Board

3.9.2 MPU6050

The InvenSense MPU-6050 (Figure 3.27)sensor contains a MEMS accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains a 16- bit analogue to digital conversion hardware for each channel. Therefore, it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus [34] to interface with the Arduino.

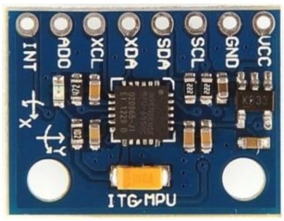

Figure 3.27MPU6050

3.9.3 Telemetry

Telemetry (Figure 3.28) provides wireless communication between the PC and Arduino.

Figure 3.28 Telemetry

3.9.4 Build Circuit Board

A pertinax board is used for the circuit board. This board makes it easy to create a prototype circuit and soldering. In this circuit there are two main categories. The first one is the gyro connection and the second one is the wireless connection. These connections are shown in (Figure 3.29). All connections are soldered on the pertinax, which connects to the Arduino and looks like an Arduino shield.

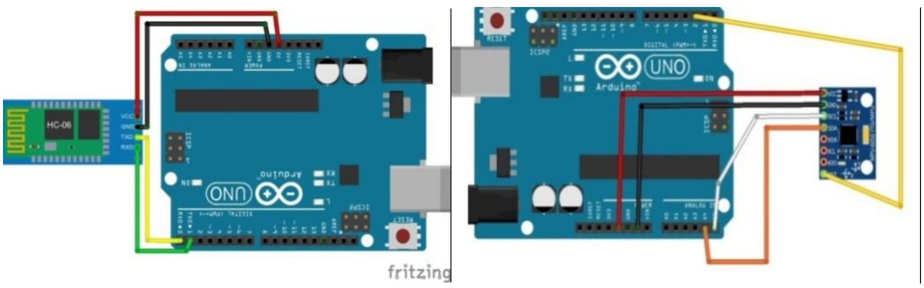

Figure 3.29 Connection Diagram

CHAPTER 4

EXPERIMENTAL STUDY

4.1 Flight Control Code Generator

In this thesis, a controlsoftware is developed for controlling the Human Gyroscope type flight simulator. Software is prepared by C++ and Visual C# in Visual Studio 2015. This software consists of 2 parts; the first is the random flight path codegenerator and the second is the user -defined flight path codegenerator. These two software parts develop a flight path code for the Human Gyroscope. The control board works with CNC G-code, so the flight path code must be compatible with G-code.

4.2 Usage of Software

The program (Figure 4.1) has a very simple and useful interface. Flight Path Code Generator also has somemenu bars. The desired operations can be performed via these menu bars.

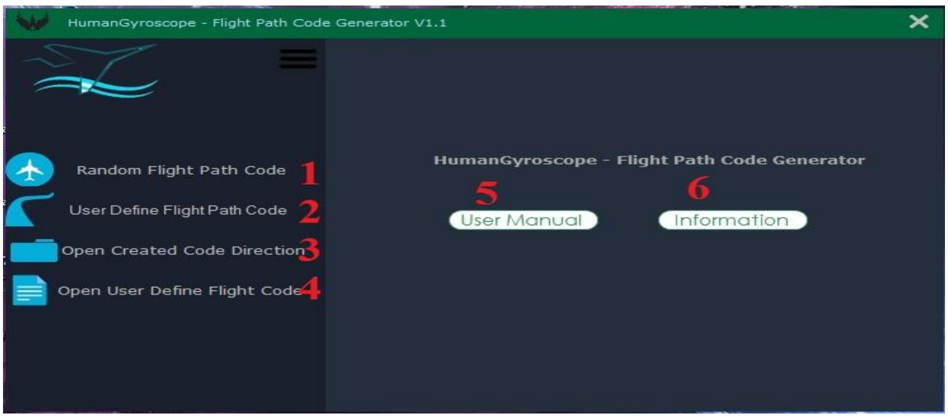

Figure 4.1 User Interface

1- Random Flight Path Code; after opening the program, Random Flight Path Code Generatormenu opens when therandom flight path code button is clicked the menu number 1 appears in the menu bar (Figure 4.2).

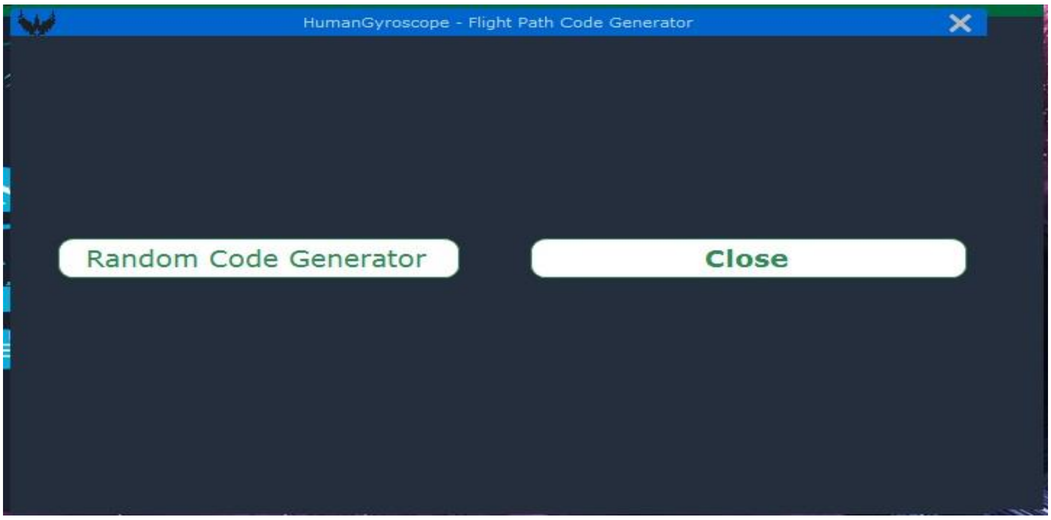

Figure 4.2: Random Flight Path Window

When theRandom Code Generator button is clicked, theprogram generatesrandom flight path. If click “Close” button, menu is close.

2- User Define Flight Path Code; When this menu bar is clicked, a new window is opened. (Figure 4.3).

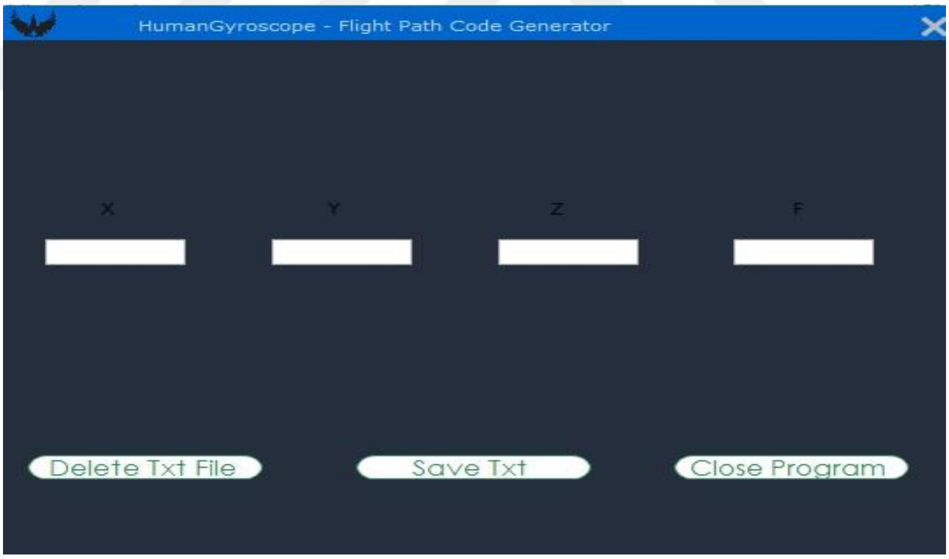

Figure 4.3 User Define Flight Path Window

After opening this window, click "Delete Txt File" button. Then the desired codes are entered in the axis boxes. After each axis box is filled, it is clicked on the menu "Save Txt" and after this process the program asks, "Do You Want to Continue". If “Yes” is selected, the code entering is active. If "Close Program" is selected, the program will close.

3- Open Created Code Direction; This menu bar opens the location where the files are saved. (Figure 4.4).

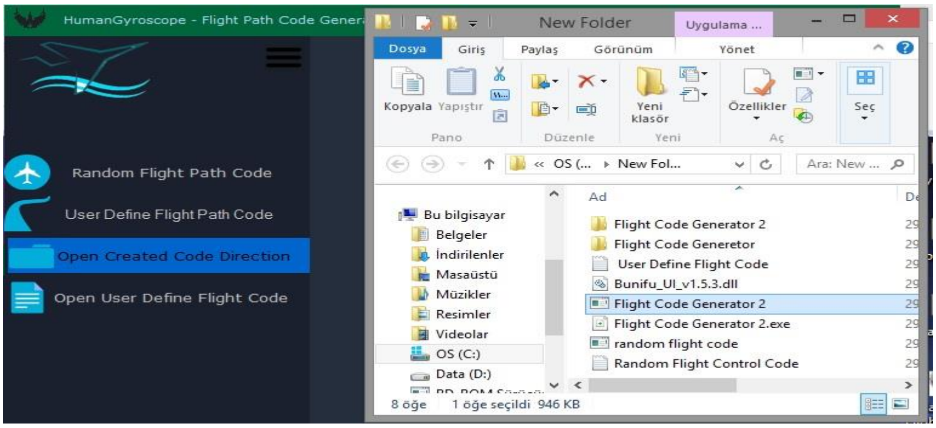

Figure4.4 Folder Direction

4- Open User Define Flight Code; This menu opens the txt file of flight path (Figure 4.5) for checking any error.

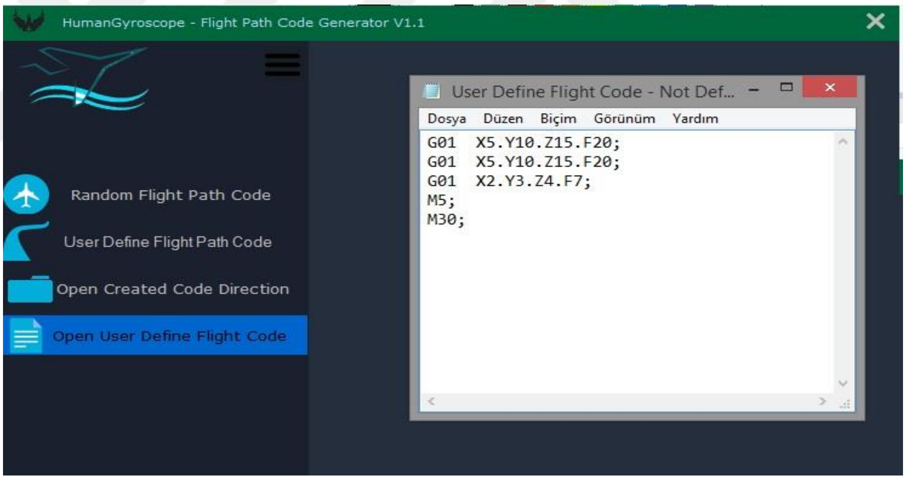

Figure 4.5 Txt File

5- User Manual: This menu opens the user manual (Figure 6) file for giving information about program.

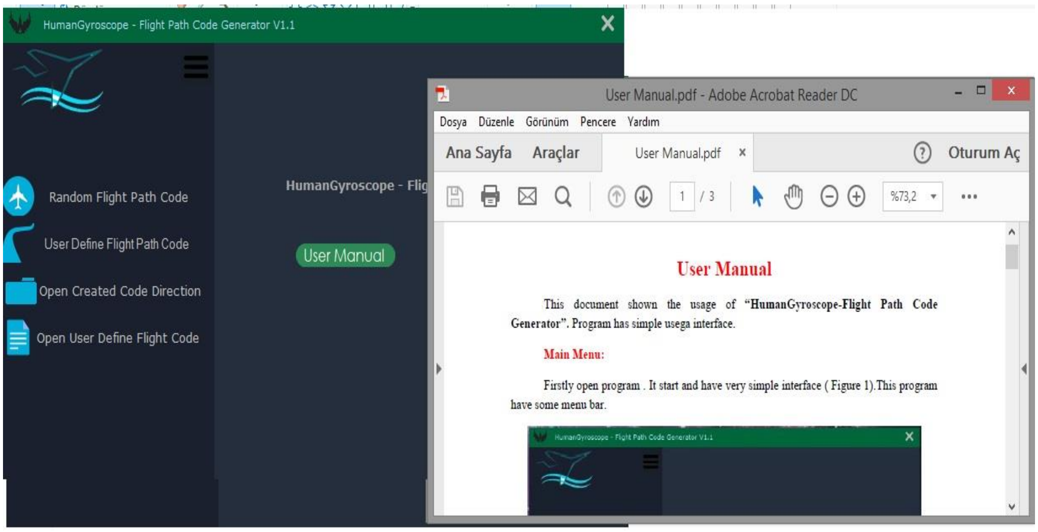

Figure 4.6 User Manual

6- Information: This menugives information about program,for example version, etc.(Figure 4.7).

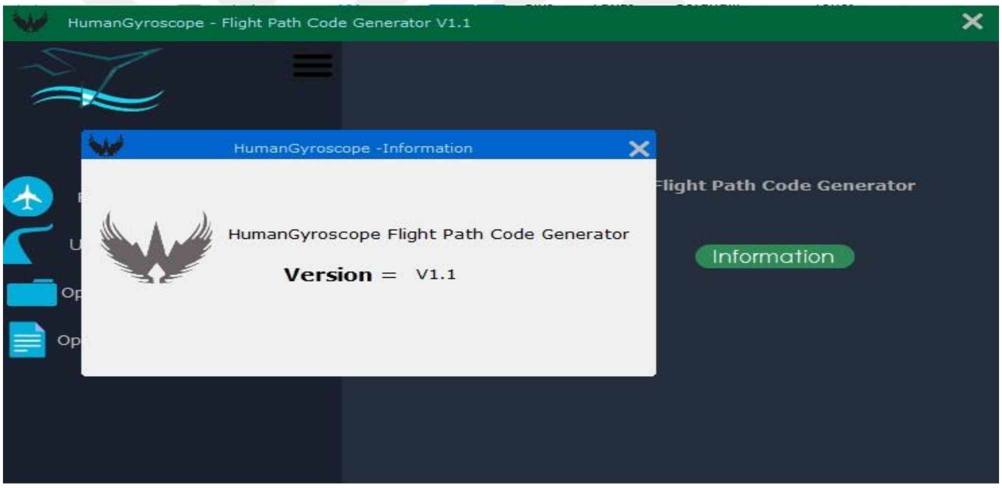

Figure4.7 Information Screen

After creating the flight path, the .txt file is used in the motion control software, called "MACH3.exe". MACH3 (Figure 4.8)is motion control software that is used for 5 axis motion control. A Human Gyroscope type flight simulator needs 3 axes motion control, so it's suitable for the Human Gyroscope control system.

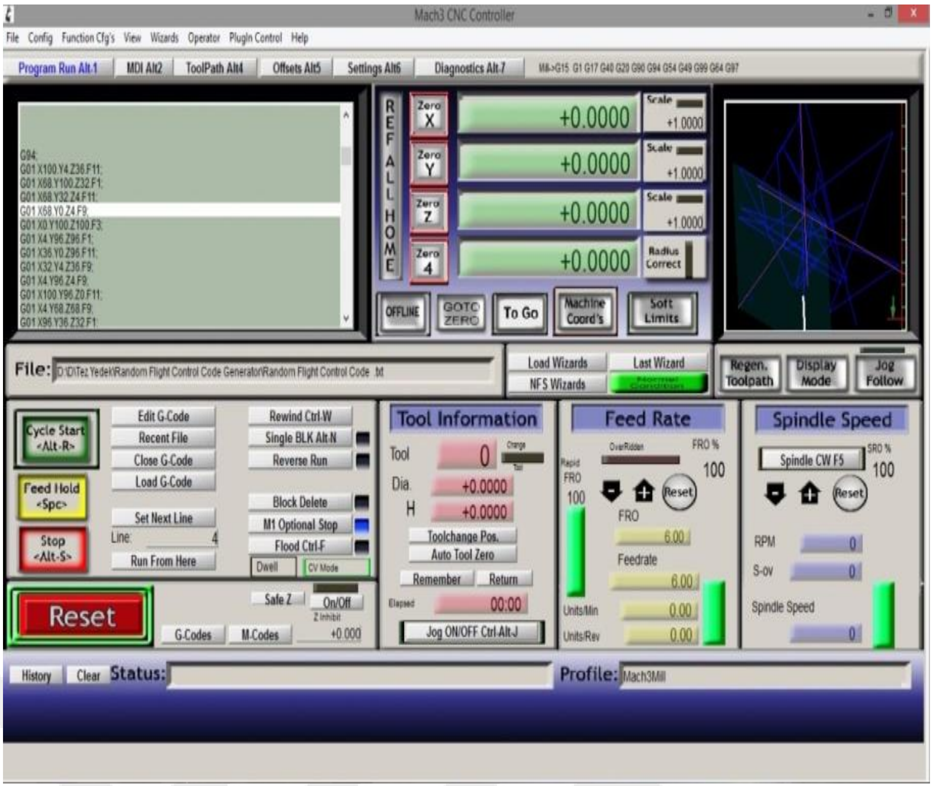

Figure 4.8 Mach 3 Interface

4.3 Motion Tracing System

The motion tracking system checks the accuracy of the Human Gyroscope. For instance, if the system turns around once, the motion tracking system shows this motion visually. The system needs both hardware and software components. The first hardware component is Arduino. The second is MPU6050 (6 Axis Gyro), and the last is telemetry.

4.4. Visualization of Motion

The motion tracking system collects data from the Human Gyroscope. After collecting data, the motion tracking system sends this data to the computer with telemetry. This data is raw, so some software must be developed. Processing is suitable for developed software for visualization of motion (Figure 4.9). After converting data in processing, the motion can be seen on the monitor.

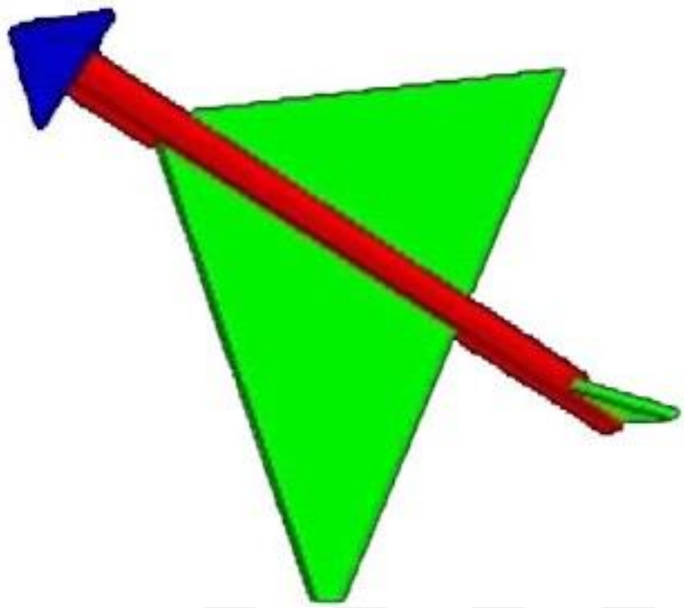

Figure 4.9 Processing Output

4.4.1 Examples of Motion

Visual simulation system has been designed to check the correctness of the movements given after the Human Gyroscopestarted to work.Electronic hardware is activated before moving the system.Some time is required to calibration. After the calibration is completed, the system is moved.HumanGyroscopeis run and the accuracy of the movements on the computer monitor is examined.While looking at the seat as shown in Figure 4.10, the model on the computer is in the same position.

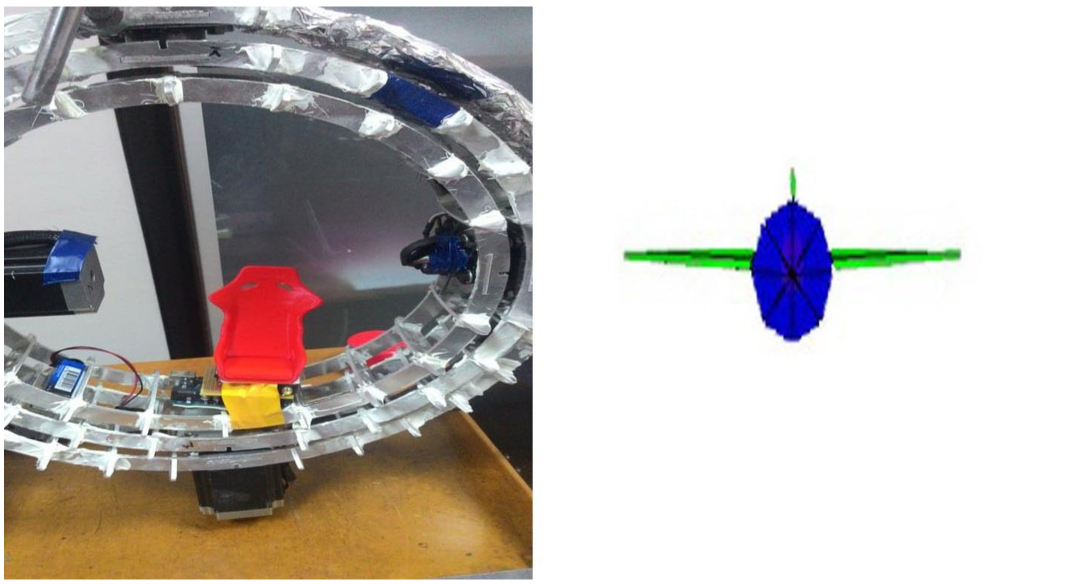

Figure4.10 Position 1

Whenthesystem is movedupwards(Figure 4.11), theaircraftlooksupwards in thesamepattern.

Figure 4.11 Position 2

Finally, when the system is moved downwards (Figure 4.12),the model aircraft is looking downwards. As it is seen in the examples made in this way, visual simulation system works smoothly.

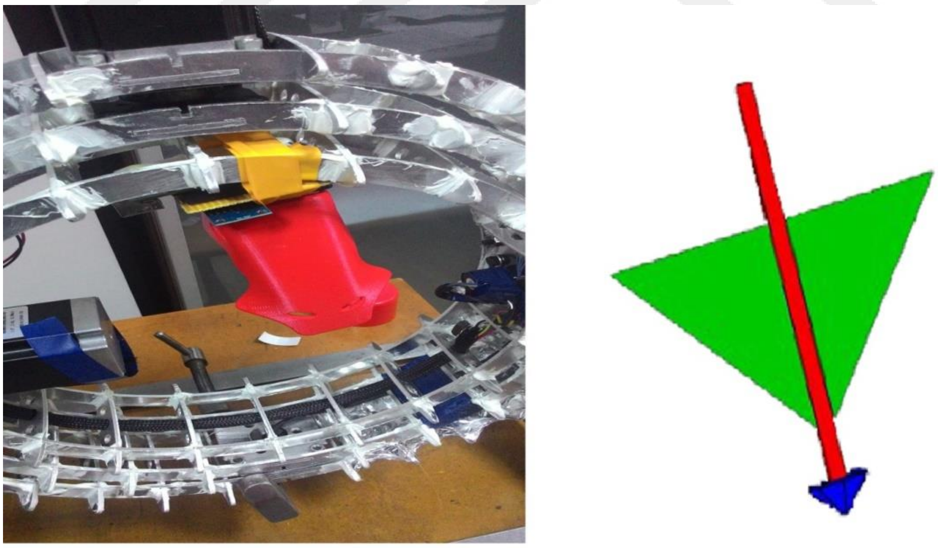

Figure 4.12 Position 3

4.5 Flight Path

The flight path includes two main categories: random flight path and user defined flight path. This is so important for simulation and the pilot's education. All conditions must be simulated virtually. So, these two flight paths include airplane manoeuvres.

4.5.1 Random Flight Path

In random flight paths, each movement condition is created randomly. All manoeuvres, motions, and velocities are different. So, when random flight paths are run, all simulation is different from each other. the random flight path has some advantages,for example, when the air condition changes, or mechanical faults happensthis flight simulation is unpredictable.

This program is coded in C ++ program. In this code, random variables are used. The important point in coding is random variables. As seen in App A, random variables are controlled.

Example of random flight path code;

```
G94;
G01 X64.Y36.Z36.F3;
G01 X64.Y100.Z68.F3;
G01 X64.Y68.Z36.F11;
M05;
M30;
```

4.5.2 User Define Flight Path

This type of flight path is created by people or takes data from airplanes. So, this type of flight path is very important. In this way, the machine can simulate a real flight. Sometimes trainers teach specific manoeuvres or trainers need to challenge their pilots, so the trainers create a flight path for these scenarios.

Example of user defines flight path code;

```
G94;
G01 X10.Y5.Z8.F6;
G01 X35.Y45.Z60.F15;
G01 X20.Y15.Z80.F5;
M5;
M30;
```

CONCLUSION

The Flight Simulation systems are used in aviation training area. Without simulators, the safety and economies of air transportation would be negotiated. The air forces increasing rely on simulation for crew training, mainly for aspects which are expensive or difficult to train in the aircraft itself. The design, development and testing of new flight simulators need extensive simulation facilities, thereby enhancing safety, reducing timescales and so, costs. other applications include research into human factors, and accident investigation.

Simulators can be operated at intensive rates by day and night, and can train for any condition, weather, time or location which is included in their data base. The traditional simulators are less expensive to operate and can perform the simplest aircraft manoeuvre. They actuate the movements by using the steward-platform, in this method the movements are limited due to the movement capability of the steward-platform. So, the high movement capable military aircraft and acrobatic aircraft movements cannot be actuated by this method.

In this study, a new type of flight simulator was designed, and a prototype was manufactured for military crew training and acrobatic flight training. A traditional fun fair equipment of Human Gyroscope was modified to a flight simulator which have motor driven three rotational axis. This fully controlled Human Gyroscope type flight simulator can perform infinite rotational movement in each axis. So, nearly all flight scenarios may be simulated by this way such as civil aircraft, military aircraft, acrobatic aircraft and helicopter manoeuvres.

A new software was developed for this purpose. The software is used for two main purpose, one is driving the Human Gyroscope type flight simulator, other is collecting data from the pilot to simulate the pilot’s capability. The manoeuvre was defined according to the actual flight condition. The first flight manoeuvre was defined as random flight scenario which may simulate random movement of any aircraft flight. The second flight scenario was defined as user defined flight path which may simulate the defined manoeuvre and conditions of any aircraft flight.

In this thesis, A new flight simulator was designed, and stepper motor driven prototype was manufactured.

REFERENCES

[1] Jean Le Bouthillier, R., 2010, " The Design, Implementation and Analysis of a Low Cost Flight Simulator Using RMC's 3DOF Motion Platform ", Msc Thesis in the Royal Military College of Canada.

[2] J Nelson, K.R., 1994, "Human-powered gyroscope", Google Patents.

[3] "www.bestonkidsrides.com/human-gyroscope-ride-for-sale/". (15.06.2017)

[4] Thöndel, E.,2010, "Electric motion platform for use in simulation technology–design and optimal control of a linear electromechanical actuator". in *Proceedings of the World Congress on Engineering and Computer Science.*ISBN: 978-988-18210-0-3.

[5] Erçin, G.H., 2008, "Flight Simulator and Control of Helicopter. " Msc Thesis in theMiddle East Technical University: p. 106.

[6] Lancaster, J.A.,2004, "Investigating pilot performance using mixed-modality simulated data link".Phd Thesis in the Virginia PolytechnicInstituteandStateUniversity.

[7] Root, E., (2004),"A Re-configurable Hardware-in-the-loop Flight Simulator", Msc Thesis in theOhio University.

[8] Kaith, I.,1960, "The design and evaluation of a flight simulator apparatus, Msc Thesis in theUniversity of Arizona. ".

[9] Abbasi, H.,2010,"Simulation of flexible aircraft".Msc Thesis in the Toronto University.

[10] Stueck, P.G.,1973, "LSO/pilot interaction simulator"Msc Thesis in theNaval Postgraduate School.

[11] Sherman, K.L., 1992,"The design and construction of a hang glider flight simulator".Msc Thesis in theRochester Institute of Technology.

[12] Albayrak, O., 2005, "Modelling And Real-time Control System Implementation For A Steward Platform" Msc Thesis in the Middle East Technical University.

[13] Cristofaro, M.,2014,"Elements od Computational Flight Dynamics for Complete Aircraft", Msc Thesis in theSouthamptonUniversity.

[14] Çoban, U., 2010,"Design of a Reconfigurable Flight Display Unit for a Network Flight Simulator", Msc Thesis in the Istanbul Technical University.

[15] Kesserwan, N.,1999, "Flight Simulation", Msc Thesis in theMcGîll University, Montreal.

[16] Bauer, M., 2005, "Evaluating The Effectiveness Of Training System Approaches For Highly Complex Flight Training".PhD Thesis in the Central FloridaUniversity.

[17] Ackerman, K., 2014, "Development of a pilot-in-the-loop flight simulator using NASA's transport class model".Msc Thesis in the IllinoisUniversity.

[18] Martínez, B., 2017, "Flight Simulator Development with Safety System Implementation".Bachelor's Thesis in the Polytechnic University of Catalonia.

[19] Lawn, P., 1998,"The enhancement of a flight simulator system with teaching and research applications", Msc Thesis in the Concordia University.

[20] Shi, W.,2010, "Electro-Hydraulic Servo-Valve and Motion and Control Loading of Full Flight Simulator",Msc Thesis in the Ryerson University.

[21] Plumpton, J., 2014,"Development of a Full-flight Simulator for Ab-initio Flight Training with Emphasis on Hardware and Motion Integration", Carleton University, Msc Thesis in the Carleton University.

[22] Zeitoun, O.,2014, "A Distributed System Interface for a Flight Simulator". Msc Thesis in the Mcmaster University.

[23] DiBello, B.J., 2006,"Developing effective management flight simulators", Msc Thesis in theMassachusetts Institute of Technology.

[24] Brownridge, A.M. ,2014"Real-time motion capture for analysis and presentation within virtual environments", PhD Thesis in the Manchester Metropolitan University.

[25] Homan, W.J. ,1997,"Effects of Multimedia Pretraining on Pilot's Simulator Performance".Msc Thesis in the Northern Arizona University.

[26] Robinson, T.D. , 2004,"A real-time Interactive Non-linear Flight Simulator of a Piloted Ornithopter".Msc Thesis in the University of Toronto.

[27] Sagoo, G.K., 2008,"Pilot in Loop Assessment of Fault Tolerant Flight Control Schemes in a Motion Flight Simulator".Msc Thesis in the West Virginia University.

[28] Rekleitis, G. ,2002, "Modeling and Evaluation of a New Parallel Mechanism as a Flight Simulator Motion Base".Msc Thesis in the McGill University.

[29] Lee, P.T.S. , 2004, "Determination of the Human Perception Threshold of Phase Difference Between Motion and Visual Cues in a Moving-based Flight Simulator". Msc Thesis in the University of Toronto.

[30] Li, D. and S. Salcudean., 1997, "Modeling, simulation, and control of a hydraulic Stewart platform". in Robotics and Automation. Proceedings., *1997 IEEE International Conference on*, April 20-25 1997,**4** ,3360-3366.

[31] Runevad, M. and A. Kjellin,2012, "The Human Gyroscope: Motor Driven Simulator with a Gyroscope Design, Bachelor Thesis in the Halmstad University ".

[32] Rolfe JM, Staples, KJ,1988,“Flight simulation.”,V1,Cambridge University Press.

[33] "www.arduino.cc/en/Tutorial/LibraryExamples". (20.08.2017)

[34] "http://www.github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library" (20.08.2017

APPENDIX

Appendix A

C++ Code
Random Flight Path Code:

```
#include <iostream>
#include <fstream>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
using namespace std;
int main()
{
srand(time(NULL));
ofstreampoz("Random Flight Control Code  .txt",ios::out);
intx,y,z,f,k;
   k=rand()%100;
cout<<k<<endl<<"CODE GENERATED, PLEASE PUSH THE ENTER
BUTTON...";
poz<<"G94;"<<endl;
for(int i=1;i<=k;i++){
        x=rand()&100;
         y=rand()&100;
         z=rand()&100;
          f=rand() &10;
poz<<"G01"<<"X"<<x<<"."<<"Y"<<y<<"."<<"Z"<<z<<"."<<"F"<<f+1<<";"<<end
l;
}
poz.close();
cin.get();
return 0;
}
```

Appendix B

Arduino Code;

```
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
   #include "Wire.h"
#endif
```

```
MPU6050 mpu;
//#define OUTPUT_READABLE_QUATERNION
//#define OUTPUT_READABLE_EULER
//#define OUTPUT_READABLE_YAWPITCHROLL
//#define OUTPUT_READABLE_REALACCEL
//#define OUTPUT_READABLE_WORLDACCEL
#define OUTPUT_TEAPOT
#define INTERRUPT_PIN 2
#define LED_PIN 13
boolblinkState = false;
booldmpReady = false;
uint8_tmpuIntStatus;
uint8_tdevStatus;
uint16_tpacketSize;
uint16_tfifoCount;
uint8_tfifoBuffer[64];
Quaternion q;
VectorInt16 aa;
VectorInt16 aaReal;
VectorInt16 aaWorld;
VectorFloat gravity;
floateuler[3];
floatypr[3];
volatile bool mpuInterrupt = false;
voiddmpDataReady() {
mpuInterrupt = true;
}
//
==========================================================
====
// ===                    INITIAL SETUP                    ===
//
==========================================================
====

void setup() {
   #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
Wire.begin();
Wire.setClock(400000);
   #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
Fastwire::setup(400, true);
   #endif
Serial.begin(57600);
while (!Serial);
Serial.println(F("Initializing I2C devices..."));
mpu.initialize();
pinMode(INTERRUPT_PIN, INPUT);
Serial.println(F("Testing device connections..."));
Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") :
F("MPU6050 connection failed"));
```

```
Serial.println(F("\nSend any character to begin DMP programming and demo: "));
while (Serial.available() &&Serial.read());
while (!Serial.available());
while (Serial.available() &&Serial.read());
Serial.println(F("Initializing DMP..."));
devStatus = mpu.dmpInitialize();
mpu.setXGyroOffset(220);
mpu.setYGyroOffset(76);
mpu.setZGyroOffset(-85);
mpu.setZAccelOffset(1788);
if (devStatus == 0) {
Serial.println(F("Enabling DMP..."));
mpu.setDMPEnabled(true);
Serial.println(F("Enabling interrupt detection (Arduino external interrupt 0)..."));
attachInterrupt(digitalPinToInterrupt(INTERRUPT_PIN), dmpDataReady, RISING);
mpuIntStatus = mpu.getIntStatus();
Serial.println(F("DMP ready! Waiting for first interrupt..."));
dmpReady = true;
packetSize = mpu.dmpGetFIFOPacketSize();
   } else {
Serial.print(F("DMP Initialization failed (code "));
Serial.print(devStatus);
Serial.println(F(")"));
   }
pinMode(LED_PIN, OUTPUT);
}
=================================================================
====
// ===                   MAIN PROGRAM LOOP                   ===
//
=================================================================
====

void loop() {
if (!dmpReady) return;
while (!mpuInterrupt&&fifoCount<packetSize) {
   }
mpuInterrupt = false;
mpuIntStatus = mpu.getIntStatus();
fifoCount = mpu.getFIFOCount();
if ((mpuIntStatus& 0x10) || fifoCount == 1024) {
mpu.resetFIFO();
Serial.println(F("FIFO overflow!"));
   } else if (mpuIntStatus& 0x02) {
while (fifoCount<packetSize) fifoCount = mpu.getFIFOCount();
mpu.getFIFOBytes(fifoBuffer, packetSize);
fifoCount -= packetSize;
        #ifdef OUTPUT_READABLE_QUATERNION
            // display quaternion values in easy matrix form: w x y z
mpu.dmpGetQuaternion(&q, fifoBuffer);
```

```
Serial.print("quat\t");
Serial.print(q.w);
Serial.print("\t");
Serial.print(q.x);
Serial.print("\t");
Serial.print(q.y);
Serial.print("\t");
Serial.println(q.z);
        #endif
        #ifdef OUTPUT_READABLE_EULER
            // display Euler angles in degrees
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetEuler(euler, &q);
Serial.print("euler\t");
Serial.print(euler[0] * 180/M_PI);
Serial.print("\t");
Serial.print(euler[1] * 180/M_PI);
Serial.print("\t");
Serial.println(euler[2] * 180/M_PI);
        #endif
        #ifdef OUTPUT_READABLE_YAWPITCHROLL
            // display Euler angles in degrees
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);
mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
Serial.print("ypr\t");
Serial.print(ypr[0] * 180/M_PI);
Serial.print("\t");
Serial.print(ypr[1] * 180/M_PI);
Serial.print("\t");
Serial.println(ypr[2] * 180/M_PI);
        #endif
        #ifdef OUTPUT_READABLE_REALACCEL
            // display real acceleration, adjusted to remove gravity
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetAccel(&aa, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);
mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);
Serial.print("areal\t");
Serial.print(aaReal.x);
Serial.print("\t");
Serial.print(aaReal.y);
Serial.print("\t");
Serial.println(aaReal.z);
        #endif
        #ifdef OUTPUT_READABLE_WORLDACCEL
mpu.dmpGetQuaternion(&q, fifoBuffer);
mpu.dmpGetAccel(&aa, fifoBuffer);
mpu.dmpGetGravity(&gravity, &q);
mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);
```

```
mpu.dmpGetLinearAccelInWorld(&aaWorld, &aaReal, &q);
Serial.print("aworld\t");
Serial.print(aaWorld.x);
Serial.print("\t");
Serial.print(aaWorld.y);
Serial.print("\t");
Serial.println(aaWorld.z);
        #endif
        #ifdef OUTPUT_TEAPOT
teapotPacket[2] = fifoBuffer[0];
teapotPacket[3] = fifoBuffer[1];
teapotPacket[4] = fifoBuffer[4];
teapotPacket[5] = fifoBuffer[5];
teapotPacket[6] = fifoBuffer[8];
teapotPacket[7] = fifoBuffer[9];
teapotPacket[8] = fifoBuffer[12];
teapotPacket[9] = fifoBuffer[13];
Serial.write(teapotPacket, 14);
teapotPacket[11]++;
        #endif
blinkState = !blinkState;
digitalWrite(LED_PIN, blinkState);
    }
}
```

Appendix C

Processing Code

```
importprocessing.serial.*;
importprocessing.opengl.*;
importtoxi.geom.*;
importtoxi.processing.*;
ToxiclibsSupportgfx;
Serial port;
char[] teapotPacket = new char[14];
intserialCount = 0;
int synced = 0;
int interval = 0;
float[] q = new float[4];
Quaternion quat = new Quaternion(1, 0, 0, 0);
float[] gravity = new float[3];
float[] euler = new float[3];
float[] ypr = new float[3];
void setup() {
size(300, 300, OPENGL);
gfx = new ToxiclibsSupport(this);
lights();
smooth();
println(Serial.list());
    String portName = Serial.list()[0];
```

```
port = new Serial(this, portName, 57600);
port.write('r');
}
void draw() {
if (millis() - interval > 1000) {
port.write('r');
interval = millis();
    }
background(0);
pushMatrix();
translate(width / 2, height / 2);
float[] axis = quat.toAxisAngle();
rotate(axis[0], -axis[1], axis[3], axis[2]);


fill(255, 0, 0, 200);
box(10, 10, 200);
fill(0, 0, 255, 200);
pushMatrix();
translate(0, 0, -120);
rotateX(PI/2);
drawCylinder(0, 20, 20, 8);
popMatrix();
fill(0, 255, 0, 200);
beginShape(TRIANGLES);
vertex(-100,  2, 30); vertex(0,  2, -80); vertex(100,  2, 30);
vertex(-100, -2, 30); vertex(0, -2, -80); vertex(100, -2, 30);
vertex(-2, 0, 98); vertex(-2, -30, 98); vertex(-2, 0, 70);
vertex( 2, 0, 98); vertex( 2, -30, 98); vertex( 2, 0, 70);
endShape();
beginShape(QUADS);
vertex(-100, 2, 30); vertex(-100, -2, 30); vertex(  0, -2, -80); vertex(  0, 2, -80);
vertex( 100, 2, 30); vertex( 100, -2, 30); vertex(  0, -2, -80); vertex(  0, 2, -80);
vertex(-100, 2, 30); vertex(-100, -2, 30); vertex(100, -2,  30); vertex(100, 2,  30);
vertex(-2,   0, 98); vertex(2,   0, 98); vertex(2, -30, 98); vertex(-2, -30, 98);
vertex(-2,   0, 98); vertex(2,   0, 98); vertex(2,   0, 70); vertex(-2,   0, 70);
vertex(-2, -30, 98); vertex(2, -30, 98); vertex(2,   0, 70); vertex(-2,   0, 70);
endShape();
popMatrix();
}
voidserialEvent(Serial port) {
interval = millis();
while (port.available() > 0) {
intch = port.read();
if (synced == 0 &&ch != '$') return;
synced = 1;
print ((char)ch);
if ((serialCount == 1 &&ch != 2)
        || (serialCount == 12 &&ch != '\r')
        || (serialCount == 13 &&ch != '\n'))  {
```

```
serialCount = 0;
synced = 0;
return;
        }
if (serialCount> 0 || ch == '$') {
teapotPacket[serialCount++] = (char)ch;
if (serialCount == 14) {
serialCount = 0;
q[0] = ((teapotPacket[2] << 8) | teapotPacket[3]) / 16384.0f;
q[1] = ((teapotPacket[4] << 8) | teapotPacket[5]) / 16384.0f;
q[2] = ((teapotPacket[6] << 8) | teapotPacket[7]) / 16384.0f;
q[3] = ((teapotPacket[8] << 8) | teapotPacket[9]) / 16384.0f;
for (int i = 0; i < 4; i++) if (q[i] >= 2) q[i] = -4 + q[i];
quat.set(q[0], q[1], q[2], q[3]);
            }
        }
    }
}
voiddrawCylinder(float topRadius, float bottomRadius, float tall, int sides) {
float angle = 0;
floatangleIncrement = TWO_PI / sides;
beginShape(QUAD_STRIP);
for (int i = 0; i < sides + 1; ++i) {
vertex(topRadius*cos(angle), 0, topRadius*sin(angle));
vertex(bottomRadius*cos(angle), tall, bottomRadius*sin(angle));
angle += angleIncrement;
    }
endShape();
if (topRadius != 0) {
angle = 0;
beginShape(TRIANGLE_FAN);

        // Center point
vertex(0, 0, 0);
for (int i = 0; i < sides + 1; i++) {
vertex(topRadius * cos(angle), 0, topRadius * sin(angle));
angle += angleIncrement;
        }
endShape();
    }
if (bottomRadius != 0) {
angle = 0;
beginShape(TRIANGLE_FAN);
        // Center point
vertex(0, tall, 0);
for (int i = 0; i < sides + 1; i++) {
vertex(bottomRadius * cos(angle), tall, bottomRadius * sin(angle));
angle += angleIncrement;
        }
endShape();
```

```
    }
}
```

Appendix D:

Visual C# Code;
Form 1

```
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;

namespace Flight_Code_Generator_2
{
publicpartialclassForm1 :Form
    {
public Form1()
        {
InitializeComponent();
        }

privatevoid Form1_Load(object sender, EventArgs e)
        {

        }

privatevoid panel1_Paint(object sender, PaintEventArgs e)
        {

        }

privatevoid pictureBox1_Click(object sender, EventArgs e)
        {

        }

privatevoid bunifuCustomLabel1_Click(object sender, EventArgs e)
        {

        }

privatevoid bunifuImageButton2_Click(object sender, EventArgs e)
```

```
        {
this.Close();
        }

privatevoid bunifuFlatButton1_Click(object sender, EventArgs e)
        {
Form2 usercode1 = newForm2();
usercode1.Show();
        }

privatevoid bunifuThinButton21_Click(object sender, EventArgs e)
        {
Form4 usercode3 = newForm4();
usercode3.Show();
        }

privatevoid bunifuThinButton22_Click(object sender, EventArgs e)
        {


        }

privatevoid bunifuFlatButton2_Click(object sender, EventArgs e)
        {
System.Diagnostics.Process.Start(@"C:\New Folder");
        }

privatevoid bunifuFlatButton3_Click(object sender, EventArgs e)
        {
System.Diagnostics.Process.Start(@"C:\New Folder\ User Define Flight Code.txt");
        }

privatevoid bunifuFlatButton5_Click(object sender, EventArgs e)
        {
Form3 usercode2 = newForm3();
usercode2.Show();
        }

privatevoid bunifuThinButton23_Click(object sender, EventArgs e)
        {
System.Diagnostics.Process.Start(@"C:\New Folder\User Manuel.pdf");
        }
    }
}
```

Form2;

```
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
```

```
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;

namespace Flight_Code_Generator_2
{
publicpartialclassForm2 :Form
    {
public Form2()
        {
InitializeComponent();
        }

privatevoid bunifuImageButton2_Click(object sender, EventArgs e)
        {
this.Close();
        }

privatevoid bunifuThinButton21_Click(object sender, EventArgs e)
        {
System.Diagnostics.Process.Start(@"C:\New Folder\random flight code.exe");
MessageBox.Show("Code Created!", "Attention");
        }

privatevoid bunifuThinButton22_Click(object sender, EventArgs e)
        {
this.Close();
        }
    }
}
```

Form 3;

```
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;
using System.IO;

namespace Flight_Code_Generator_2
{
publicpartialclassForm3 :Form
```

```
{
public Form3()
        {
InitializeComponent();
        }

privatevoid bunifuImageButton2_Click(object sender, EventArgs e)
        {
this.Close();
        }

privatevoid bunifuThinButton23_Click(object sender, EventArgs e)
        {
this.Close();
        }

privatevoid bunifuThinButton21_Click(object sender, EventArgs e)
        {

if (System.IO.File.Exists(@"C:\New Folder\ User Define Flight Code.txt"))
            {
System.IO.File.Delete(@"C:\New Folder\ User Define Flight Code.txt");
MessageBox.Show(" File Deleted", "Succes!");
            }
else
            {
MessageBox.Show("There is no File ", "Cannot Delete");

            }
        }

privatevoid bunifuThinButton22_Click(object sender, EventArgs e)
        {
StreamWriterdosya = File.AppendText("C:\\New Folder\\ User Define Flight
Code.txt");

dosya.Write("G01  ");
dosya.Write("X");
dosya.Write(textBox1.Text);
dosya.Write(".Y");
dosya.Write(textBox2.Text);
dosya.Write(".Z");
dosya.Write(textBox3.Text);
dosya.Write(".F");
dosya.Write(textBox4.Text);
dosya.WriteLine(";");
DialogResultsecenek = MessageBox.Show("Do you wat to continue to write code?",
"Information Window", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
if (secenek == DialogResult.Yes)
            {
```

```
dosya.Write("G01  ");
dosya.Write("X");
dosya.Write(textBox1.Text);
dosya.Write(".Y");
dosya.Write(textBox2.Text);
dosya.Write(".Z");
dosya.Write(textBox3.Text);
dosya.Write(".F");
dosya.Write(textBox4.Text);
dosya.WriteLine(";");
MessageBox.Show("Next Line");
          }
elseif (secenek == DialogResult.No)
          {
dosya.WriteLine("M5;");
dosya.WriteLine("M30;");
this.Close();
          }


dosya.Close();
foreach (Controlnesneinthis.Controls)
if (nesneisTextBox)
                (nesneasTextBox).Clear();

       }
    }
    }
```

Form 4;

```
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;

namespace Flight_Code_Generator_2
{
publicpartialclassForm4 :Form
    {
public Form4()
       {
InitializeComponent();
       }
```

```
privatevoid bunifuImageButton2_Click(object sender, EventArgs e)
        {
this.Close();
        }

privatevoid label1_Click(object sender, EventArgs e)
        {

        }

privatevoid panel2_Paint(object sender, PaintEventArgs e)
        {

        }
    }
}
```

Example of User Define Flight Path Code;

```
G94;
G01 X10.Y5.Z8.F6;
G01 X35.Y45.Z60.F15;
G01 X20.Y15.Z80.F5;
G01 X20.Y10.Z0.F40;
G01 X20.Y0.Z0.F12;
G01 X6.Y12.Z18.F15;
G01 X0.Y0.Z20.F14;
G01 X10.Y10.Z10.F30;
G01 X6.Y1.Z30.F6;
G01 X10.Y20.Z30.F18;
G01 X20.Y10.Z40.F18;
G01 X20.Y20.Z30.F18;
G01 X64.Y64.Z96.F1;
G01 X64.Y32.Z68.F11;
G01 X32.Y36.Z96.F1;
G01 X96.Y4.Z68.F9;
G01 X100.Y32.Z96.F3;
G01 X100.Y0.Z0.F9;
G01 X36.Y36.Z100.F1;
G01 X10.Y5.Z8.F6;
G01 X35.Y45.Z60.F15;
G01 X20.Y15.Z80.F5;
G01 X20.Y10.Z0.F40;
G01 X10.Y20.Z30.F18;
G01 X20.Y10.Z40.F18;
G01 X20.Y20.Z30.F18;
G01 X64.Y64.Z96.F1;
G01 X64.Y32.Z68.F11;
G01 X68.Y96.Z68.F3;
G01 X0.Y96.Z36.F9;
```

```
G01 X0.Y96.Z32.F1;
G01 X0.Y64.Z96.F11;
G01 X36.Y36.Z96.F9;
G01 X0.Y100.Z32.F11;
G01 X68.Y68.Z64.F11;
G01 X68.Y96.Z100.F9;
G01 X68.Y36.Z68.F1;
M5;
M30;
```

Example of Random Flight Path Code;

```
G94;
G01 X64.Y36.Z36.F3;
G01 X64.Y100.Z68.F3;
G01 X64.Y68.Z36.F11;
G01 X32.Y36.Z100.F1;
G01 X100.Y68.Z36.F3;
G01 X68.Y0.Z100.F9;
G01 X0.Y4.Z96.F11;
G01 X4.Y4.Z68.F1;
G01 X64.Y64.Z4.F1;
G01 X68.Y96.Z96.F1;
G01 X96.Y100.Z36.F3;
G01 X4.Y96.Z36.F9;
G01 X64.Y64.Z96.F1;
G01 X64.Y32.Z68.F11;
G01 X32.Y36.Z96.F1;
G01 X96.Y4.Z68.F9;
G01 X100.Y32.Z96.F3;
G01 X100.Y0.Z0.F9;
G01 X36.Y36.Z100.F1;
G01 X0.Y4.Z68.F1;
G01 X4.Y68.Z100.F1;
G01 X68.Y96.Z100.F9;
G01 X68.Y36.Z100.F9;
G01 X100.Y100.Z96.F11;
G01 X68.Y36.Z36.F1;
G01 X100.Y68.Z36.F1;
G01 X32.Y0.Z100.F11;
G01 X68.Y64.Z0.F9;
G01 X100.Y32.Z64.F11;
G01 X100.Y32.Z36.F1;
G01 X64.Y4.Z68.F9;
G01 X0.Y64.Z36.F3;
G01 X96.Y32.Z100.F1;
G01 X36.Y32.Z36.F1;
G01 X96.Y96.Z36.F11;
G01 X36.Y0.Z68.F9;
G01 X32.Y32.Z4.F9;
```

```
G01 X36.Y96.Z64.F9;
G01 X100.Y4.Z32.F11;
G01 X96.Y68.Z36.F11;
G01 X68.Y32.Z64.F11;
G01 X100.Y68.Z4.F3;
G01 X96.Y68.Z96.F9;
G01 X68.Y96.Z68.F3;
G01 X0.Y96.Z36.F9;
G01 X0.Y96.Z32.F1;
G01 X0.Y64.Z96.F11;
G01 X36.Y36.Z96.F9;
G01 X0.Y100.Z32.F11;
G01 X68.Y68.Z64.F11;
G01 X68.Y96.Z100.F9;
G01 X68.Y36.Z68.F1;
G01 X96.Y32.Z96.F3;
G01 X4.Y64.Z100.F1;
G01 X32.Y4.Z64.F3;
G01 X4.Y4.Z96.F1;
G01 X4.Y4.Z4.F11;
G01 X96.Y96.Z32.F9;
G01 X96.Y96.Z96.F1;
G01 X4.Y64.Z4.F1;
G01 X100.Y96.Z100.F11;
G01 X64.Y64.Z100.F11;
G01 X4.Y36.Z36.F11;
G01 X100.Y36.Z64.F3;
G01 X96.Y68.Z36.F11;
G01 X64.Y68.Z96.F9;
G01 X4.Y100.Z68.F3;
G01 X36.Y0.Z64.F1;
G01 X4.Y0.Z36.F11;
G01 X36.Y100.Z96.F9;
G01 X96.Y32.Z36.F3;
G01 X4.Y96.Z96.F9;
G01 X36.Y32.Z96.F1;
G01 X68.Y32.Z64.F11;
G01 X100.Y4.Z32.F11;
G01 X4.Y4.Z0.F1;
G01 X68.Y100.Z64.F11;
G01 X68.Y0.Z36.F3;
G01 X0.Y0.Z32.F9;
G01 X68.Y96.Z100.F3;
M05;
M3;
```