UAV Based Remote Sensing For Developing Countries

A Thesis Submitted to the
Department of Electrical & Electronic Engineering
of
BRAC University
By

Abdulla Hil Kafi - 10221033
Maisun Ibn Monowar - 10221004
Raihana Shams Islam Antara - 10221051
Munir Muhammad Maruf - 11121027

Supervised By
Professor Dr. Md. Khalilur Rhaman
Associate Professor
Department of Computer Science and Engineering
School of Engineering & Computer Science
BRAC University

In partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical and Electronic Engineering
April 2015
BRAC University, Dhaka

Declaration:

We hereby declare that research work titled "UAV Based Remote Sensing" is our own work. The work has not been presented elsewhere for assessment. Where materials were used from other sources it has been properly acknowledged/ Referred.

Signature of the Supervisor

Professor Dr. Md. Khalilur Rhaman
Associate Professor
Department of Computer Science and Engineering
School of Engineering & Computer Science
BRAC University

Signature of the Authors

Abdulla Hil Kafi

Maisun Ibn Monowar

Raihana Shams Islam Antara

Munir Muhammad Maruf

TABLE OF CONTENT

Acknowledgement

We are greatly indebted to our supervisor Dr. Md. Khalilur Rhaman. Without his continuous support and guideline, this project wouldn’t have been possible.

We have to thank BRAC University for having faith in us and allow us to conduct research and experimentation.

We are also grateful to Kyushu Institute of Technology and all the faculty members of Cho & Toyoda lab, specially, Prof. Mengu Cho, Asst. Prof. Arifur R. Khan and Asst. Prof. John L. Polansky for their support. We also have to thank all the members of Cho & Toyoda lab for helping us in various ways while two of us were in Japan.

We would like to take this opportunity to thank Zubayer Al Billal khan for teaching us the basics of RC planes. Without his support it would have been extremely difficult to reach our goal in time.

We also received enormous amount of support and appreciation from Runway32, a strong community of RC plane builders. It has been a wonderful experience to witness their activities first hand.

Much of our work is credited to the members of Robotics Club of BRAC University. Not only they tolerated us in their robotics lab, they also extended their helping hand to build and fly our first RC plane.

Abstract

Remote sensing is vital tool for academics and researchers in the field of agriculture, urban planning, public health and many more. A trained eye can extract useful information where any ordinary person will see an aerial picture. However, acquiring image directly from a satellite is not always possible. Again, many do not have access to such data due their limitations, such as time or funds. Unmanned Aerial Vehicle or UAV has proved itself a reliable tool for aerial photography. Most of the aerial mapping systems utilizing UAV or a remote controlled (RC) multi-rotor aircraft are costly and requires a specialist team to operate it. Again cheaper, easy to fly and widely available remote controlled (RC) multi-rotor aircraft are intended for children and hobbyists. They do not offer neither the quality nor the functionalities required for a scientific project. In such specific cases, where aerial imaging is required on low budget, only solution is to build a custom multi-rotor aircraft, tuned precisely for scientific purpose. Unfortunately, it still requires a team who are skilled builders and experience RC pilot. Our project aims to bridge the gap that exists. Our project goal is to provide a platform for remote sensing based on UAV for developing countries such as Bangladesh where budget is not abundant. We have devised a cost efficient way to build semi-automated UAVs which is easy to operate for a novice RC pilot, having the necessary functionality a researcher is likely to need to complete his experimentation or data collection. Our design utilizes local components to build the main frame for a quad copter to keep the cost down, ArduPilot for high precision control to ensure the safety of surrounding people and software interface which delivers cutting edge performance almost free of cost. For control and communication link with our UAV, we used 2.54 GHz band, which is free of cost and produces satisfactory result for operation within line of sight. To create a basic image processing platform, we have used Open CV, an open sourced, community maintained computer vision library. Altogether we have been able to create a cost effective and efficient system which can be easily adapted to any form of aerial imaging where both time and resources are limited.

CHAPTER I: INTRODUCTION

1.1 Motivation

The Field Remote sensing refers to the activities of observing activities or objects from far away. Since it can be done remotely we don't need a physical contact. On the other hand Micro Aerial Vehicle, a part of emerging Unmanned Aerial Vehicle (UAV) is a lucrative platform for military, commercial, scientific and mapping application activities because of its variation of autonomy [1]. This aerial vehicle can be restricted by size, shape, controlled by an operator or full autonomous. The main features of UAV based remote sensing is it can be used in hazardous environment for aerial photography or micro drones that cannot be accessed through ground vehicle. This UAV based remote sensing provides us platforms like a.) Airplane like fixed wing models b.) Flapping wind insect like models and c.) Helicopter like rotary wing model. A quad copter or quad rotor is an aircraft that is classified as rotorcraft because of its derivation of lift from its four rotors. They can make sharp turns and perform indoor operations as well [2]. However Quad copter is also an emerging fields of robotics in Bangladesh for last few years. Due to its relatively simple model it has importance on the field of aerial photography.
The problem associated with this UAV, it has a very high power consumption rate and vibration. Whatever its job to fly/aerial photography/remote sensing is done from its battery. The heavier the battery is there will be more power and less autonomy of the robot because of the heavy weight. If we mount camera or other sensors for taking environmental data that will make it even heavier. With a fully charged batteries it can able to fly around twenty to thirty minutes. So it needs to be charged very often using human contact. Besides disadvantages, it still can be an option of remote sensing for a country like Bangladesh via UAV compared to satellite because of its huge cost of deployment and maintenance.

1.2 Literature Studies

An Unmanned Aerial Vehicle (UAV) which does not have an onboard pilot or passenger and can be fully autonomous or remotely controlled drones. The history of UAV goes back to late seventeenth century as – The Montgolfier brothers of France experienced with balloons in 1783, another popular opinion was by Austrian balloons for war fighting occurred on August 22, 1849 against Italy. Later used in American earlier civil war. Japan and USA to set fire on the opposing sides. That later developed by USA and Germans for mostly military purposes [3].

1.2.1 Fixed Wing UAV- Glider:

Among the platform fixed wing and Rotor Wing we first choose fixed wing gliding ability. Our initial consideration was mostly flight time and payload. After extensive study on the subject, we found a model for Glider named 'Big Brother' which can be launched by hand [4]. It has the gliding capacity but it cannot be hovered in a particular place for examining more closely. For the body material we have used PVC sheet which can be easily bend and shaped and can provide necessary stability and strength. We have also used two light weight steel kind of rod to provide the balance and center of gravity. Most fascinating thing about PVC sheet it can be curved, cut or given any shape by using heat and precise bending and yet the structure stays surprisingly strong. Even if some accident happened it can be fixed within an hour or two. It had a pusher aircraft configuration means we have to place the propeller behind the engine. We have used 1800KV BLDC motor for the thrust but it was not enough and the structure seemed heavy so we have decided to change the structure.

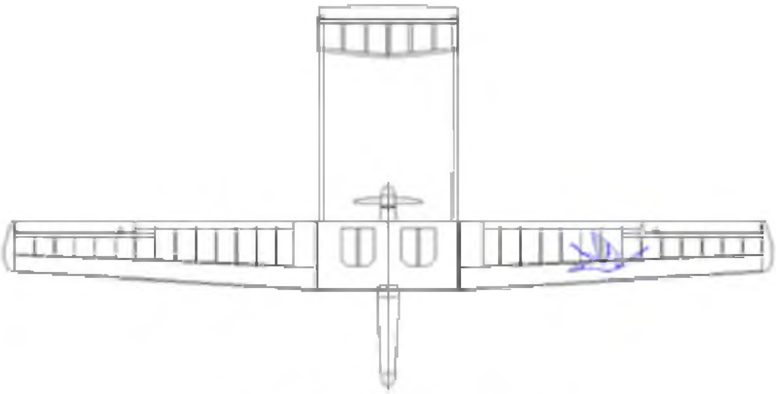


FIGURE 1.2.1 (A) CAD FILE OF GLIDER

FIGURE 1.2.1 (B) TESTING OF BLDC MOTOR

FIGURE 1.2.1 (C) SUB-ASSEMBLY OF GLIDER

1.2.2. Fixed Wing – Extra 300S:

Another attempt to build an aircraft with glider qualities and limited controlling of rudder, aileron and rudder by using servos mounting on its body This model was designed by German award winning aerobatic pilot and built by Extra Flugzeugbau [5]. We use 180 Degree mini servos to control its flight. The concept of using elevators is to make plane climb or dive; aileron to roll left or right; rudder to turn left or right; throttles to make go faster or slower. The initial body was built by cork sheet then given aero foiled shape by rubbing its body. Finally it was covered by some sort of special paper to make its body robust. Fixing the center of gravity or CG using two aluminum light weight rod was one of the challenging part of it, since we have to insert it within cork sheet. This model needs a lot of aerodynamic study. However Extra 300S is not a beginner airplane in assembly, setup or flying since it used all the key factors a genuine aircraft use. Besides its complexity it would be an excellent choice for experienced large scale modular. This model helps us to understand different variables of flight. Unfortunately For not having enough experience we need to redundant this model.

FIGURE 1.2.2 (A) HORIZONTAL COMPONENT FOR EXTRA 300S

FIGURE 1.2.2 (B) VERTICAL PART FOR EXTRA 300S

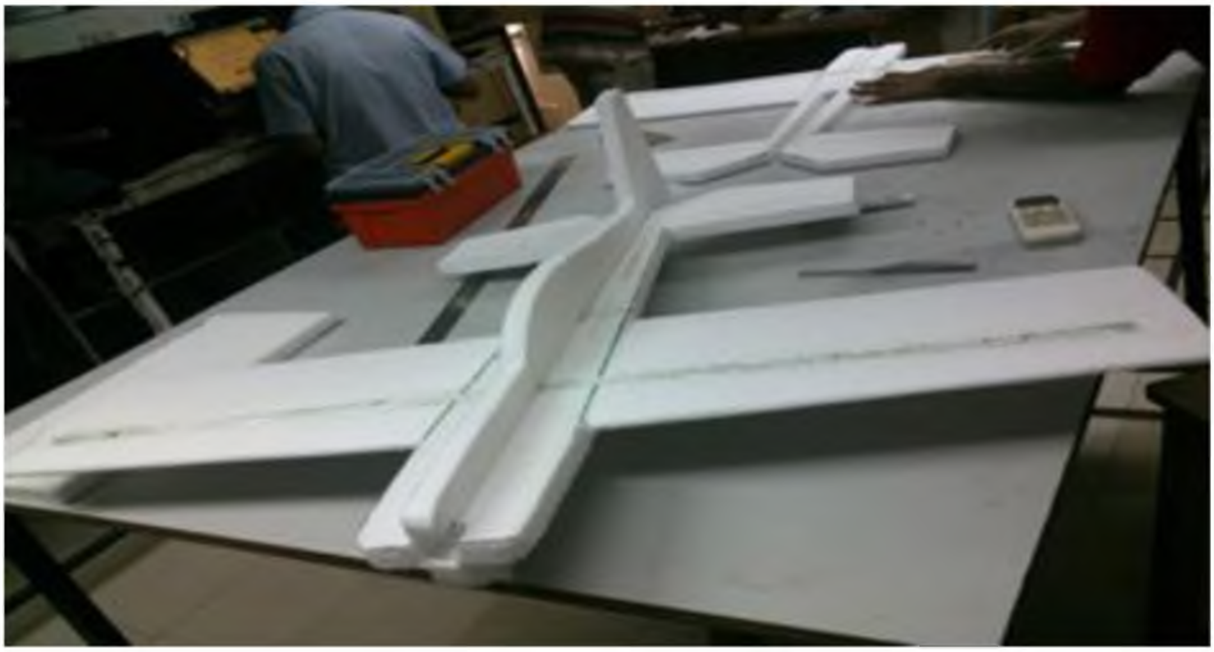


FIGURE 1.2.2 (C) ASSEMBLED EXTRA 300S

1.2.3 Fixed Wing- Raptor:

Next experiment we did with another fixed wing model inspired by F-22 US fighting aircraft raptor. Because of its full aerobatic capability, high roll rate, inverted flight, looping, hammerhead and low wind landing. We use very light weight material as foam board, tapes and aluminum rods to make the body. By default for the aerodynamic design it has shock resistance, stability of nose up and down and can be hand thrown from ground. We installed ESC, radio receiver bind with ground radio transmitter, EMAX Brushless 2815/09 Motor and 11*7 inch propeller to fly it. After packaging the structure was stable enough to hold the weight, the thrust was maximum enough to fly and aerodynamic enough to fly. Finally we have succeeded to fly a handmade RC plane made by using local instrument that can fly even in light wet weather.

Figure 1.2.3 (a) Raptor (Blue Bird).

1.2.4 Rotor Wing– Quad X:

Firstly we choose an X model to be precise X-600 model because of its Rotor Efficiency and Rotor Load to get gest wind resistance compared to '+' model. Though these two models are pretty same except for some motor position mixing and forward direction. We have used Ardupilot (also known as APM) an open source UAV development platform, which is able to control autonomous multi-rotor, helicopter or ground hover. The main frame was made of carbon fiber to make it lightweight. EMAX BL2815/09 motors were mounted on the edge of four arms. There was a separate compartment for putting the battery and on board camera, Ardupilot board, GPS module, Radio receiver. ESCs are mounted by zip tie with its arm. After doing all the calibrations from ground station we when it was ready to fly we have found that the structure seems a bit heavy. Besides that the motors have huge vibration though we have used vibration damper and placed them as far as possible from the Ardupilot board. Ardupilot has internal compass that gets seriously affected by vibration. While flying this board tries to keep the copter in the maximum stable position. When it gets the vibration it makes some calculation error and found to misbehave. After trying to fly for several times and breaking propellers we finally get to know about the Ardupilot board's miss behavior due to vibration and motors magnetic field. Because of that Ardupilot 2.6 was realized just after Ardupilot 2.5 [6].

Figure 1.2.4 (a) X Type QUAD COPTER

1.2.5 Rotor Wing- Quad H:

Quad copter H frame is becoming popular day by day because it offers the ideal aerial video platform. Traditional quad copters based on 'X' or '+' frames tend to block the view of the onboard camera with its arms. In 'H' frame the arms are being positioned as perpendicular to the main frame, what can offer maximum sight for the camera. The H frame we built was bigger compared to X frame and most fascinating thing about this frame is it is made by completely local material PVC pipes. A frame can be made within three hours of effort. H frames has bigger Arms or booms and PVC pipe itself acts as vibration damper. By using a covering tape the vibration gets minimized and hardly reachable to the CG of the frame. Ardupilot 2.5 can do its calibration and calculation pretty smoothly compared to previous X frame. Unlike X frame we use props of 11*4.5 inch instead of 11*7 Inch. This model can fly pretty smoothly. However H has some disadvantages as its low torsional (twisting/ turning) rigidity. "This means that the front wing wobbles against the back, and twists when yawing. This is particularly bad for yawing, as the twisted wings counteract yawing forces.". Because of landing position arms can be broken as well. Though that can be repaired easily.

Figure 1.2.5 (a) Flight capable 'H' frame quad copter

1.3 Introduction

Unmanned Aerial Vehicle (UAV) which is often referred as drones is an emerging field of robotics that can be remotely controlled. This technology has been developed and implemented by military since World War I (WWI) for causing destruction to the enemy force and land. This platform is being open for everyone after NASA's few UAV projects that does not have direct connection with military. Unmanned aerial photography was developed due to the improvement of cameras that enables us to do remote sensing or observing an object from a distance. This platform is cheap, fully or partially autonomous, can be produced locally and be accessed by researcher, hobbyist etc. Accessibility is one of the basic features of UAV based remote sensing. For effective remote sensing and analyzing the data to determine the change pattern in landscape geophysics, hydrology and vegetation information's are required. UAV platform can be customized as they come both as ready to fly configuration and basic frame configuration to customize own design. The frames are usually made of carbon fiber to reduce the weight and the price is comparatively high for a developing country. However readymade frames has some serious balancing problem that can be measured later by making sure all the force is working through the center of gravity (CG). Every minor details like balancing propeller, installing same rpm motor with same response is very important to hold in on air. Another problem is damping the vibration caused by the motor, if vibration reaches the onboard decision making board then the gyroscope and compass acts strangely that causes the quad copter to fall on the ground. It happens because the job of flight controlled board is to balance and stabilize the whole system. It tries to counter the imbalance part by changing position of the other side of it. While getting constant vibration it cannot decide the movement or action it should be taking, ends in falling on the ground. The magnetic field caused by the locally available Chinese motors make interference with the board if the board is close to it. Most of the prebuilt X or H frames are smaller in size and affected by all the factors. For gathering experience we used a pre-built X-600 carbon fiber frame at first but that had to be abandoned because of the problems we were experiencing. Finally we decided utilize local components and make out own frame which is significantly cheaper compared to pre-built frames. Previous experience with Radio Controlled plane, Quad copter frames inspires us to do it. In H frame we make the arms long enough so that vibration and magnetic field interference does not reach the decision making board. Besides it gives us advantages over previous X or + frames because for remote sensing we need an onboard camera that will be transmitting the video or images. H frames tend to give more space for the observation because of its parallel position of its arms. While flying in the urban area we have found slight miscalculation in GPS module that can be solved by flying on a tall building or open field where we GPS module can get the signals better. After acquiring the videos from onboard CCD camera having 2.54 Gigahertz analog video transmitter we used open source platform Open CV which library is maintained by volunteers around the world. Any code

Figure 1.3 Experimented UAV Systems

written in Open CV can be converted in python or java depending on the necessity. Our first goal was to measure crop health based on color detecting. Health of a batch of crop can be determined using appearance of crop in different spectrum. Besides we wanted to create super resolution map using panoramic photo stitching techniques by capturing a series of image that can be turned into a composite image using image processing software. That image has higher resolution compared to any single image taken by the camera. Finally, River path documentation using simple edge detection algorithm. For monitoring a river bank UAV needs to fly according to its GPS coordinated, later can be run by a custom software following Canny Edge Detection algorithm. Lab View software is used for friendly user interface instead of giving them complex graphical view. The most fascinating part of the project is it has its flexibility, from building structures to adopting processing platform.

CHAPTER II: UAV ENGINEERING

2.1 Aeronautical Engineering

Aerospace engineering, an interdisciplinary part of engineering consist of researching aerodynamics, designing aircraft and implementation, testing, retesting of air and space vehicle. Inspired by the flying machines from Leonardo Da Vinci's skeleton sketches to Wright Brothers flight and later progressive modern jets, It is now considered as two interchangeable disciplines Aeronautical Engineering - related to planet earth and aircraft & astronautically Engineering - Spacecraft, Astronomy, Astrophysics etc.

Aeronautics merged from the ancient Greek words ἀήρ *āēr referred* as Navigation of air. It focuses the whole process of aviation, air transport management, mathematics, thermodynamics, material science, production engineering, aerodynamics, CAD (Computer Aided Design), fluid mechanics, aircraft propulsion and stability related with Earths environment. According to the British Aeronautical Society This term has been expanded to include technology, business and aviation. Sometimes aviation is being used as interchangeable word of aeronautical engineering despite of its real meaning.

Though it emerges from many overlapping disciplines of science, technology and art as well, it has been considered as the application of applied mechanical engineering and aerodynamics.

2.1.1 The atmosphere:

Air is a mixture of nitrogen, oxygen and mixture of different gases. Since an aircraft operates mostly in the air it's important to understand the correlations between aircraft and the atmosphere. Properties of air and its effect on aircraft need to be understood

Pressure: Atmospheric pressure varies with altitude. It has is inversely proportional with altitude. The higher the object compared to sea level the pressure is lower. The ambient atmospheric pressure is important to determine flight properties. Aircraft movement both vertically and horizontally through the air is sampled at two or more location outside the craft by control system. Then the result is being converted to give electrical instructions.

Density: It is proportional to pressure and inversely proportional to temperature. Aerodynamic force is directly related with the air density flowing past the body.

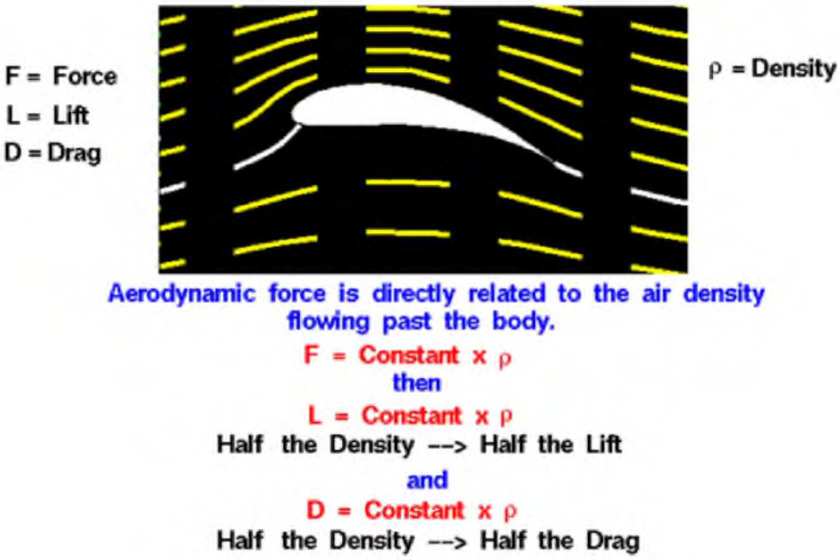


Figure 2.1.1 Effect of Density on Aircraft

Humidity: it refers to the amount of water vapor in the atmosphere that varies with temperature. If the humidity goes up air pressure for a certain volume goes down causing fewer air molecules and less lift while pushing through the air mass. Aircraft engines are designed to operate in cold, dry air. Humid air tend to have less air oxygen molecules resulting in less engine combustion and less thrust. It has its effect on both aircrafts wings and engines.

Temperature and Viscosity: Viscosity is defined as internal friction of a fluid caused by attraction of molecules results in less tendency to flow. The viscosity of air is important in the close region of the aircraft surface called boundary layer. Air viscosity increases with temperature unlike liquid.

2.1.2 Material Science & Properties:

Material Type:

- Metal Alloy: Used for basic structure required for high tension and pressure.
- Titanium: Used for high temperature application as blackbird, Concorde
- Polymer: used for lower strength than metal.
- Ceramics: used basically for heat protection and glass.
- Composite Material: composite material is being defined as fiber fibrous reinforcements bonded together with a matrix material. Natural example can be found in human body structure. By mixing few materials it can exhibit some qualities none of those materials possess. The research started in laboratory out of curiosity and later carried by military in order to build expensive military aerospace and to build general people can afford. Nowadays it's mostly consists of carbon fiber and polymer for aeronautics engineering.

Metal	Alloy	E	G	σ_y	σ_{ult}	ε_{ult}	ν	ρ
		[GPa]	[GPa]	[MPa]	[MPa]	[%]	[-]	[g/cm^3]
Steel	AISI 301	193	71	965	1275	40	0.3	8.00
	AISI 4340	205	80	470	745	22	0.29	7.85
	D6AC	210	84	1724	1931	7	0.32	7.87
Aluminium	AA 2024-T3	72	27	345	483	18	0.33	2.78
	AA 7475-T761	70	27	448	517	12	0.33	2.81
Titanium	Ti6Al-4V (5)	114	44	880	950	14	0.34	4.43
Magnesium	AZ31B-H24	45	17	221	290	15	0.35	1.78

Figure 2.1.2 Typical mechanical Properties of Few materials

2.1.3 Basic principle for aerodynamics:

Newton's Conservation Law: Aerodynamic equations are being solved by either neutrons fluid dynamics law or Bernoulli's principle.

1. Conservation of Mass or Mass continuity equation stated as mass neither can be created nor destroyed.
2. Conservation of momentum stated as without external force the momentum within a flow will be proportional to the applying force that may include surface friction, body force or some external forces. Bernoulli's theorem is also been derived from this equation.
3. Conservation of energy states that energy is neither created nor destroyed within a flow but can be transformed from one to another. For example from potential to electricity.

Bernoulli's principle: Fluid flowing through a tube reaches a narrowing of the tube , the speed of the fluid passing through the tube will be increased and pressure will be decreased.

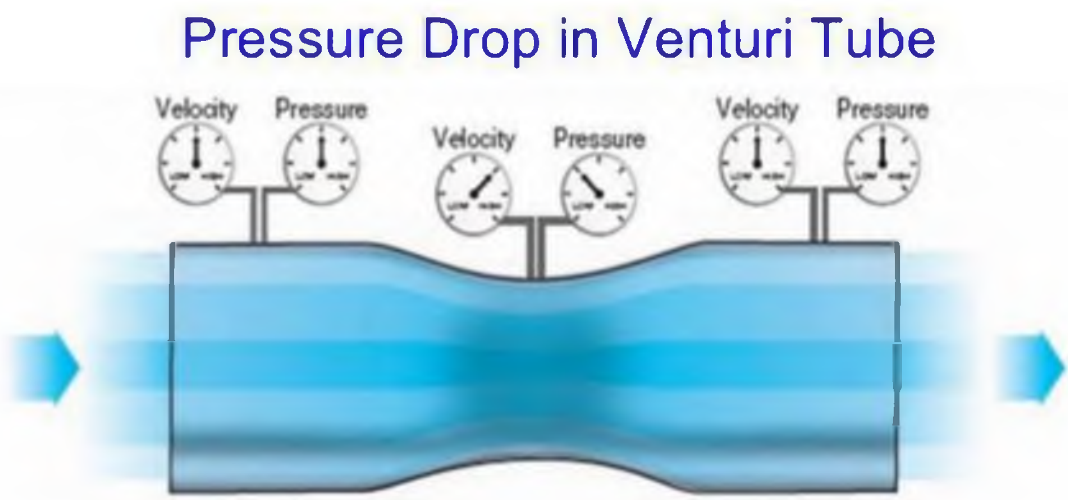


Figure 2.1.3 (a): Pressure drop in venture Tube

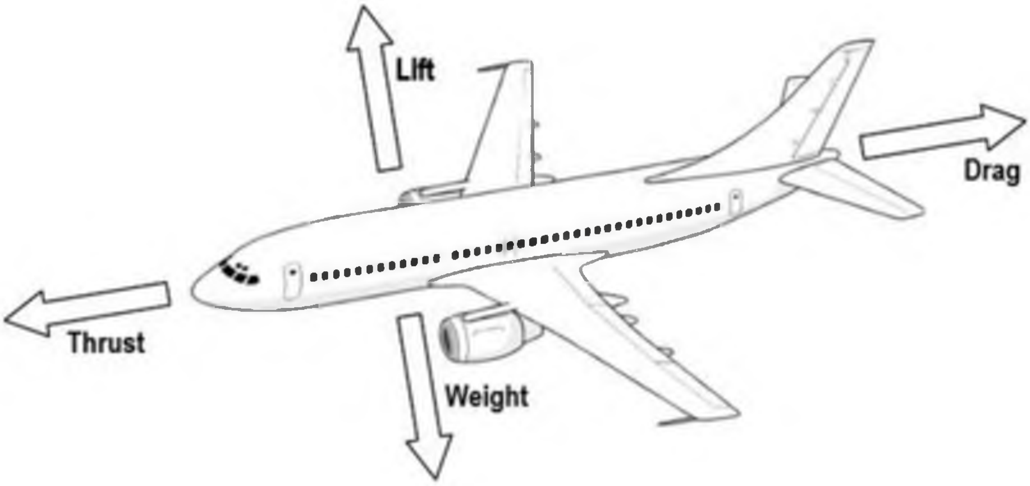


Figure 2.1.3 (b):3Four Forces of Flight:

2.1.3.1 Weight:

The force produced by gravity pulling the plane towards center of the earth. The heavier the aircraft is the bigger the weight is. If the weight is the only force working on the aircraft then it would fall straight down to the earth the moment it started to fly.

2.1.3.2 Lift:

Lift is the force create by the intersection of wings and airflow. This intersection creates a pressure difference between upper and lower surface of the object. A planes wing has its special shape named airfoil that makes sure air flows through top surface of the wing quicker than the bottom surface. The resultant leads a higher pressure below the object and lower pressure above the object. This pressure actually opposed the gravitational force and aircraft lift acts as a single point named center of pressure. The physical interpretation of the body and force interaction is quite complex and a simple change can have significant impact on the whole system.

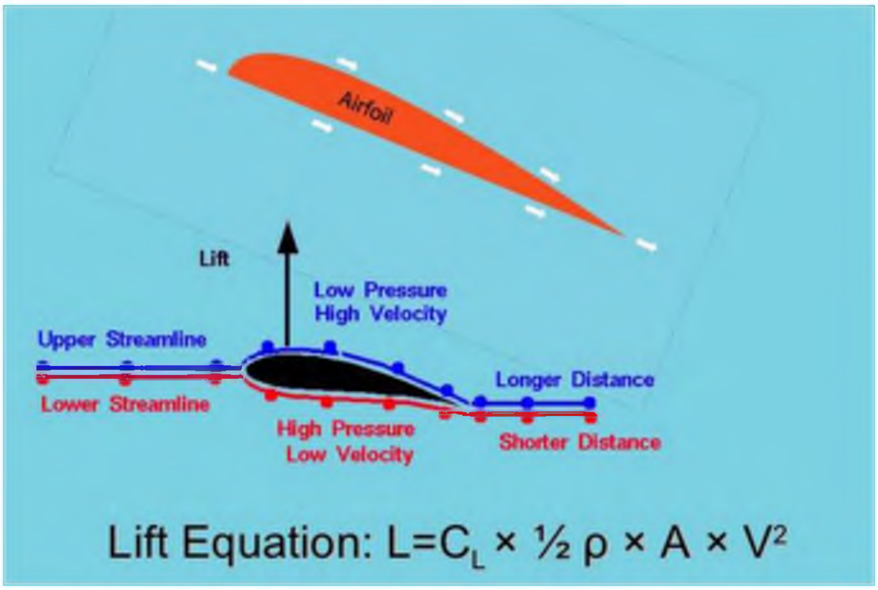


Figure 2.1.3.2: Illustration for Lift

2.1.3.3 Drag:

Drag is produced by front pressure difference as it travels through air. Drag refers to the force that resists the objects motion through the fluid and solid surface. to keep a plane moving forwards at a constant speed, another force is needed to overcome drag

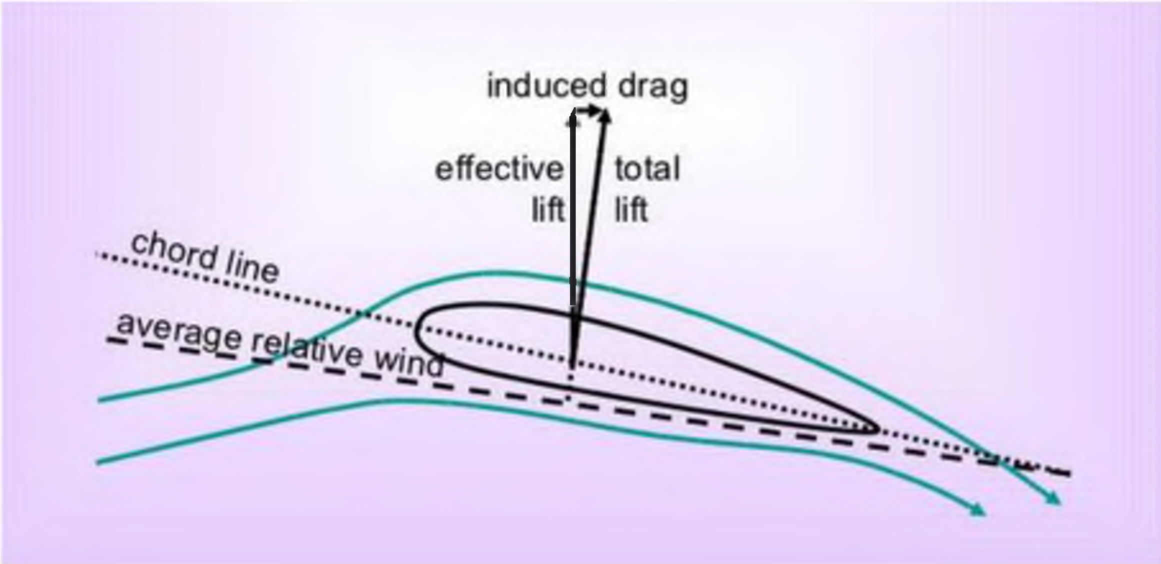


Figure 2.1.3.3 (a): Illustration of Drag

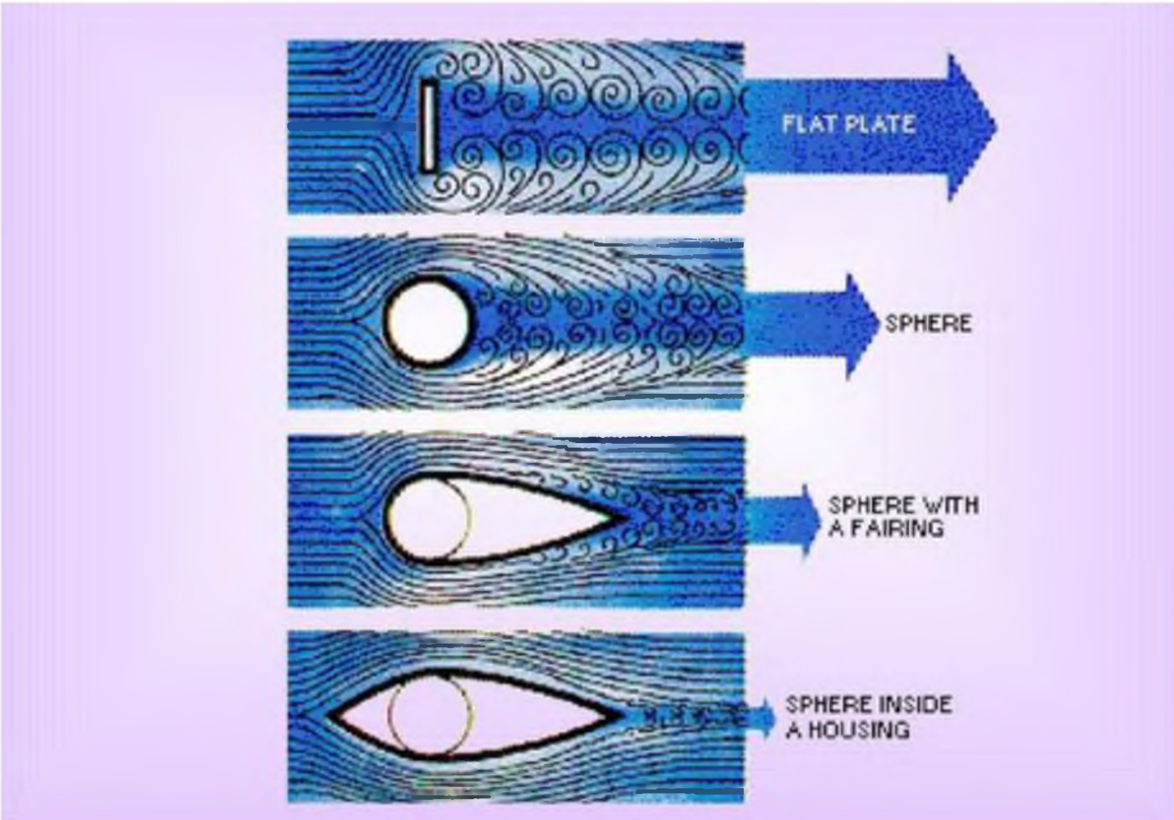


Figure 2.1.3.3 (a): Formation of drag

2.1.3.4 Thrust

Thrust is the force that causes a plane to move forward, created by mechanical means as the the plane's propeller or jet engines. Which effectively transfers energy into the flow, in the form of increased fluid momentum. Thrust is the forward reaction to this fluid momentum change. Using the same concept of lift thrust is being created by the propeller. The propeller is specially shaped like an airfoil but it uses the lift to pull the plane forward instead of pushing the plane up.

2.1.3.5 Moment:

Aerodynamic torque produced by the unbalanced force is called Moment. For a vehicle which is operating in the air needs to be balanced and all the force should work on a same point. Usually the center of gravity of a particular object. Violation may create a couple leading to the moment. It will cause the vehicle to rotate. So flight mechanics is designed to reduce this affects**.**

Aircraft flight control system has three parts a) primary control surface b) secondary control surface and c) auxiliary control surface.

2.1.4 Control System:

Primary Control System:

Elevator Control System:

An elevator is mounted on the back edge of the horizontal stabilizer of its each side of the fin in the tail. Which can operate together by moving up and down. Pushing the stick backward, the elevators go up, forward results the elevators go down. Raised elevators push down on the tail and cause the nose to pitch up. This makes the wings angle of attack high and produce more lift and drag.

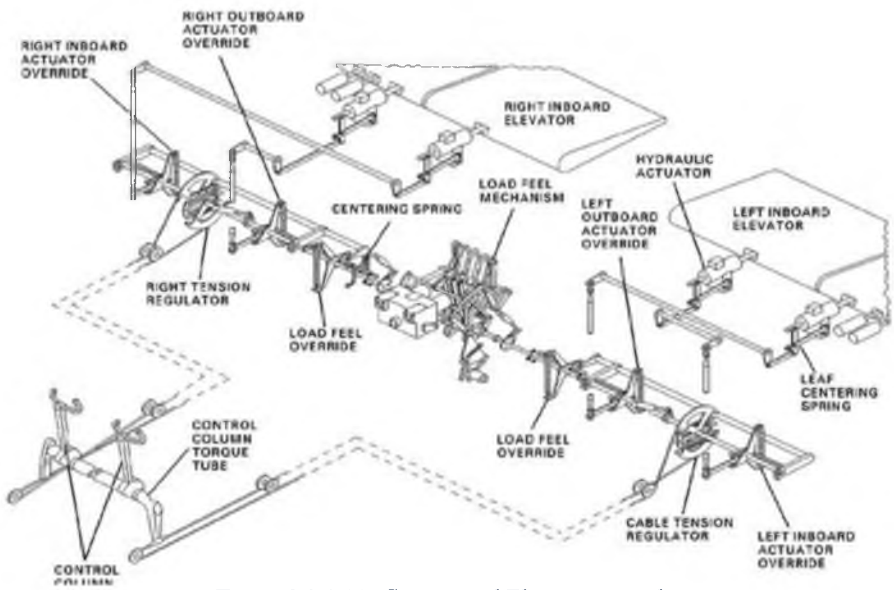


Figure 2.1.4 (a): Commercial Elevator control system

Aileron Control System:

Ailerons are mounted on the trailing edge of each wing near the wingtips, and operates in opposite directions. When the pilot moves the stick left, or turns the wheel counter-clockwise, resulting in left aileron going up and right aileron going down. A raised aileron reduces lift on that wing and a lowered one increases lift. This causes the plane to bank left and begin to turn to the left. Centering the stick returns the ailerons to neutral maintaining the bank angle.

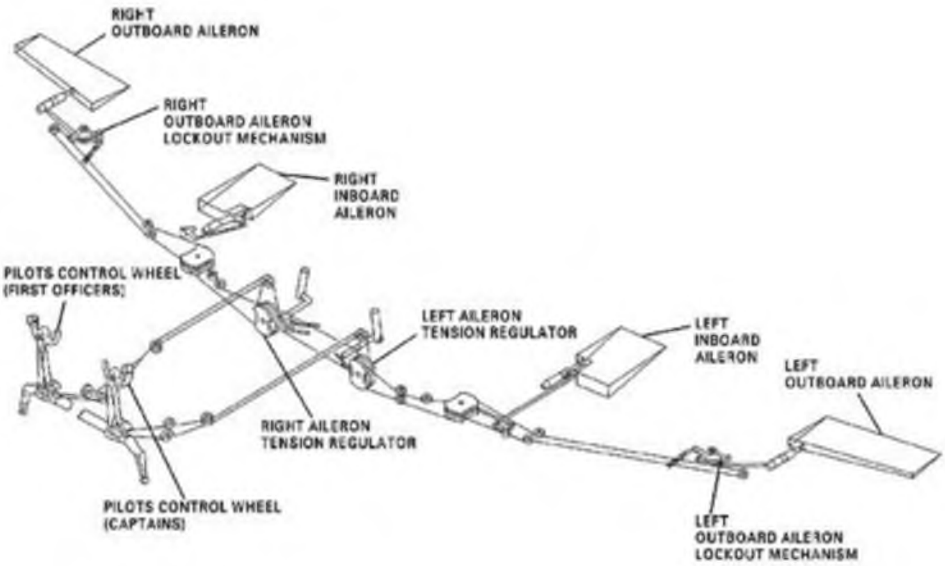


Figure 2.1.4 (b): Commercial aircraft aileron control system

Rudder Control System

The rudder is usually mounted on the back edge of the fin in the empennage. Pushing the left pedal and right pedal causes the rudder to deflects left or right. Deflecting the rudder right pushes the tail left and causes the nose to yaw right. Centering the rudder pedals returns the rudder to neutral and stops the yaw.

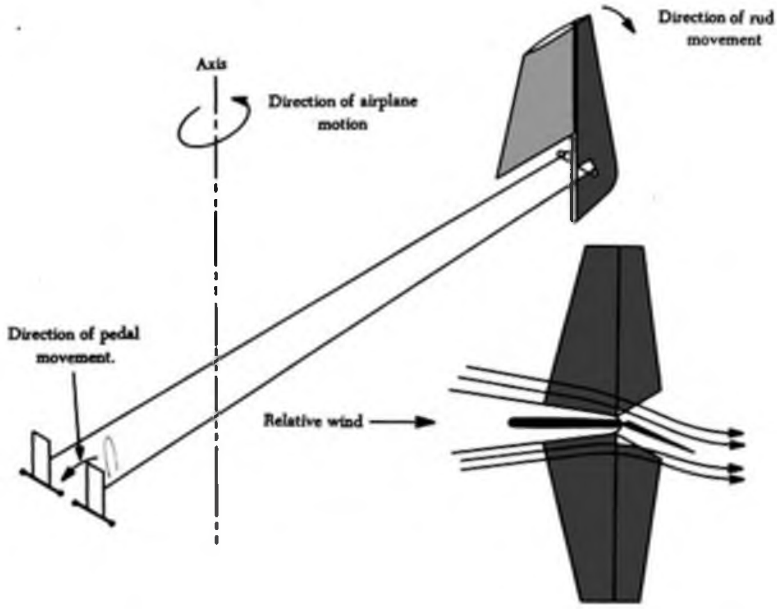


Figure 2.1.4 (c): Rudder control system

Secondary flight control:

Elevator Trim Tab System:
Elevator trim controls the force needed to maintain the aerodynamic down force on the tail. At flight a lot of trim could be required to maintain the required angle of attack. Where maintaining a nose-up attitude requires a lot of trim. The stability of the aircraft when trimmed for level flight is important due to disturbances such or turbulence will be damped over a short period of time and the aircraft will return to its level flight trimmed airspeed.

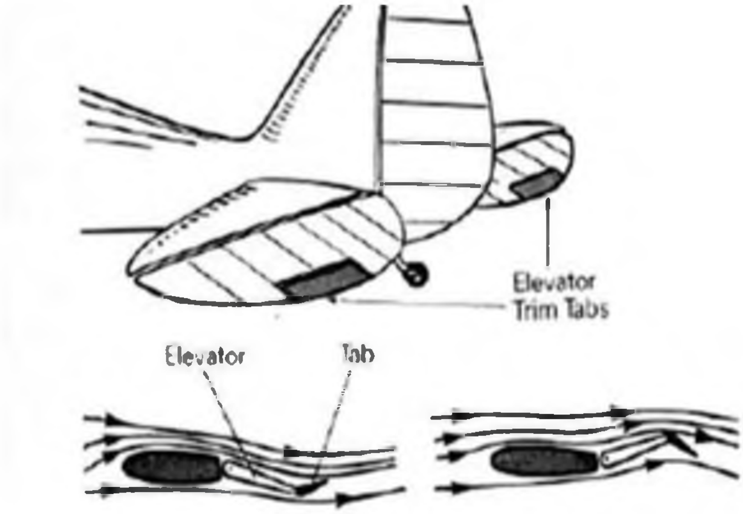


Figure 2.1.4 (d): Elevator Trim Tab System

Rudder and Aileron Trim Tab System:

Trim also applies to the elevator, rudder and ailerons. To balance the counter effect of slip stream center of gravity being to one side, can be caused by a larger weight on one side of the aircraft compared to the other. For example if one fuel tank has lot more fuel than the other one.

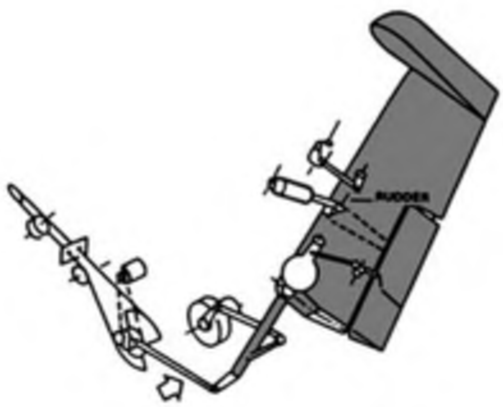


Figure 2.1.4(e): Rudder and aileron trim tab system

2.2 RC Plane Engineering

A radio controlled aircraft popularly called as RC aircraft or plane is a without pilot vehicle controlled by a remote. This remote control (RC) planes are popular among the hobbyist of all age group. In most of the cases this planes are durable and can stand standard impact. If the plane seems to have some broken parts that can be fixed by piece of tapes or epoxy gum. After some fixing the plane can be good to go again. That makes it attractive to beginner pilot and hobbyist.

A question is often being raised by people about the difference between the RC planes and drones. In technical term they both are Undammed Aerial Vehicle (UAV). Both can fly, navigate, hit target or return following its programmed path if needed. However drone referred as functional out of sight UAV can make their onboard decision without a controller. Usually drones are weapons equipped and RC planes are being flown for either hobby or collecting data through sensors. From this properties two concepts have been developed as aerial photography and law enforcement or rescue work.
This robotic aerial vehicle has come into two package one is plug and play- though one needs to connect some components as battery and secondly few components can be used as used to design custom build UAVs.

2.2.1 UAV:

We need an radio controlled UAV equipped with an onboard Fly board as Ardupilot / KK board – to control, A battery- that powers the motor through ESC, Cowl- covering motor which situates the front of the airplane, Motor- to turn the propeller, Fuselage- the main body of the aircraft containing Center of Gravitation force (CG),wings, landing gear,camera,sensors, Propeller, Radio receiver, servos, control rods etc.

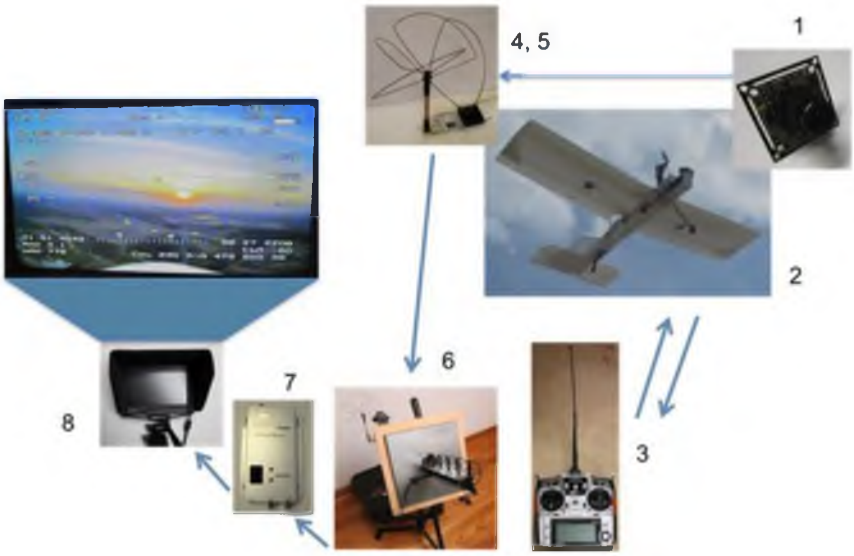


Figure 2.2.1: Communication diagram of UAV

2.2.2 Ground Station:

After building or buying a drone or RC vehicle before taking it to fly we need a ground station to control the UAV. This Ground station can be customized according to the requirement and frequency it is being used.

A ground station may contain 3D graphic interface Google 3D map view, Accurate time algorithm, Real time flight monitoring, Auto return home features, keyboard/ joystick mode, Beyond visual range, Auto/ manual takeoff and landing, simulation, upload/download flight to easy installation and efficiency.

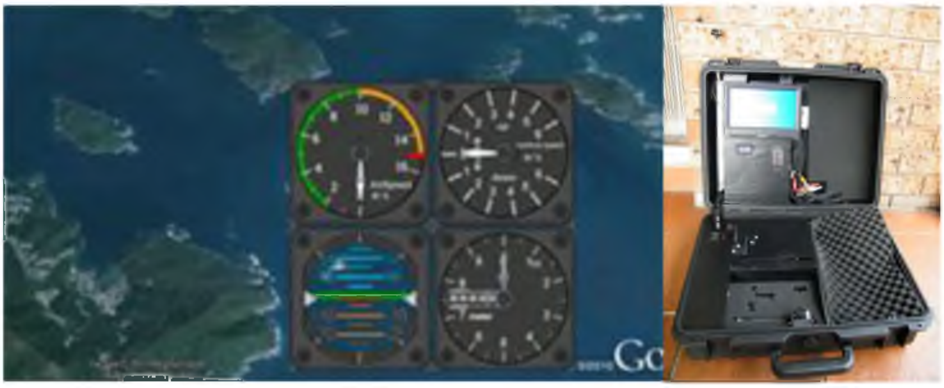


Figure 2.2.2: Typical setup for portable ground station

2.2.3 Control Link or Data link:

We need a TX- Rx module of fixed frequency to control the UAV from the ground station and another frequency if we need to transmit On board video feed or image feed. If both the transmitter and receiver works within the same frequency range will end in corrupted data and failing the mission. An UAV either RC plane or drone been operated in different ranges depending on the purpose of research, military and controlled by the aviation authorities.

Figure 2.2.3: Live feed from onboard camera

2.3 Quad copter Engineering

An Unmanned helicopter having four rotors that is lifted by four rotors or propellers is classified as quad rotor or quad copter. Among fixed wind and rotor wing aircraft it is called rotorcraft because the lift is produced from vertically positioned propeller using motors. Though it can be classified as helicopter family because of its orientation and rotor principle it is different from a traditional helicopter from many ways. It is controlled by four propeller by varying rpm, two of them are identical Clockwise and two of them are identical counter clockwise. Further manipulation can be done by adjusting torque of and thrust or lift.
Among Radio Controlled (RC) planes, helicopter and quad copter; quad copter earns its own popularity among hobbyist, amateur research projects due to its mechanical advantages. Besides four smaller rotors instead of one big one are easy to maintain and require less kinetic energy. Finally it can still be operational in challenging environment and less likely to damage property.

2.3.1 Stabilization:

An onboard gyroscope helps us to measure the tilting of quad copter in each minutes and try to balancing by same but counter tilting angle to make it stabilize. Because of this mechanism Gyroscope/ compass calibration is very important for quad copter and vibration makes the quad copter unstable in worst case it can make quad copter fall on the ground.

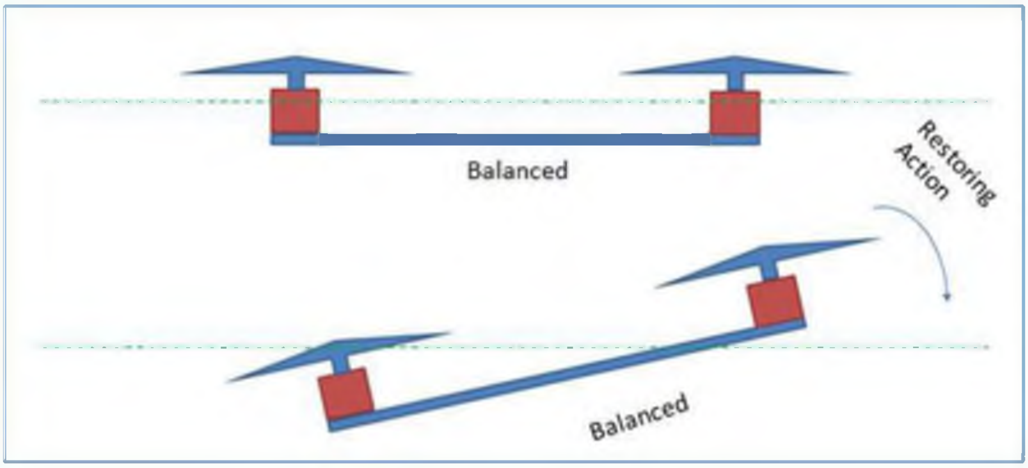


Figure 2.3.1: Basic balancing method of quad copter

2.3.2 Flight Control:

<u>2.3.2.1 Hover:</u>
Quad copter hovers its position by applying the same thrust to each of its rotors that helps us to analyze on object more closely while doing aerial photography.

Figure 2.3.2.1: Quad copter hovering position

2.3.2.2 Yaw Adjusting:

Quad copter adjust its yaw by applying more thrust to its identical same direction rotating rotors.

Figure 2.3.2.2: Yaw control

2.3.2.3 Roll/ Pitch Control:

Pitch and roll can be adjusted by applying more thrust to one particular rotor and less thrust on the other opposite geometrical rotor.

Figure 2.3.2.3: Roll/ pitch control.

2.4 Conclusion

Among All the Unmanned Aerial Vehicle (UAV) or manned vehicle the most important of all is to maintain the balance on air even in turbulent situation. Besides keeping the cost within convenient range and power consumption is another key factors in aeronautical engineering. Because of that limitations lot of initiatives have been made and failed. The research regarding UAV has stopped so many times before then started and still going on. Now a days we are looking for two qualities in aeronautics firstly gliding secondly hovering. Some of the aircraft has been designed for only those specific purpose and few of them is designed for mixed purpose. Inspired by nature for gliding properties we have designed fixed wing bird like aircraft and flapped wing insects like models. Besides we have rotary wing helicopter like models. However both of them has its own advantages and disadvantages. Fixed wing aircrafts have comparatively simple models and efficient aerodynamics structures. That may help us maximum flying time and cost effectiveness. It is appropriate for aerial survey over a large area. It may require a launcher or runway for that. On the other hand rotary wing or quad copter based aircrafts has more complex mechanical structure and power consumption. That leads us to shorter flight time around twenty to thirty minutes. It has advantages over fixed wing takeoff and landing because to can move vertically tends to take less space and functional within indoor environment. It can hover and perform agile maneuvering to inspect an object closely for more times. We tend to have more freedom regarding payloads with multi rotary aircraft. Depending on the purpose both the platform can be used because of its tradeoff on hovering and gliding an aircraft.

CHAPTER III: QUAD COPTER

The quad copter, also called RC quadrocopter falls into the category into multi rotor RC, due to them having 4 or more small rotors instead of one like the traditional helicopter. Quad copter has seen recent rise in popularity in radio controlled vertical lift platforms that is able to take off vertically, hover, and fly in all directions. In comparison with other conventional helicopters, quad-copter is more stable in flight with reduced vibration and have the simple mechanism, small size, precision control, agility and power. In this chapter we will discuss about our quad copter design and mechanism and also about the on board software systems.

3.1: Hardware

3.1.1: Frames: A Foundation for Success

First thing we need to do when building quad is absolutely good and quality frame because multi rotor depends on strong and lightweight structure for mounting components. So our first target was to make a robust frame which will be able to carry 2 kilos. There are countless frame designs of varying shapes, dimensions, and materials. One of the most common material for quad copter frame is carbon fiber which is popular for its physical properties, lightweight and strength.

3.1.1.1: X-Frame:

We chose ready-made chassis of carbon fiber as our very first quad copter for experiment. But we were not successful with this pre-build structure. It was X shape frame and the model was X- 600. It was not stable and it created lots of vibration. We were trying to reduce the vibration with vibration damper but it did not work. We were unable to fix the center of gravity and also we could not set up the Power Distribution Board (PDB) accurately on this structure. We also came to know that carbon fiber is known to block radio signals. Although this material is used most as the quad copter frame material but blocking signal is a possibility. So we moved from the uncertainty and we built our quad copter frame at home.

3.1.1.2: H-Frame:

After getting bad response from our previous frame, we started to build H but now the dimension is bigger than the previous. After researching a lot we came to know that H frame is more stable. But it was difficult to choose the material of our body. We find PVC pipe a lot cheaper than the other. So we select this material for the main frame. We use two 18 inch pipe for the middle part and four arm which is eight inch long. We attached two mid pipe with the arm using 4 T connector. We also use 4 T connector for landing gear purpose. We attached a 12 inch x 7 Inch foam with the 4 T connector. The height of the landing foam is 3 Inch. For the circuit platform we made a 4 inch x 6 inch box which is made of Balsa wood. We placed our main flight controller board, radio receiver, telemetry and the GPS module. For the mount of the Motor we bend the Pipe and

Figure 3.1.1.2: H - Frame made with PVC pipes

make a screw hole to attach the motor. This frame is very stable on fly. The motor to motor distance is 18.5 Inch. The diagonal distance of the motor is 26 Inch. It is the cheapest frame from what we used before.

3.1.2: Motors & Propellers

3.1.2.1: Motors

A multi rotor is not going anywhere deprived of good antiquated practical physics to pull it in the air. Choosing in motors plays crucial role in the achievement of a skilled setup. The variable pitch quad copter, like most small, multi rotor helicopters, utilizes brushless motors as brushless motors have higher torque and power for the same motor weight when compared to brushed motor.

Figure 3.1.2.1 (a): Brushless DC (BLDC) motor.

Motors are assigned various notations, the most consequential being the KV rating. KV does not mention to kilovolts in this case. Relatively, its motor velocity constant denoting the revolutions per minute (RPM) that a motor will turn when a 1 V potential difference is applied with zero load. It is very important as it defines a multi rotor's flight characteristics based on specification like battery voltage and takeoff weight.

There is a simple way to choose the motor by knowing the thrust of that motor Find the weight of quad and multiply that by 2, so we can hover at 1/2 throttle. Then add another 20 percent to make up for other factors. Lastly, divide that number by 4 to get how much thrust will be need from each motor.

So, if my quad weights 1200 grams, 1200 * 2 = 2400. Then 2400 * 1.2 (20% more) = 2880. Lastly, 2880 / 4 = 720. Each motor needs to produce 720 grams of thrust.

3.1.2.2: Propellers

Like other hardware used for making quad copter propeller is also important. Because propeller size directly affect power consumption and lift. There are different type of propeller such as plastic, Carbon fiber etc. we choose the plastic props among all. Large props will result in excessive motor uses and battery while too small propeller can be provided insufficient lift.

Figure 3.1.2.2: Propeller set for quad copter (1 pair clockwise, 1 pair counter clockwise)

Small props draw low current easy to stop. It is better to use large props and low power motor for larger quad copter which carry payloads. Larger quad copter with larger props will work better as they have rotational momentum which will maintain craft stability.

We use 10x4.5 inch four props. Among them two are clockwise props and two counter clockwise. These propellers are described with length and pitch. As we used 10x4.5 inch props where 10 is the length and 4.5 is pitch.

3.1.3: Speed Control & Batteries

3.1.3.1: Electronic speed controllers (ESC):

Electronic speed controller (ESC) translates signal to electrical supply. On a quad copter, every motor gets its own ESC and Every ESC connects with the flight controller. By computing the inputs, the flight controller directs each ESC to adjust its speed in order for the quad to perform them. Each ESC has a processor, firmware and other electronics that manage this task by rapidly switching the power to the motor on and off. Here we use Brushless ESCs to control brushless motors. Before choosing ESC we had to check the maximum amperage an ESC can handle because it needs to be greater than the motor and prop combination will draw. If the amperage rating is low or not high enough then it could overheat and die.

To figure out which ESC will suit for our motor at first we found out our motor's amperage. We also got the amperage information supplied by the manufacturer. But for confirmation that the amperage for the motor running at 100%, we calculated it by ourselves.
To do this, we took the number of Watts of the motor is rated at and divide it by the voltage of our battery pack.

$$\text{Maximum motor amperage} = \text{Motor watts/Battery voltage}$$
$$= 380 \text{ watts}/14.8\text{volt}$$
$$= 25.67 \text{ A}$$

Our motor is rated at 380 Watts and we used a three cell Li-Po battery of 14.8 Volt maximum amperage of roughly 25.67. Theoretically having ESC with 20% extra or more Amps than maximum amperage of motor, is a good rule of thumb to ensure that ESC does not burn out and will never hold quad copter back. So that's why we chose our ESC with 60 amps rating.

Figure 3.1.3.1: Electronic Speed Controller (ESC)

ESC refresh rates vary. ESCs that accept fast refresh rates should be chosen. Not only expensive ESCs have this capability but also many inexpensive have this capability. ESCs with high refresh rates probably have firmware that is not optimized for quad copters. In this field of multi-rotors, SimonK is the supreme ruler of ESC firmware. SimonK has written firmware which is optimized for multi-rotor. Luckily we got the ESC's with SimonK' optimizations pre-loaded. To connect motors with ESCs we used 2mm gold connectors. The wires are zip-tied with the arm to keep them out of the way.

3.1.3.2: Batteries:

One more important thing is to choose a right battery pack. We should definitely searching for a lighter battery to improve that vehicle's speed and also the performance. We did ourselves a favor by not making copter carry the unnecessary, heavy weight battery pack. The industry standard is lithium-ion polymer (LiPo) batteries. LIPO is a type of rechargeable battery which is well suited for multi rotors for its lightweight, compact and also it offers high discharge rates. There are 3 factors that has to consider before taking battery packs. The very first one is voltage. We took Turnigy Nano-tech 5000mah 4S 35~70C Lipo Battery. S stands for cell and it is a 4 cell battery pack in which every single cell supplies maximum voltage of 3.7V (at full charge). The 4 cell wired in series and gives the whole pack total 14.8 volt.

Figure 3.1.3.2: 5000 mAH LiPo battery from Turnigy.

The second factor is the C ratings of this battery. It indicates the maximum rate at which the battery packs can be discharged [7]. C ratings for our battery is 35. The third important factor is capacity. Our 35C pack has a capacity of 5000 mAh. As we have the battery's milliamp (mAh) rating and C rating, we calculate the amperage of the battery. We multiply the mAh rating by the C rating, then divide the lot by 1000.

Our battery rated 5000 mAh and C ratings 35. So we have an amperage= 5000 x 35 = 175000 mA, 175000 mA /1000 = 175A.

We keep the battery pack between the top and the bottom part of the frame. We use only one battery to power up the craft and also the camera. Lipo Batteries life do not last forever. Following 80% rule is the best way to extend LiPo's life [8]. 80% rule is to avoid discharging more than 80% of the battery's listed capacity. Voltage falls more rapidly during the flight time and it is a safe way to land before 3.3 v per cell. LiPo's vary in cost and the pricier LiPos typically last for more cycles than the cheaper ones. There is another misconceptions about getting the maximum power from the copter. Plenty of people saying that if we switch up to using a bigger battery then we will get more power. It is definitely true that 4s is more efficient than 3s but on the flip side we are carrying an extra battery cell means extra dead weight. It is true that 4s battery suits for the experts not for the beginners. Generally, with smaller props needs higher voltage (4s) and there is possibility to get a more acrobatic quad. The best setup for quad is a balance of motors, props, and copter weight and battery capacity. At the beginning, we used 11x4.5 propellers which is larger and 3s battery. Finally we make our quad more efficient by using 4s battery and it also extends the flight time.

3.2 Flight Controllers

A Flight Controller (FC) is mainly a circuit board which keeps the quad copter balanced and in control by reading sensors data and user commands, and makes adjustment to the motor speed. It is brain of the quad copter. All FC have similar hardware or sensors but their different software and calculation algorithms, which results in different flight characteristics, and user interface differs them from each other. Multi-copter flight controllers have Gyroscopes (Gyro) for orientation, the barometers for holding altitudes, GPS for failsafe of auto pilot purpose and Accelerometer (ACC).There is an abundance of choice when it comes to flight controller options because there are many controller board available in the market. Here is a chart of some of the flight controller for multi-copter.

FC Name	Price	RX Modes	Baro/Compass	GPS	MicroController
APM 2.6	$240	PWM, PPM	External	Yes	8-bit, 16MHz
BrainFPV	$130	PWM, PPM, S.Bus, DSM, HoTT	Internal/External	Yes	32-bit, 168MHz
CC3D	$16	PWM, PPM, S.Bus, DSM	No	Limited	32-bit, 72MHz
Crius AIO	$48	PWM, PPM	Internal/External	Yes	8-bit, 16MHz
Flip32	$25	PWM, PPM, S.Bus	No	Limited	32-bit, 72MHz
KK2.1.5	$22	PWM, PPM	No	No	8-bit, 16MHz
Multiwii SE 2.5	$15	PWM, PPM	Internal	Yes	8-bit, 16MHz
Naza M Lite	$170	PWM, PPM, S.Bus	External	Yes	unknown
Naza M V2	$300	PWM, PPM, S.Bus	External	Yes	unknown
Naze32 Acro	$25	PWM, PPM, S.Bus	No	Limited	32-bit, 72MHz
Naze32 Full	$53	PWM, PPM, S.Bus	Yes	Yes	32-bit, 72MHz
Quanton	$66	PWM, PPM, S.Bus, DSM, HoTT	Internal/External	Yes	32-bit, 168MHz
Revo	$130	PWM, PPM, S.Bus, DSM	Internal/External	Yes	32-bit, 168MHz
Sparky	$60	PPM, S.Bus, DSM	Internal	Yes	32-bit, 72MHz

Figure 3.2: Different board and their Specification

3.2.1 KK2.0 Board

We bought a less expensive starter setup as our Flight Controller and that is KK2.0 board. Many versions of this board is available. It is one of the most cheap and popular FC. To control ESC It has built in gyros, accelerometers and also the Atmel Mega324PA 8-bit AVR RISC-based microcontroller with 32k of memory. The LCD screen makes this board extra special. One of the main advantage of this board is to everything is easily accessible and configurable from the LCD in the field, without using computer. This board is good enough for first flying and perfect for the beginners. It is also popular for its plug and play simplicity.

Figure 3.2.1: KK 2.0 Multi-Rotor Control Board

The first flight of our copter with this FC board was successful. But the disadvantages of this board is it cannot auto-level much which is very necessary for flying quad copter. It uses cheap piezo gyros instead of the more modern MEMS gyros that Arducopter uses. It is best for the beginning but this board is not best performer for a good flight.

3.2.2 Multi-Wii Pro

Multi-Wii pro is an open source software project that gives intelligence to a multi rotor RC flying platform. It can be used with different hardware boards and sensors. The stability achieved for FPV is excellent and capable of doing acrobatics. A close control loop is being used to ensure its stability and maneuverability. This is a Proportional-Integral-Derivative (PID) regulator. The Firmware of this board can be developed in Arduino IDE. Some pre built code can be found in the main website of Multi-Wii. We used the same code but it need to change according to the requirement of the system. As we are working on Quad we make changes as our quad needed. Flying in multi-wii pro was very nice but self-balancing functionality is not so accurate and it needs a lot of tuning to make it perfect. Giving flight mode is difficult in this board. It can support highest 3 kind of mode for flight.

Figure 3.2.2: Multi-Wii pro mounted on our H-Frame.

3.2.3 Ardupilot Mega (APM) 2.5

Ardupilot is an open source UAV platform that gives us freedom to flying vehicle as well as ground vehicle [9]. This platform was established by DIY Drones Community based on the arduino another open source electronics prototyping platform. Ardupilot Mega is based on the arduino mega platform which can able to control multi copters, fixed-wing aircraft, traditional helicopters, RC cars. It is the best flight controller board among the 3 controller board that we used. There are few FC board in the market which GPS features are reliable and Ardupilot Mega is one of them. Although this board is expensive, but its features make it excellent compare with its price. It has barometric pressure sensor, magnetometer, GPS, accelerometer, telemetry radio, 3DR power module and it is easy to set up. We used APM 2.5 along with CJMCU-108 GPS. The APM2.5 has a lots of sensors and they all supported natively in the code. It has also full autopilot with multiple waypoints, return to launch which features it great ground control station. Although many people complain that it is extremely hard to get the GPS but we did not face this problem in the open Area. Even it worked perfectly on the roof-top. The first flight

Figure 3.2.3(a): Ardupilot 2.5 with casing

with this board was not successful because of some problem with the X-frame structure and magnetic interference caused by the motors. We fixed the problem later in our H frame and have successful flight.

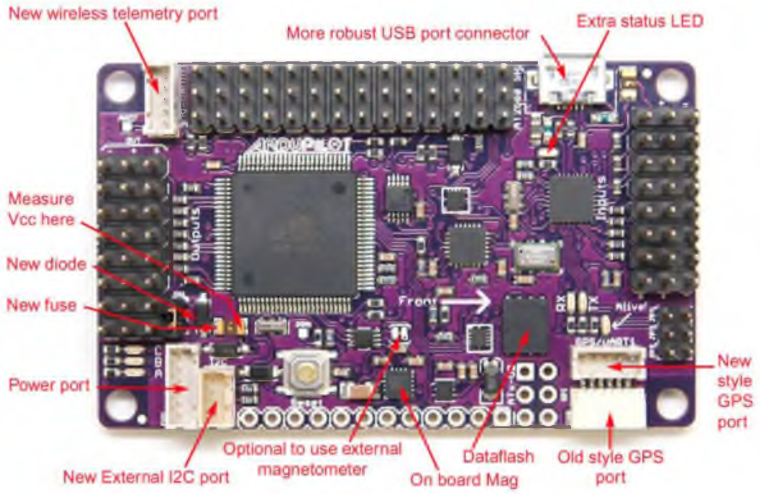


Figure 3.2.3 (b): Ardupilot APM Multi -Rotor Control Board (Pin Mode)

3.3 Camera and Video System

For transmitting the video feed we mounted an onboard camera Quanum complete FPV bundle 5.8 GHz 32 Channel video, TX & Rx, Cp antenna and camera. This package comes with a Sony CCD camera, video transmitter, receiver, polarized antennas and plug and fly configurations. We choose this FPV combo because of its popularity among amateur, professionals and comparative less price. It has the frequency range of 5645-5945 MHz with 200 mW power output and handsome open air range around 1500 meter.

Figure 3.3: Onboard Camera Combo Module

The camera is mounted on the right side of our frame. We made a platform for camera which is built with balsa wood. We fixed the camera by using double sided tape on it. We focused the camera towards ground to take the necessary images and videos that we need to experiment. For recording purpose we use the USB TV card and the video is recorded in PAL mode. The frame width 720, frame height 576, data rate 7000 kbps, total bitrate 7224 kbps and frame rate

25 frames/second of the recorded video that we got. We also took the still images with this camera for our experiment.

3.4 Telemetry:

3DR radio set is an open source based platform enabling us to set up a telemetry connection between on-air module and ground modules for real time flight experience. It gives the opportunity of viewing the flight data, tuning or changing the waypoint while flying. This module is surprisingly light weight (around 4 grams without antennas) and can be found in two operating frequencies 915 MHz or 433 MHz. Most importantly it can be connected via android tablet, 3DR ratio driver or Mission planner. This USB device has plug and play option and transmission power is around 100 mW. For flying and calibration we used this device mostly with APM Mission Planner.

Figure 3.4: 3DR Telemetry Module

3.5 Flight Controller:

Quad copter can control in many different ways but the most common one is RC Transmitter. We use Fly Sky-TH9X 9CH Radio Model RC Transmitter & Receiver as our control system for this quad copter. Mainly two frequencies are used for radio transmitter; 72 MHz and 2.4 GHz. We use 2.4 GHZ frequencies which is most common. There are two main sticks for the throttle and direction control and some optional switches as well which are often used for switching between flying modes. Throttle controls the up and down of the copter by controlling and varying the speed of the rotors. Yaw; also referred as rudder rotates the copter and finally Roll or aileron tilts the quad copter left and right. Ours RC transmitter has 9 channel which indicates individual thing on the quad copter to be controlled. There is also a RC receiver which comes with the transmitter. Every transmitter is only compatible to their own receivers. We set up the receiver besides the ardupilot on the same platform.

Figure 3.5: 9 channel fly sky transmitter.

There are different flying modes for different flying purposes for a quad copter like rate mode or Acro mode, stabilization mode, AltHold mode, Loiter mode, self-level mode and so on. We use stabilize mode most of the time for quad flying. Stabilization mode is the act of keeping or balancing to keep the same tilt angle and gyro sensor is used for this task.

3.4 Onboard Software

Mission planner is an open source community supported software designed by Michael Oborne in order to provide the platform of APM ardupilot based project. This works as the controllers Ground Station for plane, copter or rover. The connection between flying object and ground station is kept using 3DR Radio Telemetry. By using this software we can launch, fly or land a flying vehicle autonomously. Before flying the copter we need to initialize its basic condition like the frame or campus or gyroscope calibration. We have the ability to disarm the flying copter by using the command or loading code. A waypoint or path have to be created for flying the copter and after security check it is ready to go.

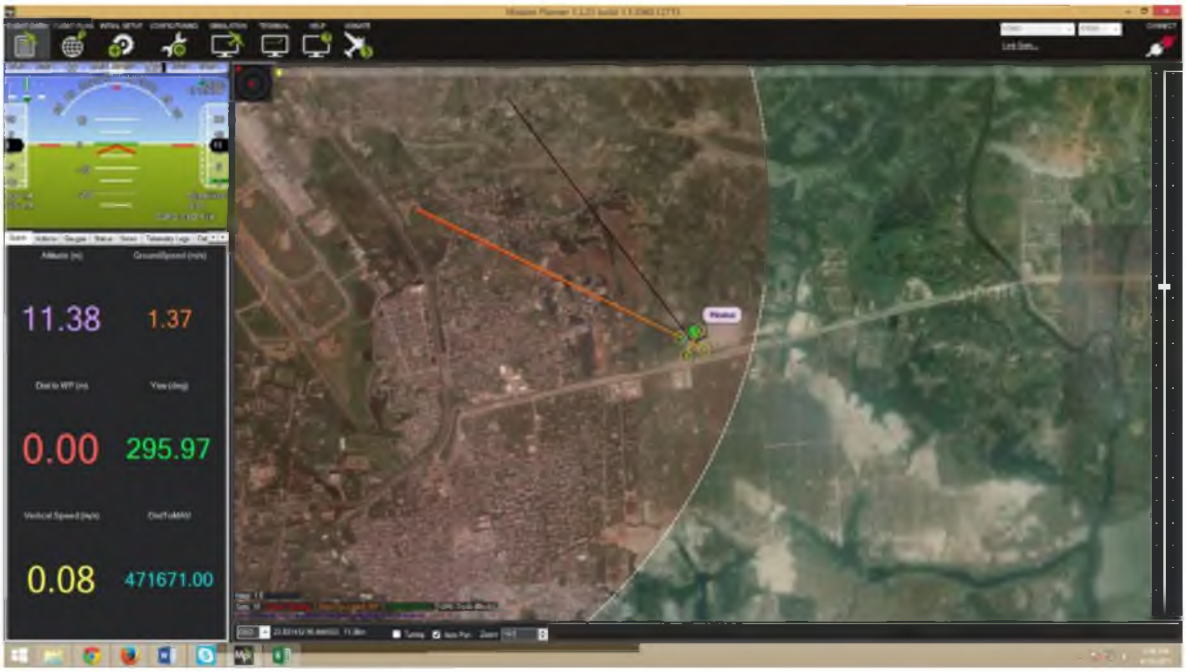


Figure 3.4: Mission Planner Software interface 1(Flying at PURBACHAL)

3.5 Ready to Fly

After successfully adjusting all the components it is now ready to fly perfectly. At first we tested our first flight at home to check its stability. In the beginning it was moving right and left frequently. Than we move to a open field for test.

Figure 3.5(a): Our H-Frame on flight.

Figure 3.5(a): Our H-Frame on flight.

Figure 3.5(b): Quad Copter Whole System

CHAPTER IV:
IMAGE PROCESSING

4.1 Objectives of Image Processing

One of the main objective of this project is to bring the best functionality of each components of remote sensing and deliver an easy to use, out of the box solution for any urgent demand based remote observation projects. To ensure most compatibility of software's and ease of customization, C++ deemed to be the perfect language for this project. Therefore most of the software's used for image processing part of this project are written in C++ or C compatible language.

In order utilize most out of pre-built code libraries, we decided to use Open CV, an open source computer vision library. Among all the available resources for academic reuse for computer vision, Open CV is by far most extensive, reliable and widely used library. Moreover, because it is open source, no financial transaction is involved to use this code. Main repository for Open CV is maintained online by volunteers. Although, original library is written in C++ language, there are also wrappers for Java and Python. Therefore any code written in Open CV, can be easily converted to be used in almost any platform.

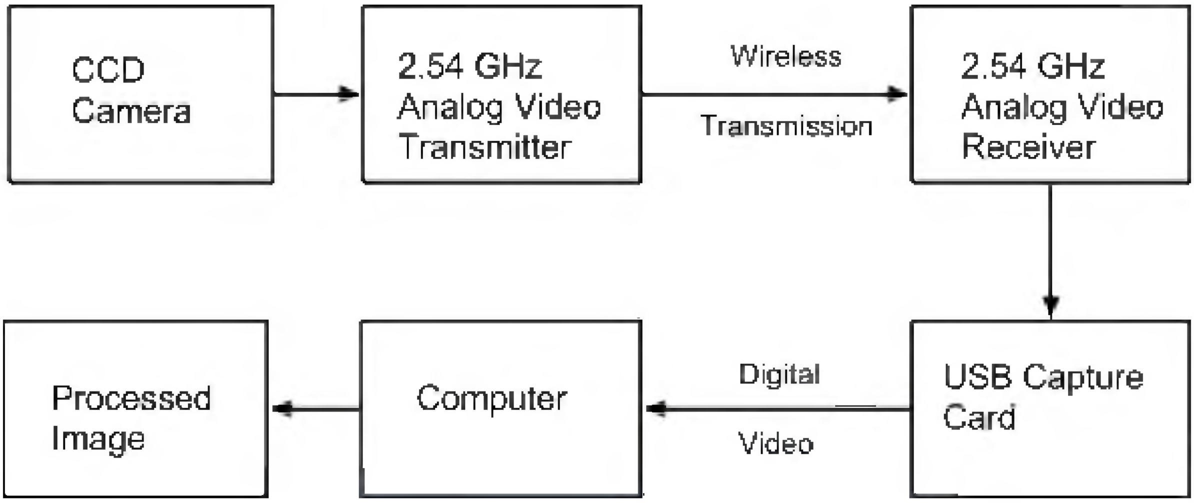
In order to demonstrate our framework and the flexibility of our system, we choose to include three basic programs which we consider most common application of aerial imaging. As because our system was developed with a color camera within visible range, all of our image processing programs were hence focused but not limited to color image processing. Our 3 sample image processing software are,

1. Crop health estimation based on color detecting.
2. Super resolution map using panoramic photo stitching techniques.
3. River path documentation using simple edge detection algorithm.

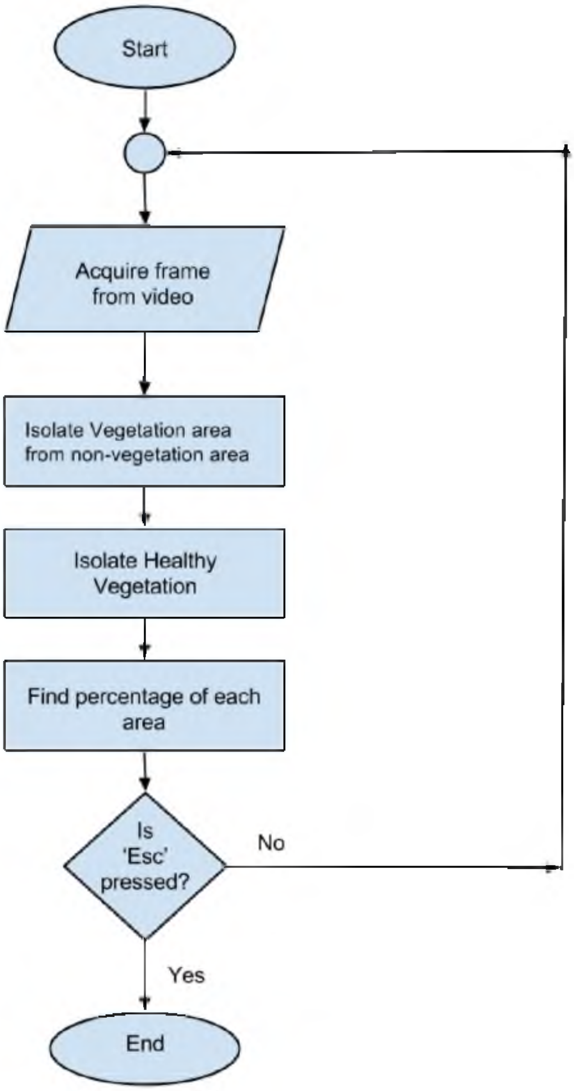
4.1.1 Crop Health Estimation

Overall health of a batch of harvest can be determined using the color and the appearance of the crops. Vital minerals such as Nitrogen and Phosphorus content of crops can be determined by analyzing appearance of the crop in different spectrum [10]. Any UAV equipped with powerful imaging devices can also determine whether or not a farm has been infected by insects, if so what sort of insect [11] and how to solve them.

Therefore, to estimate crop heath from color image received from UAV, we designed a real time system capable of giving out an estimate of crop health as seen within the camera range. The CCD camera mounted on our UAV transmit true color image using an analog transmitter. The video is then received in ground station. The analog video is converted to digital format using a portable USB TV card.



Block Diagram: Video capture and processing workflow.



Flowchart: Estimating crop health through image processing

After successful capturing of video, first vegetation area is segmented from none vegetation area in order to define Region of Interest. This is done by segmenting area based on color which is more likely to be vegetation area. To do this, a broad range of green spectrum is defined and considered as the vegetation area. Next, within the region of interest, we calculate the amount of visible area which is more likely to be representative of healthy crop. Based on the result achieved, user is notified of the results and advises on further action.

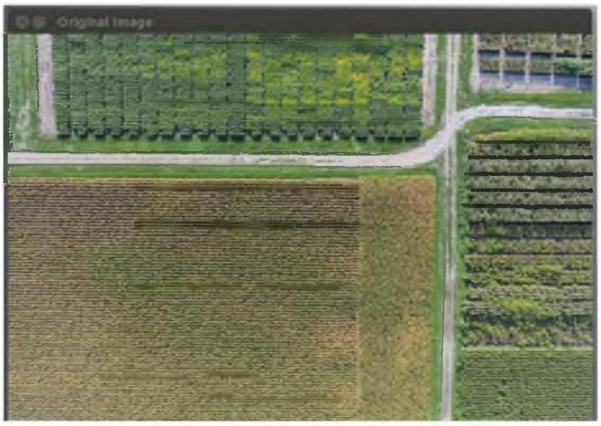


Figure 4.1.1(a): Original Image

Figure 4.1.1(b): Portion of the image found healthy

Figure 4.1.1(c): Separation of Green pixels

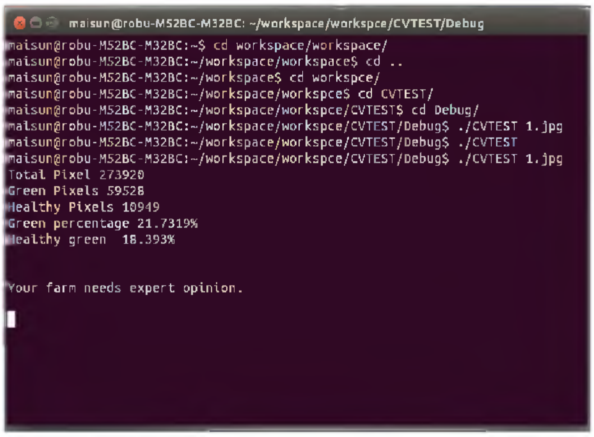
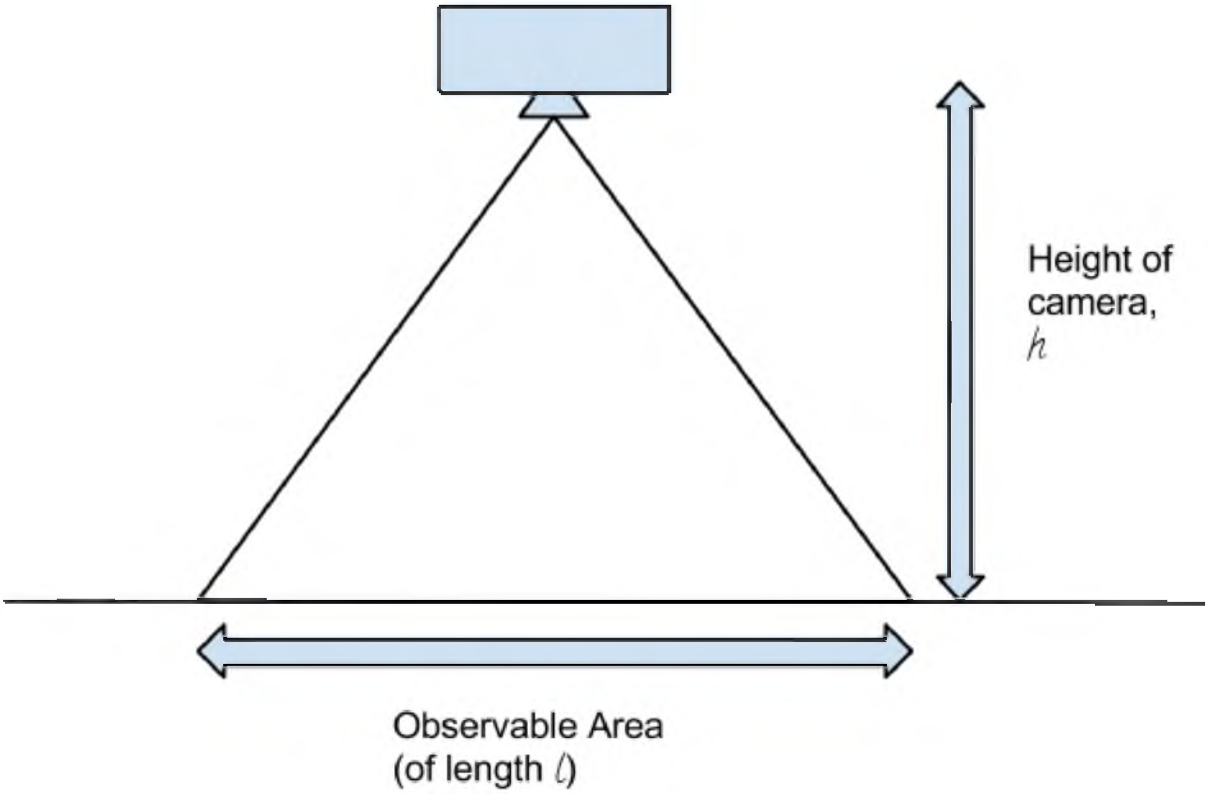


Figure 4.1.1(d): Command line output

4.1.2. Super Resolution Map

A typical application of aerial photography is to construct super resolution map from series of images [12] captured during UAV flight. Utilizing numerous photographs captured using a professional image processing software, a composite image can be made. The composite image is of much higher resolution compared to a single photo taken of the same area in a single frame. Using such techniques, high resolution image can be obtained without investing significant amount of money to upgrade hardware related to imaging.



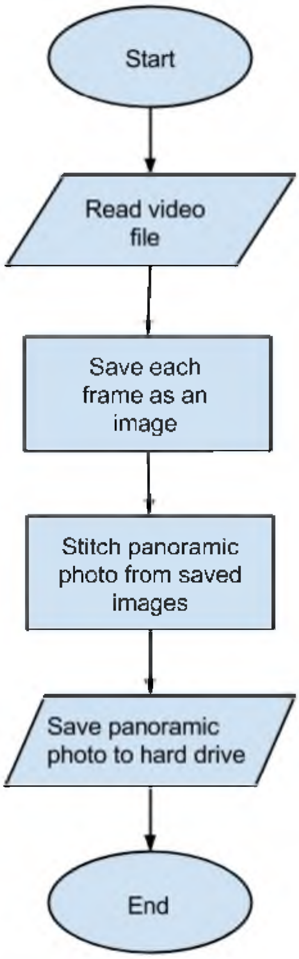
For any camera of,

Sensor resolution, r x r
Angle of view, θ
Observable area of lxl
And hovering at height, h

The resolution of the picture taken of the observable area is simply, l / r meter per pixel

As we increase observable area, length cover per pixel increase, hence deteriorating the quality of image. So any map which is created by taking a single picture high above the ground, will never have good enough resolution for practical use.

This problem can be solved by taking multiple image of the target area and stitch them together in a single high resolution picture.



Flowchart: Creating a super resolution map.

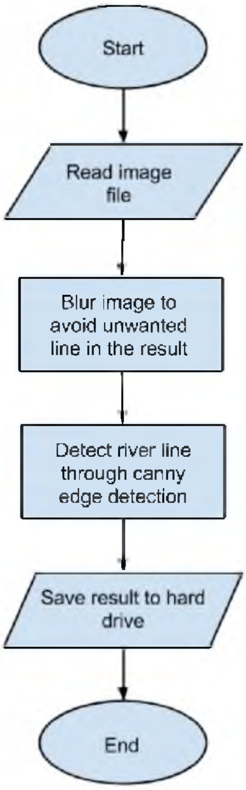
In order to achieve Super Resolution map from our UAV, we first film the area in video format. When an acceptable video of the target area has been achieved, the video is then processed by our custom made stitching program using OpenCV.

To reduce processing time required to produce a high resolution image, we skipped 40 consecutive frame. Most of our recorded videos are in 25 fps format and slow cruising speed of UAV, we found in our experimentation that skipping 40 consecutive frames improves both the photo quality and cuts short processing time.

Figure 4.1.2: In our laboratory experiment, we could successfully create a panoramic view from a video file.

4.1.3. River line mapping

Monitoring river bank erosion is vital for urban planning. Constant monitoring of river banks can provide us with the vital data such as the rate of erosion, possible course taken by the river in the future etc.



Flowchart: Mapping river course by UAV.

In order to consistently monitor a certain portion of river bank, first the UAV needs to be flown at the exact GPS coordinates required by researchers. The image taken is then run by our custom program, which is basically Canny Edge Detection algorithm. Studying the edges of the processed image enables us to study the course taken by our target river.

4.2. Lab View User Interface

Lab View software was chosen due to its versatility and wide acceptance in developers community as it can be interfaced with almost any data acquisition hardware. Programming Lab View interface is very user friendly as everything is graphical. Lab View is also equipped with advance libraries and device drivers which gives more acceptance in almost any project. Due its flexibility, any modification or update which may be necessary in the future will be easy to accommodate and manage. Therefore Lab View was chosen as the GUI of our software.

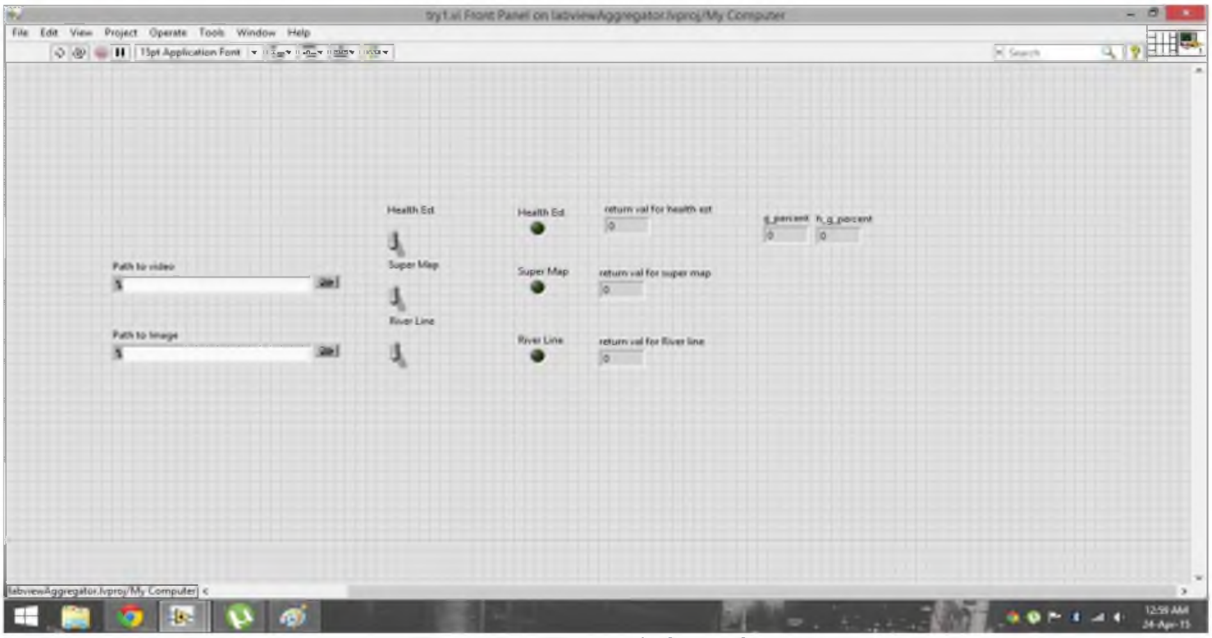


Figure4.2(a): Front end of our software.

In order to port our custom made program using Open CV, we had to compile our program as Dynamic Link Library (DLL) file and link it to Lab View using its “Call Node Library” functionality. According to the requirement of our software, a suitable control panel was created.

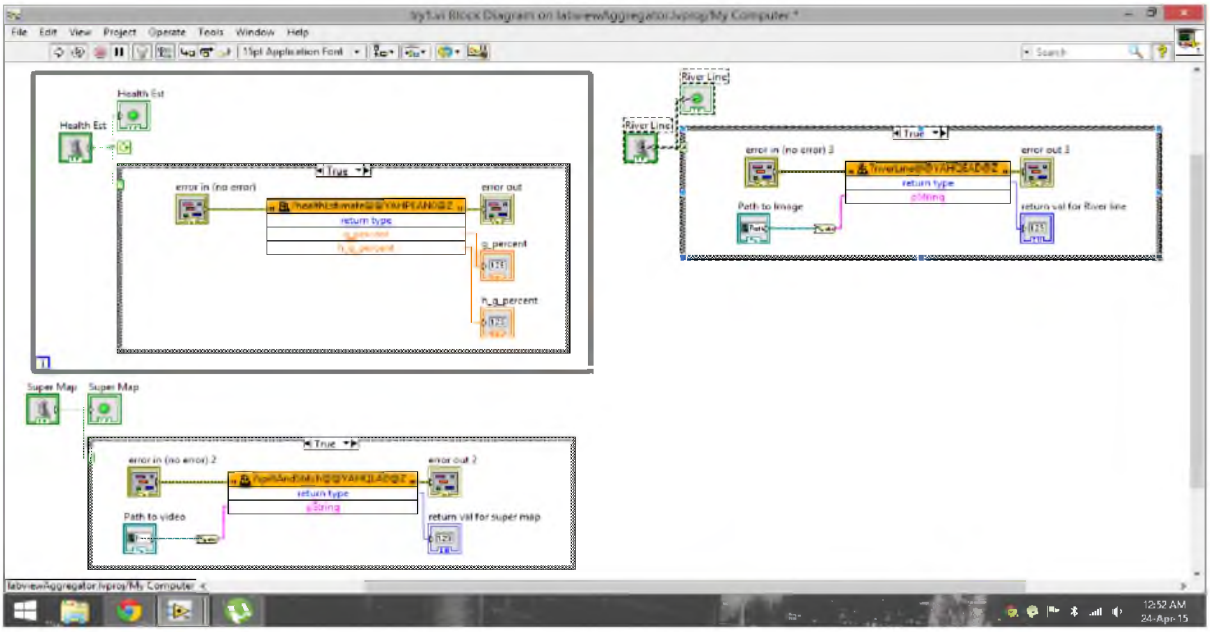


Figure4.2 (b): Function diagram or 'Control VI' of our software.

4.2.1 Creating Dynamic Link Library

Lab View is completely a graphical based programming software. Although it doesn't allow to incorporate C code, it does allow to add a Dynamic Link Library to expand functionality a user might require. Only limitation of Lab View is that, it can use only one DLL file per VI, therefore all three programs were compiled in a single DLL file.
Individual functionality of our three programs were achieved through writing a function representing each of our objective.

In first case, the function was called within a 'While' loop as it needs to continuously extract information from incoming video feed from our UAV. At the end of each loop, the DLL returns a value from which we can know the estimation of crop health. The value obtained from calculation were passed by 2 double precision pointers.

In our second case, path to video file was passed to DLL file by LabView interface. The Video is then processed and result is displayed on screen and saved on the hard drive for future reference.

In our third and final case, path to an image file was passed DLL file in a similar way as the second case. Result of the processed image is simultaneously displayed and saved on the hard drive.

To operate the software, appropriate Boolean switch must be press prior to running the entire code. For example, the obtain Crop Health estimation, one must first open the front panel vi file from Lab View. Toggle switch corresponding to "Health Estimation" must be switched on (flipped up) and rest of the switches must be off. Then pressing "Run" button will give us the result of processing.

CHAPTER V
RESULT ANALYSIS

5.1 Experiment and Result Analysis

After successfully completing the assembly we have 6 flights. Among these two flights were having GPS. The response of our quad was very stable. We successfully implemented six flight modes. But unfortunately two were tested. At the beginning we were using 3300mah 3S Battery. As our motor Power Consumption was 380W, the flight time was only six minutes. But later on we used 5000mah 4S battery, it increased the flight time to 18 Minutes. Different kind of flight data are given below. In the graph. Mainly showing the response of the roll, pitch, yaw, radio noise, altitude, throttle value are shown with respect to time

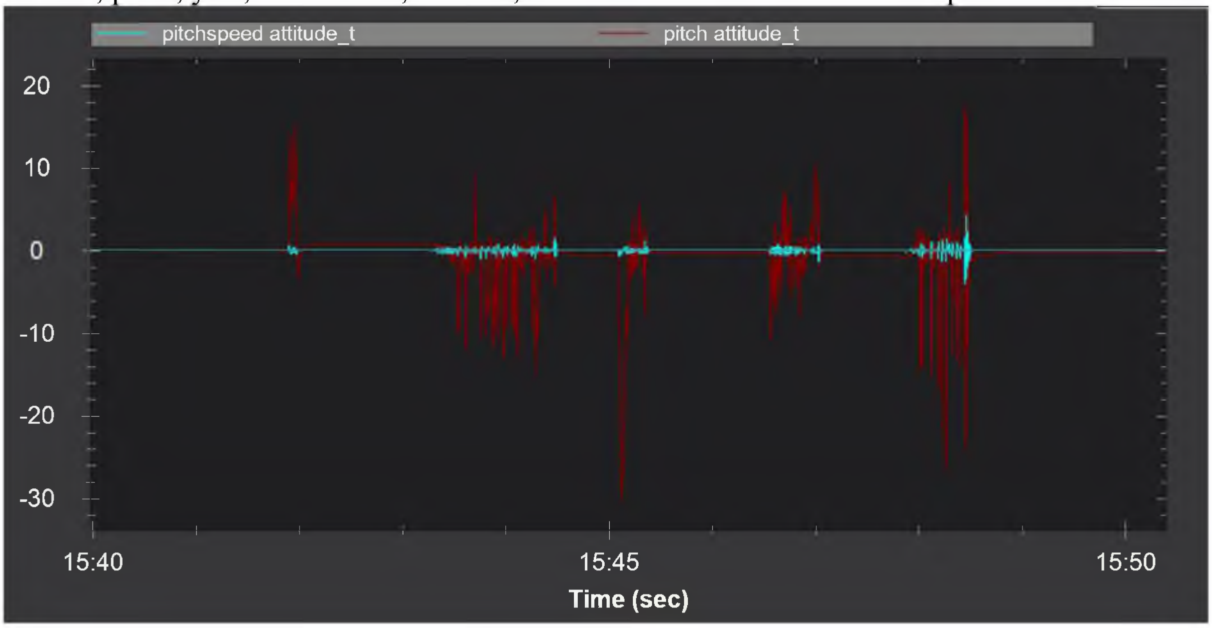


Figure 5.1(a): Pitch speed attitude and Pitch attitude comparison with respect to time.

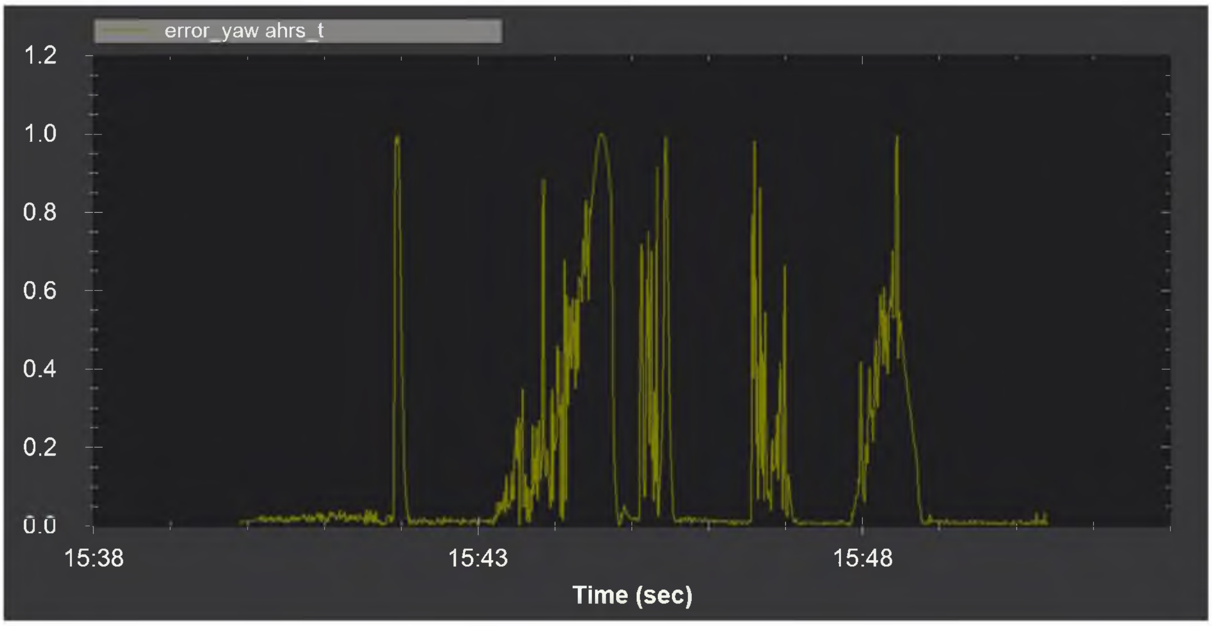


Figure 5.1(b): Error rate of Yaw with respect to time.

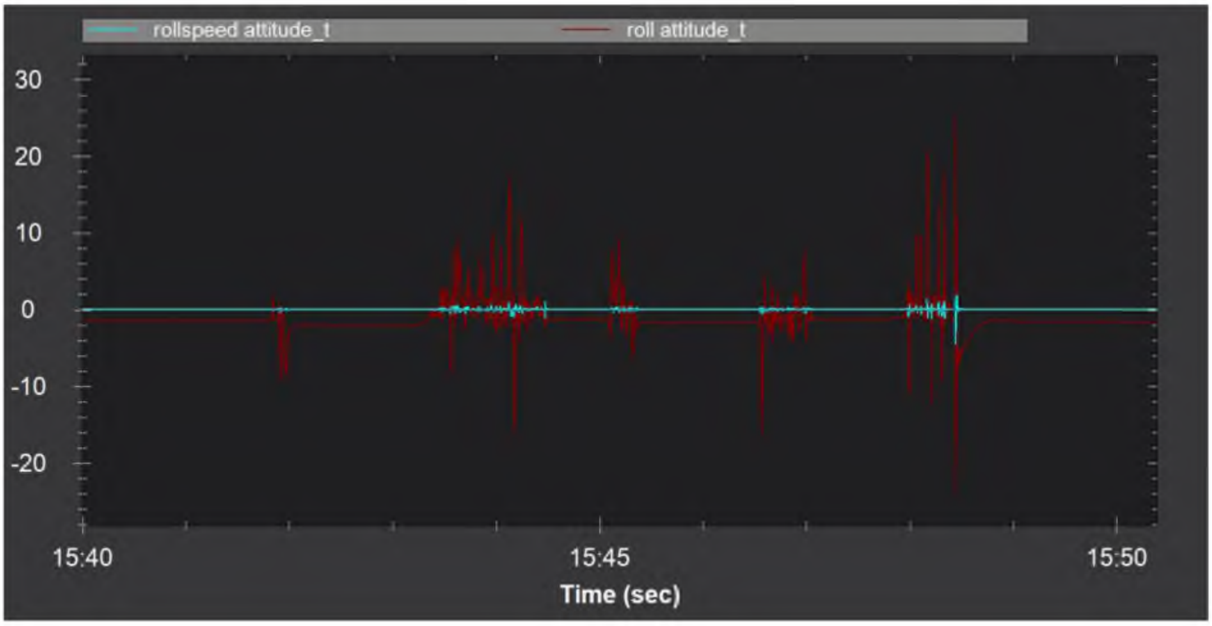


Figure 5.1(c): Roll speed attitude and Roll attitude comparison with respect to time.

Figure 5.1(d): Yaw speed attitude and Yaw attitude comparison with respect to time.

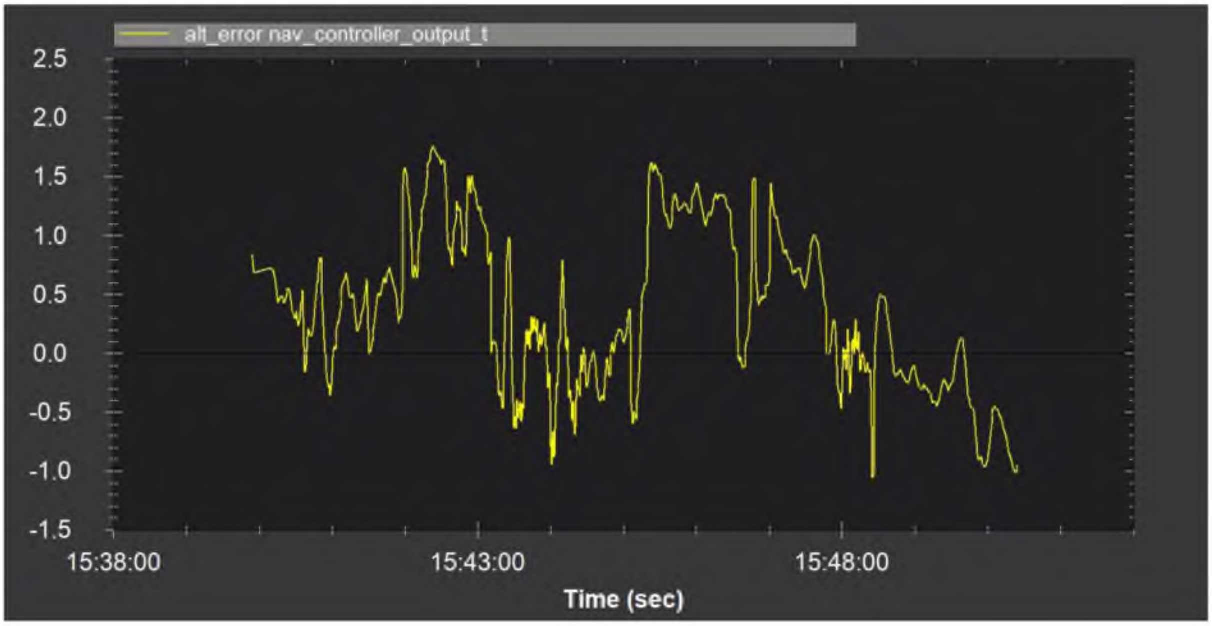


Figure 5.1(e): Altitude error with respect to time

Figure 5.1(e): Altitude from GPS with respect to time.

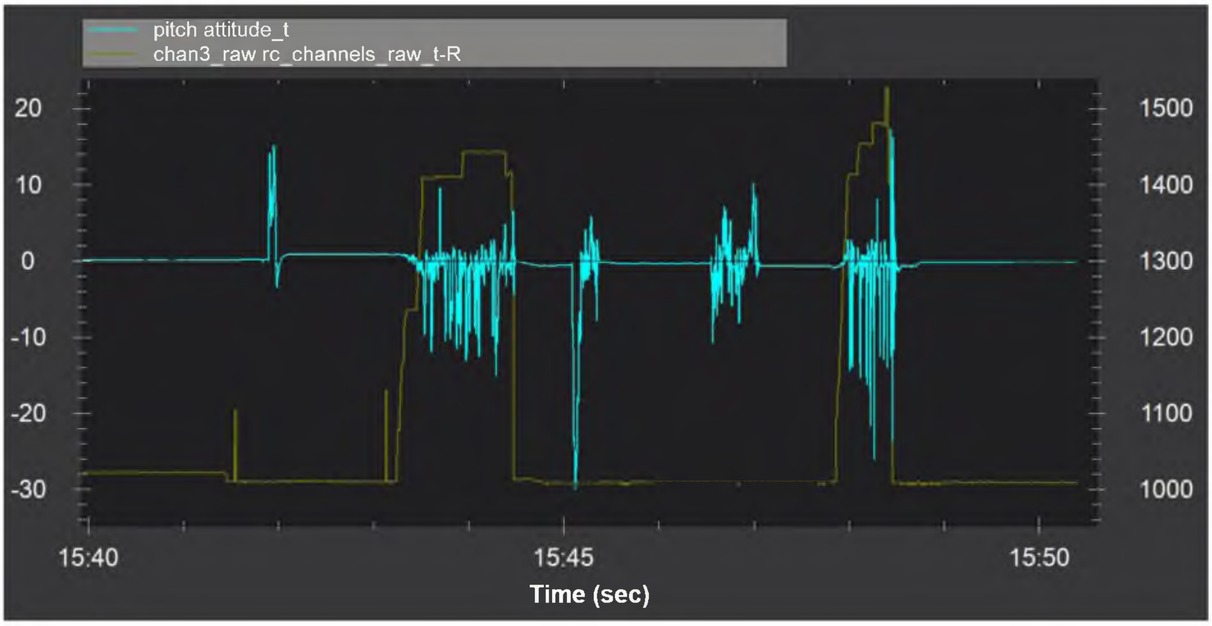


Figure 5.1(f): Pitch attitude and Channel three of radio(throttle) response with respect to time.

Figure 5.1(g): Satellite response with respect to time.

The accuracy of the value and response time was remarkable. Two of our flight has GPS. By plotting the GPS value in The Google earth we were surprised to see the pinpoint accuracy of the location.

GPS DATA:

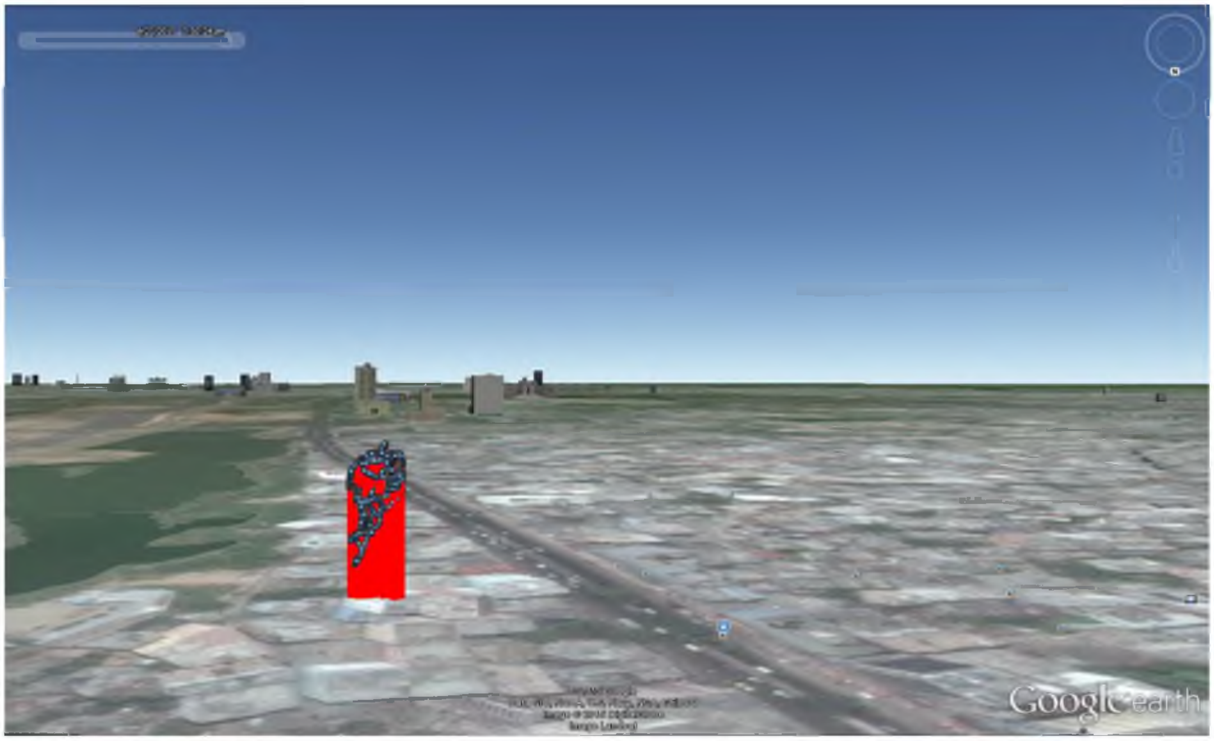


Figure 5.1(h): Roof top flight at East Shewrapara (GPS data plotting on Google earth).

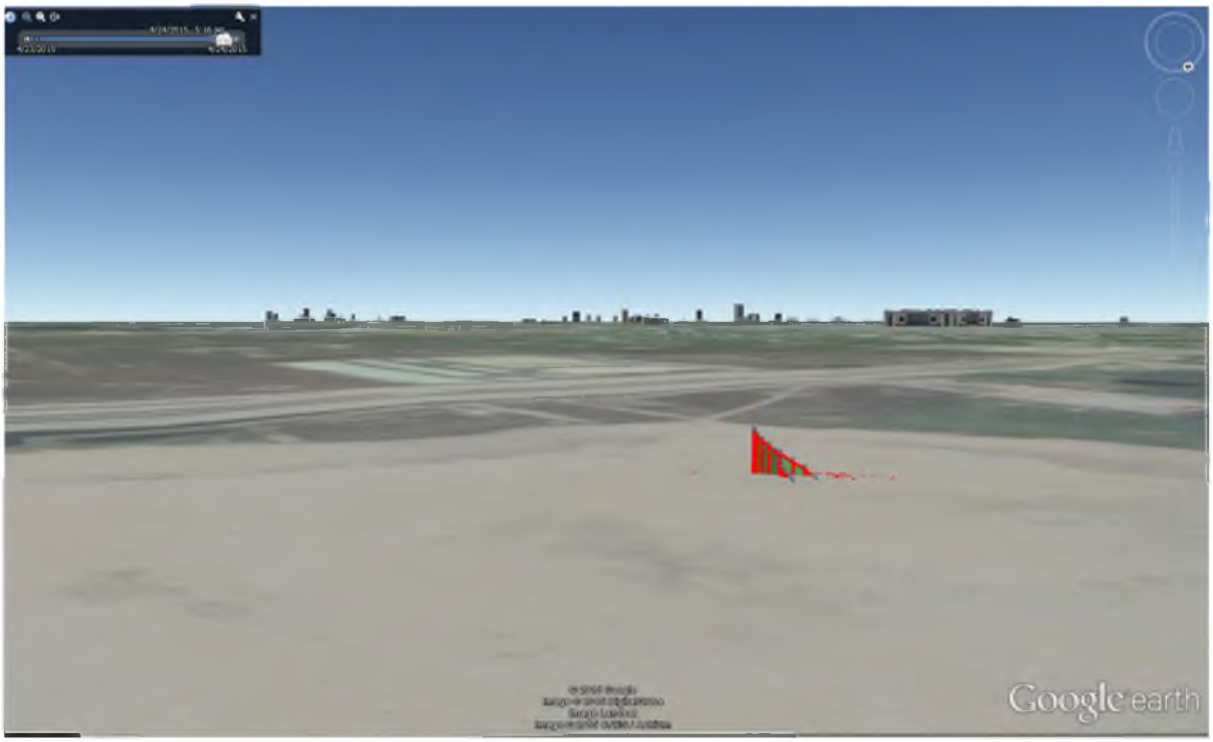


Figure5.1 (h): Flight at Purbachol Link Road (GPS data plotting on Google earth).

CHAPTER VI
CONLUSION

6.1 Conclusion

From our experimentations we found out that, fixed wing remote controlled aircraft presents us with enormous amount of challenges to program complete autonomous takeoff and landing along with imaging algorithms. Although larger wing aircraft increases payload size and decreases battery consumption, processing power required in order to successfully complete autonomous function is very high, hence adding more cost to entire project. On the other hand, ability of a quadcopter to hover when it encounters any unforeseen event (such as, obstructions, tree line or may be even another aircraft) gives us major benefit to simplify control algorithms. Moreover, we can get closer to target to capture photographs. This allows us to mount even cheaper cameras onboard a quadcopter. Only drawback of a quadcopter is its battery life. We found out, in cases where target area for mapping is less than 4 square kilometers, a medium capacity battery performs fairly well.

One of the major advantage of our design of a quadcopter frame is that, it is modular in design, cheap and highly shock absorbent. Since the frame is made from common PVC pipes, our frame can be made almost anywhere in the world. Even in the unlikely event of crash, entire body can be remade within matters of hours, which increases the usefulness of our design.

Our image processing samples are developed using open source computer vision library (OpenCV) making it completely flexible to any further adaptation or modification. Finally, our image processing programs are brought together by LabView software interface, making it easily understandable to practically anybody due to it’s vivid graphical representation.

6.2 Future Scope

Our developed system can be easily adopted to any other aerial photography application. Although our system is primarily developed for vegetation monitoring and river erosion mapping, with minor changes in software algorithm and swapping true color camera with other advanced imaging system, it can be easily adapted to case specific application. Such as, equipped with an infrared camera and motion tracking algorithm, we can monitor remote forest region for theft of forest resources, which belongs to the government. Equipped with a high resolution true color camera and pattern recognizing software, traffic control can divert crowd in any mass event, such as concert. Similarly, our system can be used in traffic monitoring and diversion, tracking suspects in real time, providing security in special events, monitoring and mapping a disaster struck area etc.

Reference

1. M. Aubert, S. Üzümcü, A. Hutchins and M. Cummings, 'Toward the Development of a Low-Altitude Air Traffic Control Paradigm for Networks of Small, Autonomous Unmanned Aerial Vehicles', 2015.
2. Ben Moshe, B.; Shvalb, N.; Baadani, J.; Nagar, I.; Levy, H., "Indoor positioning and navigation for micro UAV drones — Work in progress," *Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of* , vol., no., pp.1,5, 14-17 Nov. 2012 doi: 10.1109/EEEI.2012.6377021
3. C. Tetrault, 'A Short History of Unmanned Aerial Vehicles (UAVs)', *Draganfly.com*, 2015. [Online]. Available: http://www.draganfly.com/news/2009/03/04/a-short-history-of-unmanned-aerial-vehicles-uavs/. [Accessed: 16- Apr- 2015].
4. Rcgroups.com, 'Free plan for a nice AP plane. Big Brother. - RC Groups'. [Online]. Available: http://www.rcgroups.com/forums/showthread.php?t=454843. [Accessed: Nov- 2013].
5. *70" EXTRA 300 EXP ARF*, 1st ed. Extreme Flight RC.
6. Copter.ardupilot.com, 'common-APM 2.5 and 2.6 Overview | Copter'. [Online]. Available: http://copter.ardupilot.com/wiki/common-autopilots/common-apm25-and-26-overview/. [Accessed: Jun- 2014].
7. Batteryuniversity.com, 'Charles-Augustin de Coulomb's C-Rate for Batteries'. [Online]. Available: http://batteryuniversity.com/learn/article/what_is_the_c_rate. [Accessed: Feb- 2014].
8. Lithium Polymer Batteries, 1st ed. Horizon Hobby.
9. .com, 'ArduPilot | Open source autopilot'. [Online]. Available: http://ardupilot.com/. [Accessed: Jan- 2014].
10. S. Moges, W. Raun, R. Mullen, K. Freeman, G. Johnson and J. Solie, 'Evaluation of Green, Red, and Near Infrared Bands for Predicting Winter Wheat Biomass, Nitrogen Uptake, and Final Grain Yield'. DOI: 10.1081/PLN-200025858
11. Y. Koumpouros, B. Mahaman, et al.'Image processing for distance diagnosis in pest management', 2015.
12. 'A MAP Approach for Joint Motion Estimation, Segmentation, and Super Resolution', *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 16, no. 2, 2007.
13. Saska, M.; Krajnik, T.; Faigl, J.; Vonasek, V.; Preucil, L., "Low cost MAV platform AR-drone in experimental verifications of methods for vision based autonomous navigation," Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on , vol., no., pp.4808,4809, 7-12 Oct. 2012 doi: 10.1109/IROS.2012.6386277
14. Bartak, R.; Hrasko, A.; Obdrzalek, D., "A controller for autonomous landing of AR.Drone," Control and Decision Conference (2014 CCDC), The 26th Chinese , vol., no., pp.329,334, May 31 2014-June 2 2014 doi: 10.1109/CCDC.2014.6852167
15. Jimenez Lugo, J.; Zell, A., "Framework for autonomous onboard navigation with the AR.Drone," Unmanned Aircraft Systems (ICUAS), 2013 International Conference on , vol., no., pp.575,583, 28-31 May 2013 doi: 10.1109/ICUAS.2013.6564735
16. Camara, D., "Cavalry to the rescue: Drones fleet to help rescuers operations over disasters scenarios," Antenna Measurements & Applications (CAMA), 2014 IEEE Conference on , vol., no., pp.1,4, 16-19 Nov. 2014 doi: 10.1109/CAMA.2014.7003421

17. pvrille, L.; Tanzi, T.; Dugelay, J.-L., "Autonomous drones for assisting rescue services within the context of natural disasters," General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI , vol., no., pp.1,4, 16-23 Aug. 2014 doi: 10.1109/URSIGASS.2014.6929384
18. Venugopalan, T.K.; Taher, T.; Barbastathis, G., "Autonomous landing of an Unmanned Aerial Vehicle on an autonomous marine vehicle," Oceans, 2012 , vol., no., pp.1,9, 14-19 Oct. 2012 doi: 10.1109/OCEANS.2012.6404893
19. Tanzi, T.; Apvrille, L.; Dugelay, J.-L.; Roudier, Y., "UAVs for humanitarian missions: Autonomy and reliability," Global Humanitarian Technology Conference (GHTC), 2014 IEEE , vol., no., pp.271,278, 10-13 Oct. 2014 doi: 10.1109/GHTC.2014.6970292
20. Jayatilleke, L.; Nian Zhang, "Landmark-based localization for Unmanned Aerial Vehicles," Systems Conference (SysCon), 2013 IEEE International , vol., no., pp.448,451, 15-18 April 2013 doi: 10.1109/SysCon.2013.6549921
21. Pestana, J.; Sanchez-Lopez, J.L.; Saripalli, S.; Campoy, P., "Computer vision based general object following for GPS-denied multirotor unmanned vehicles," American Control Conference (ACC), 2014 , vol., no., pp.1886,1891, 4-6 June 2014 doi: 10.1109/ACC.2014.6858831
22. Carro Perez, I.; Flores-Araiza, D.; Fortoul-Diaz, J.A.; Maximo, R.; Gonzalez-Hernandez, H.G., "Identification and PID control for a quadrocopter," Electronics, Communications and Computers (CONIELECOMP), 2014 International Conference on , vol., no., pp.77,82, 26-28 Feb. 2014 doi: 10.1109/CONIELECOMP.2014.6808571
23. Xiao-Long Chen; Qiang Tang; Jun Che, "An indoor quadrotor locating and object-following algorithm using monocular vision," Intelligent Control and Automation (WCICA), 2012 10th World Congress on , vol., no., pp.4747,4753, 6-8 July 2012 doi: 10.1109/WCICA.2012.6359378

Appendix

A. Code

A.1. Crop Health Estimation

```
//============================================================================
// Name        : Health_Estimation.cpp
// Author      : Maisun Monowar, Abdulla Kafi, Raihana Antara, Munir Maruf
// Version     : 02
// Copyright   : GPL V.2
// Description : This program analyzes video from default source and calculates
//               crop health and makes appropriate suggestion to user.
//Online Reporsitory of this code is at github.com/maisunmonowar/workspace
//============================================================================
//including required files to compile this program
#include <iostream>
#include <opencv2/opencv.hpp>
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
//Using Standard Namespace
using namespace std; //Using OpenCV Namespace
using namespace cv;//Defining 'green' color in HSV color space
#define green_h_h 80
#define green_h_l 45
#define green_s_h 255
#define green_s_l 30
#define green_v_h 200
#define green_v_l 30
//Defining "Healthy Green" Color in HSV color space
#define healthy_h_h 65
#define healthy_h_l 55
#define healthy_s_h    255
#define healthy_s_l 30
#define healthy_v_h 200
#define healthy_v_l 30
//Main Function
int main(int, char**)
{
        //Initializing Variables to use in the program

        Mat img_hsv;  Mat imgThresholded;
        Mat healthyThreshold;          Mat edges;      int total_pixel = 0;
        int total_green = 0;     int healthy_green = 0;
        float percentage_green = 0;float percentage_healthy_green ;// open the default camera
        VideoCapture cap(0); // check if we succeeded
        if(!cap.isOpened())               return -1;
        //Creating windows to display results
        namedWindow("edges",1);
        namedWindow( "Original Image", WINDOW_AUTOSIZE );
        namedWindow( "Thresholded Image", WINDOW_AUTOSIZE );
        namedWindow( "Healthy Thresholded Image", WINDOW_AUTOSIZE );
```

```
//Running an infinite loop to process incoming videos
for(;;)
{
//Create a Temporary Matrix to hold image
Mat image;
// get a new frame from camera
cap >> image;          total_pixel = image.total();
//Convert image to HSV color space
cvtColor(image, img_hsv, COLOR_BGR2HSV);
//Thresholding image to display
inRange(img_hsv, Scalar(green_h_l, green_s_l, green_v_l), Scalar(green_h_h,
green_s_h, green_v_h), imgThresholded); //Threshold the image
inRange(img_hsv, Scalar(healthy_h_l, healthy_s_l, healthy_v_l), Scalar(healthy_h_h,
healthy_s_h, healthy_v_h), healthyThreshold);
//Following block of code is credited to
//http://stackoverflow.com/questions/7899108/opencv-get-pixel-channel-value-from-
mat-image
for(int i = 0; i < img_hsv.rows; i++)
{
    for(int j = 0; j < img_hsv.cols; j++)
    {
        Vec3b hsvPixel = img_hsv.at<Vec3b>(i, j);
        // Compare Each pixel
        if (hsvPixel.val[1] > green_s_l)
        {
            if (hsvPixel.val[2]>green_v_l && hsvPixel.val[2]<green_v_h)
            {
                if (hsvPixel.val[0]>green_h_l &&
hsvPixel.val[0]<green_h_h)
                {
                    //this Pixel is a "Green" pixel as defined above
                    total_green++;


                    //Searching for "Healthy Green"
                    if (hsvPixel.val[1] > healthy_s_l)
                    {
                        if (hsvPixel.val[2]>healthy_v_l &&
hsvPixel.val[2]<healthy_v_h)
                        {
                            if (hsvPixel.val[0]>healthy_h_l
&& hsvPixel.val[0]<healthy_h_h)
                            {
                                //This Pixel is "Healthy
Green"
                                healthy_green++;
                            }}}}} }}
}
//Pixel comparison complete
    //Calculating Percentage of Green Pixel
```

```
        percentage_green = total_green * 100.0 / total_pixel;
        //Calculating Percentage of Healthy Green Pixel
        percentage_healthy_green = healthy_green * 100.0/ total_green ;
        //display the result on window
        imshow( "Original Image", image );
        imshow( "Thresholded Image", imgThresholded );
        imshow( "Healthy Thresholded Image", healthyThreshold );
                //Save result as image if required
                //imwrite("imgThresholded.jpg", imgThresholded);
                //imwrite("healthyThreshold.jpg", healthyThreshold);
        //state calculation data to user
        std::cout <<"Total Pixel ";                 std::cout <<total_pixel;
        std::cout <<"\nGreen Pixels ";              std::cout <<total_green;
        std::cout <<"\nHealthy Pixels ";            std::cout <<healthy_green;
        std::cout <<"\nGreen percentage ";                  std::cout <<percentage_green;
        std::cout <<"% \nHealthy green  ";                  std::cout <<percentage_healthy_green;
        std::cout <<"% \n ";            std::cout <<"\n";        std::cout <<"\n";
        if((int)percentage_healthy_green < 20)
        {
                std::cout <<"Your farm needs expert opinion.\n\n";
        }
        else
        {
                if((int)percentage_healthy_green < 50)
                {
                        std::cout <<"Your farm appears to be unfertilized.\n\n";
                }
                else
                {
                        if((int)percentage_healthy_green < 80)
                        {
                                std::cout <<"Your farm appears to be in good shape.\n\n";
                        }
                        else
                        {
                                std::cout <<"Your farm appears to be in excellent shape.\n\n";
                }}
        }
      //break from infinite loop if user presses 'Esc' key
      if(waitKey(30) >= 0) break;
    }
    // the camera will be deinitialized automatically in VideoCapture destructor
    // Return an integer
    return 1;
}
```

A.2. Super Resolution Map

```
//============================================================================
// Name        : splitAndStich.cpp
```

```
// Author      : Maisun Ibn Monowar
// Version     : 2
// Copyright   : Sample from internet
// Description : Hello World in C++, Ansi-style
//============================================================
#include <iostream>
#include <fstream>
#include "opencv2/highgui/highgui.hpp"
#include <opencv2/imgproc/imgproc.hpp>
#include "opencv2/stitching/stitcher.hpp"
#include <sys/types.h>
#include <sys/stat.h>
using namespace std;using namespace cv;
vector<Mat> imgs;string result_name = "result.jpg";
int main(int argc, char* argv[])
{
	struct stat info;VideoCapture cap(argv[1]); // open the video file for reading
	if ( !cap.isOpened() )  // if not success, exit program
	{
		cout << "Cannot open the video file" << endl;return -1;
	}
	double fps = cap.get(CV_CAP_PROP_FPS); //get the frames per seconds of the video
	cout << "Frame per seconds : " << fps << endl;
	namedWindow("MyVideo",CV_WINDOW_KEEPRATIO); //create a window called
"MyVideo"
	int a = 0;	string name;
	if( stat( "image", &info ) != 0 )
	{
		cout <<  "creating directory >>"  "image" ;
					string folderName = "image";
					string folderCreateCommand = "mkdir " + folderName;
					system(folderCreateCommand.c_str());
	}
	while(1)
	{
		Mat frame;bool bSuccess = cap.read(frame); // read a new frame from video
		if (!bSuccess) //if not success, break loop
		{
			cout << "Cannot read the frame from video file" << endl;
			break;
		}
		//
		name = format("image/image_00%d.jpg", a);
		if(a%40)
		{
			//nothing to do
		}
		else
		{
			imwrite(name, frame);
```

```
		}
		a++;imshow("MyVideo", frame); //show the frame in "MyVideo" window if
required
		if(waitKey(30) == 27) //wait for 'esc' key press for 30 ms. If 'esc' key is
pressed, break loop
		{
			cout << "esc key is pressed by user" << endl;break;
		}
	}
	a = 0;	for (int a=0; a<=10000;a++)  // a <=Count would do one too many...
	{
		name = format("image/image_00%d.jpg", a);
		Mat img = imread(name);
		if ( img.empty() )
		{
			cerr << "image file " << name << " can't be loaded!" << endl;
			continue;
		}
		imgs.push_back(img);
	}
	Mat pano;	Stitcher stitcher = Stitcher::createDefault(false);
	Stitcher::Status status = stitcher.stitch(imgs, pano);
	imwrite(result_name, pano);   imshow("MyVideo", pano);   waitKey(0);    return 1;
}
```

A.3. River Line Detection

```
//===========================================================================
// Name        : River Line.cpp
// Author      : Maisun Monowar, Abdulla Kafi, Raihana Anto, Munir Maruf
// Version     : 1.0
// Copyright   : GPL V.2
// Description : This program takes an image file and displays edges
//Online Repository: github.com/maisunmonowar/workspace
//===========================================================================
//Including files necessary to compile
#include <iostream>
#include <fstream>
#include "opencv2/opencv.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/stitching/stitcher.hpp"
//Using Standard and OpenCV Namespace
using namespace std;using namespace cv;
//Main function
int main(int argc, char** argv)
{
	//Reading Image file
	Mat img = imread(argv[1]);
```

```
            //Checking for errors
            if (img.empty())
            {
                        cout << "Can't read image '" << argv[1] << "'\n";     return -1;
        }
    //if source is from video. following three lines should be used
    //VideoCapture cap(0); // open the default camera
    //if(!cap.isOpened())  // check if we succeeded
      //  return -1;
    Mat edges;    //creating display windows
    namedWindow("original",WINDOW_AUTOSIZE);
    namedWindow("first",WINDOW_AUTOSIZE);
    namedWindow("result",WINDOW_AUTOSIZE);
       //if source is video, use loop
    //for(;;)
    //{
       // Mat frame;
        //cap >> frame; // get a new frame from camera
        cvtColor(img, edges, CV_BGR2GRAY);
        imshow("original", edges);
        GaussianBlur(edges, edges, Size(9,9), 10, 10);
        GaussianBlur(edges, edges, Size(9,9), 10, 10);
        //imshow("second", edges);//debug purpose
        Canny(edges, edges, 10, 30, 3);
        imshow("result", edges);
        imwrite("result.jpg", edges);
  //    if(waitKey(30) >= 0) break;
    //}
    // the camera will be deinitialized automatically in VideoCapture destructor
    waitKey(0);
    //Program finished. Return an Integer
    return 1;
}
```

A.4. Aggregated DLL code

```
/*
Name            : aggregated_DLL.cpp
Author          : Maisun Monowar, Abdulla Kafi, Raihana Antara, Munir Maruf
Version         : 02
Copyright       : GPL V.2
Description     : This is a code for a DLL file which is to be called by
                  LabView GUI

Online Repo     :github.com/maisunmonowar/workspace
*/
//Including files in order to compile the program correctly
#include "stdafx.h"
#include "Windows.h"
```

```
#include "extcode.h"
#include "dlltr3.h"      //Name of our custom made header file. To be changed as necessary
#include <iostream>
#include <fstream>
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <string>
#include <string.h>
#include <stringapiset.h>
#include <winstring.h>
#include <xstring>
#include <mshtml.h>
#include <atlbase.h>
#include <comdef.h>
#include "opencv2/opencv.hpp"
#include "opencv2/stitching/stitcher.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
//Using standard and OpenCV namespace
using namespace std;
using namespace cv;
//defining 'green'
#define green_h_h 80
#define green_h_l 45
#define green_s_h 255
#define green_s_l 30
#define green_v_h 200
#define green_v_l 30
#define healthy_h_h 65
#define healthy_h_l 55
#define healthy_s_h 255
#define healthy_s_l 30
#define healthy_v_h 200
#define healthy_v_l 30

//function for Health Estimate
DLLIMPORT int32_t healthEstimate(double *g_percent, double *h_g_percent)
{
        //Declaring Local Variables
        Mat img_hsv, imgThresholded, healthyThreshold;
        int total_pixel = 0;
        int total_green = 0;
        int healthy_green = 0;
        float percentage_green = 0;
        float percentage_healthy_green ;
        // open the default camera
        VideoCapture cap(0);
        // check if succeeded
        if(!cap.isOpened())
```

```
		return -1;
	Mat edges;
	Mat image;
	// get a new frame from camera
	cap >> image;
	total_pixel = image.total();
	cvtColor(image, img_hsv, COLOR_BGR2HSV);
	inRange(img_hsv, Scalar(green_h_l, green_s_l, green_v_l), Scalar(green_h_h,
green_s_h, green_v_h), imgThresholded); //Threshold the image
	inRange(img_hsv, Scalar(healthy_h_l, healthy_s_l, healthy_v_l), Scalar(healthy_h_h,
healthy_s_h, healthy_v_h), healthyThreshold);
	//Following block of code is credited to
//http://stackoverflow.com/questions/7899108/opencv-get-pixel-channel-value-from-mat-
image
	for(int i = 0; i < img_hsv.rows; i++)
	{
		for(int j = 0; j < img_hsv.cols; j++)
		{
			Vec3b hsvPixel = img_hsv.at<Vec3b>(i, j);
			// Calculation with BGR values...
			if (hsvPixel.val[1] > green_s_l)
			{
				if (hsvPixel.val[2]>green_v_l && hsvPixel.val[2]<green_v_h)
				{
					if (hsvPixel.val[0]>green_h_l &&
hsvPixel.val[0]<green_h_h)
					{
						//This pixel is a "Green" pixel
						total_green++;
						//Searching for "Healthy Green" pixel
						if (hsvPixel.val[1] > healthy_s_l)
						{
							if (hsvPixel.val[2]>healthy_v_l &&
hsvPixel.val[2]<healthy_v_h)
							{
								if (hsvPixel.val[0]>healthy_h_l
&& hsvPixel.val[0]<healthy_h_h)
								{
									//This pixel is "Healthy
Green"
									healthy_green++;
								}}}}}}}
	}
	//Calculating Percentage values
	percentage_green = total_green * 100.0 / total_pixel;
	percentage_healthy_green = healthy_green * 100.0/ total_green ;
	//Passing the values to LabView interface
	*g_percent = percentage_green ;
	*h_g_percent = percentage_healthy_green ;
	// the camera will be deinitialized automatically in VideoCapture destructor
```

```
        //Termination of the function
        return 1;
}
//Function of Super Resolution Map
DLLIMPORT int32_t spiltAndStitch(char pString[] )
{
        //Declaring Variables
        vector<Mat> imgs;
        string result_name = "result.jpg";
        struct stat info;
        // open the video file for reading
        VideoCapture cap(pString);
        // if not success, exit program
        if ( !cap.isOpened() )
        {
                cout << "Cannot open the video file" << endl;
                return -1;
        }
        //Offsetting the video if necessary
        //cap.set(CV_CAP_PROP_POS_MSEC, 300); //start the video at 300ms
        //get the frames per seconds of the video
        double fps = cap.get(CV_CAP_PROP_FPS);
        cout << "Frame per seconds : " << fps << endl;

        namedWindow("MyVideo",CV_WINDOW_KEEPRATIO); //create a window called
"MyVideo"
        int a = 0;
        string name;
        if( stat( "image", &info ) != 0 )
        {
                cout <<  "creating directory >>"  "image" ;
                //make a dir
                string folderName = "image";
                string folderCreateCommand = "mkdir " + folderName;
                system(folderCreateCommand.c_str());
        }
        while(1)
        {
                Mat frame;
                // read a new frame from video
                bool bSuccess = cap.read(frame);
                //if not success, break loop
                if (!bSuccess)
                {
                        cout << "Cannot read the frame from video file" << endl;
                        break;
                }
                name = format("image/image_00%d.jpg", a);
                if(a%40)
                {
```

```
                //nothing to do
            }
            else
            {
                imwrite(name, frame);
            }
            a++;
            //show the frame in "MyVideo" window if required
            imshow("MyVideo", frame);
            //wait for 'esc' key press for 30 ms. If 'esc' key is pressed, break loop
            if(waitKey(30) == 27)
            {
                cout << "esc key is pressed by user" << endl;
                break;
            }
        }
        a = 0;
        for (int a=0; a<=10000;a++)  // a <=Count would do one too many...
        {
            name = format("image/image_00%d.jpg", a);
            Mat img = imread(name);
            //Checking successful image loading
            if ( img.empty() )
            {
                cerr << "image file " << name << " can't be loaded!" << endl;
                continue;
            }
            imgs.push_back(img);
        }
        Mat pano;
        Stitcher stitcher = Stitcher::createDefault(false);
        //Using OpenCV Panoramic Stitching class
        Stitcher::Status status = stitcher.stitch(imgs, pano);
        //Saving result to hard drive
        imwrite(result_name, pano);
        //Showing result to graphical window
        imshow("MyVideo", pano);
        waitKey(0);
        //termination of program
        return 1;
}
DLLIMPORT int32_t riverLine(char pString[])
{
        Mat img = imread(pString);
        //Validating image
        if (img.empty())
        {
            cout << "Can't read image '" << pString << "'\n";
            return -1;
        }
```

```
        Mat edges;
        //Creating Graphical Windows
        namedWindow("original",WINDOW_AUTOSIZE);
        namedWindow("result",WINDOW_AUTOSIZE);
        //Converting image to grayscale
        cvtColor(img, edges, CV_BGR2GRAY);
        //Displaying original image
        imshow("original", edges);
        GaussianBlur(edges, edges, Size(9,9), 10, 10);
        GaussianBlur(edges, edges, Size(9,9), 10, 10);
        //imshow("second", edges);//debug purpose
        Canny(edges, edges, 10, 30, 3);
        //Displaying resultant image
        imshow("result", edges);
        //Saving Result
        imwrite("result.jpg", edges);
        //if halting program is required
        //waitKey(0);
        //termination of program
        return 1;
}
```

A.5. Aggregated DLL header file

```
#pragma once
#ifndef _DLL_H_
#define _DLL_H_
#define DLLIMPORT __declspec (dllexport)
DLLIMPORT int32_t spiltAndStitch(char* argv[]);
#endif
```

B. Additional Diagrams / Pictures

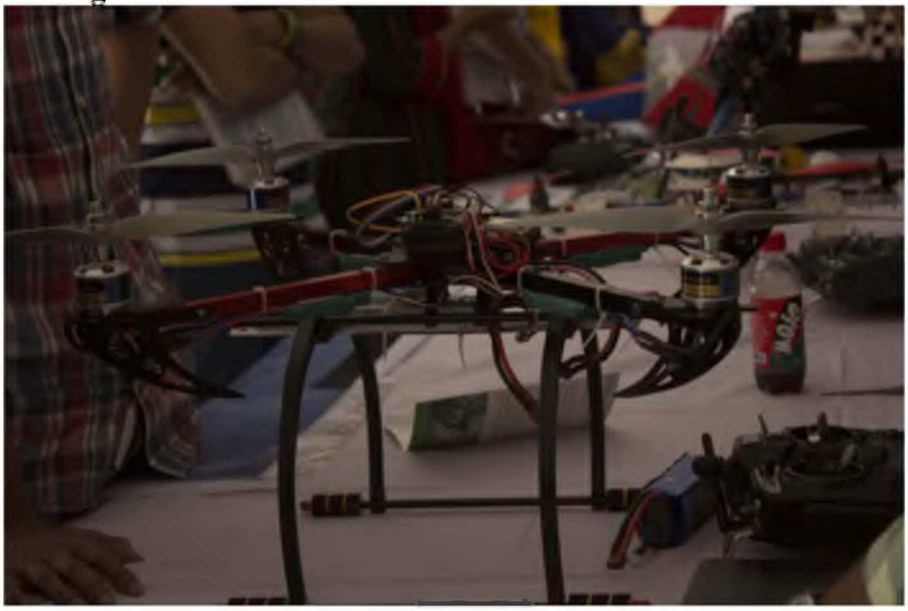


Figure 1: Participating in IUB Hover Fest, 2014.

Figure 2: FPV still image capture

Figure 3: Quad Copter after crashing

Figure 4: Indoor Flight Test

C. Additional files / links / resources

1. Test video file used for testing “Super_map.cpp” can be found at, https://www.youtube.com/watch?v=mSu9nrARi6Y
2. http://www.grc.nasa.gov/WWW/k-12/UEET/StudentSite/aeronautics.html
3. http://www.fpvforme.com/fpv-system-setup-first/
4. https://alselectro.wordpress.com/2014/10/13/quadcopter1-things-you-need-to-build-your-own-copter/
5. http://www.github.com/itseez/
6. http://docs.opencv.com/
7. Open CV codes http://www.github.com/itseez/
8. Open CV documentations http://docs.opencv.com/
9. Running Open CV sample program in Lab View https://www.youtube.com/watch?v=M6v9Xf4_KaI
10. Green Color detection in Open CV https://www.youtube.com/watch?v=wBOZLHPz9_I
11. Video to panorama test run https://www.youtube.com/watch?v=hihNAMuDwJw

D. Air TABLE

Purpose of air table

Before launching a satellite, it is needed to test the attitude control devices (Magnetic torque, sun sensor, and gyro) on board. Attitude control occurs in all three axes in the frictionless environment of space. This means that the Air table must simulate these aspects of space as accurately as possible in order to properly test the attitude control devices.Air bearing can not satisfy all the requirements of testing facility, an interface is needed between the satellite and the air bearing and that is the test bed (jig).

Relation to magnetic field

HORYU 4 has one permanent magnet to stabilize the attitude control passively. To know the attitude of this permanent magnet due to magnetic field of orbit this simulator is needed.If we have Helmholtz Coils then we can create the simulator of the magnetic field in the orbit with air table

General specifications of air bearing acquired by kyutech:

- Freedom of rotation in 3 directions
- Center of rotation stability of ±5 millionths of an inch (near 2 millionths of a cm)
- Range of motion up to +/- 45 degrees
- Load capacity from 1 lb. to 150 lbs.(68kg)
- Operating pressure from 30 psi to 90 psi (206 Kpa to 620 Kpa)

Specific requirements for Kyutech air table:

- Make a Air table jig where HORYU 4 can be placed (max. 60kg, max. 50cm)
- Get the gyro data from Air table wirelessly to know the position of our testing satellite
- Make a software interface for the user to see the data of gyro

Design consideration

•In order to accommodate wide variety of small satellite the test bed has to be highly adjustable in every dimension

•The center of mass of entire assembly must be aligned above the center of rotation.

•It is assumed that center of mass will be located within 20 % of the satellite radius from the geometric center

Air table design and dimension

Material has change to Aluminum 5052 from Aluminum Alloy 6061 T-6. As newer is more cheaper

Material : Aluminum 5052-O

Young's Modulus --- 7 x 10^10 N/m^2

Yield / Stress Strength = 9 x 10^7 N/m^2

Target Load– 600 N or 61.2244 Kg

From Stress equation, Stress Strength = Force / Cross Section Area

=> Area = Force / S.S

=>Area = 600/90mm = 6.6 mm^2

If we fix our length to 700 mm then we get the width parameter of 9.52 x 10^(-3) mm

Model Dimension

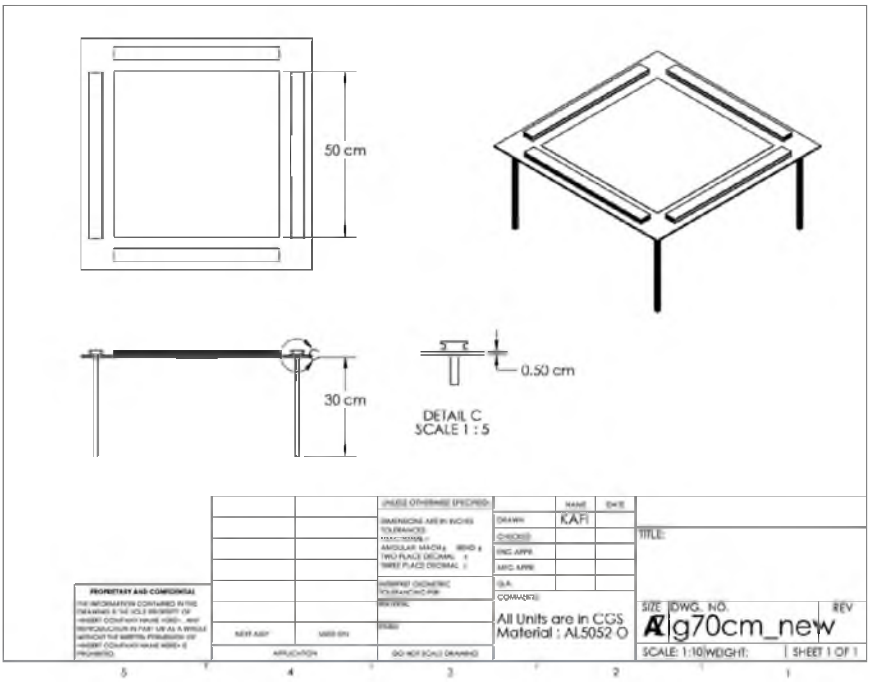


Figure Air Table Dimension

Stress test :

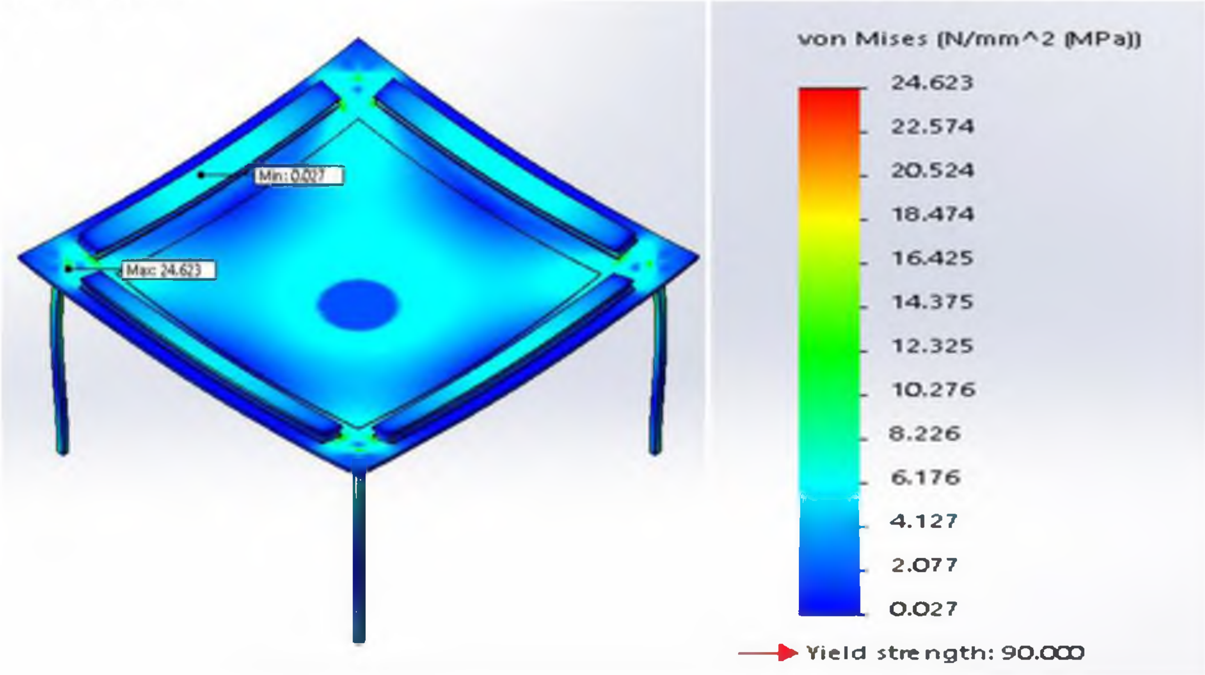


Figure : Stress Analysis

Here FACTOR OF SAFETY FOS is 3.655 for 600N force .FOS: A factor of safety of 1.0 at a location indicates that the material at that location has just started to fail The material at a location will start to fail if you apply new loads equal to the current loads multiplied by the resulting factor of safety, and assuming that the stresses/strains remain in the linear range.

Gyro Value for Air table

Name: L3GD20

Selectable Resolution:

- 250 dps
- 500 dps
- 2000 dps

Output Interface:

- Two wire Communication
- SPI Communication

16 bit Gyro value data Output.At lowest at dps resolution is 0.008 d/s.

Wireless Communication.

Equipment Using

- Xbee pro s2b – 2
- 2 Arduino

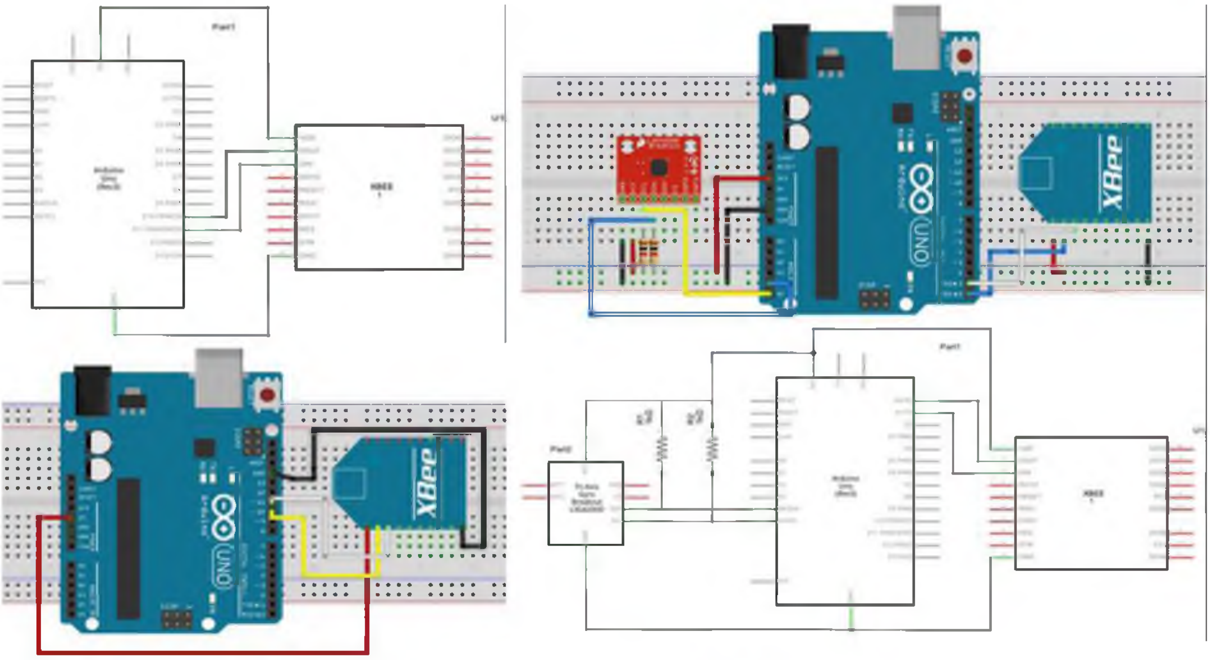


Figure : Circuit Diagram of Air-Table

Software interface

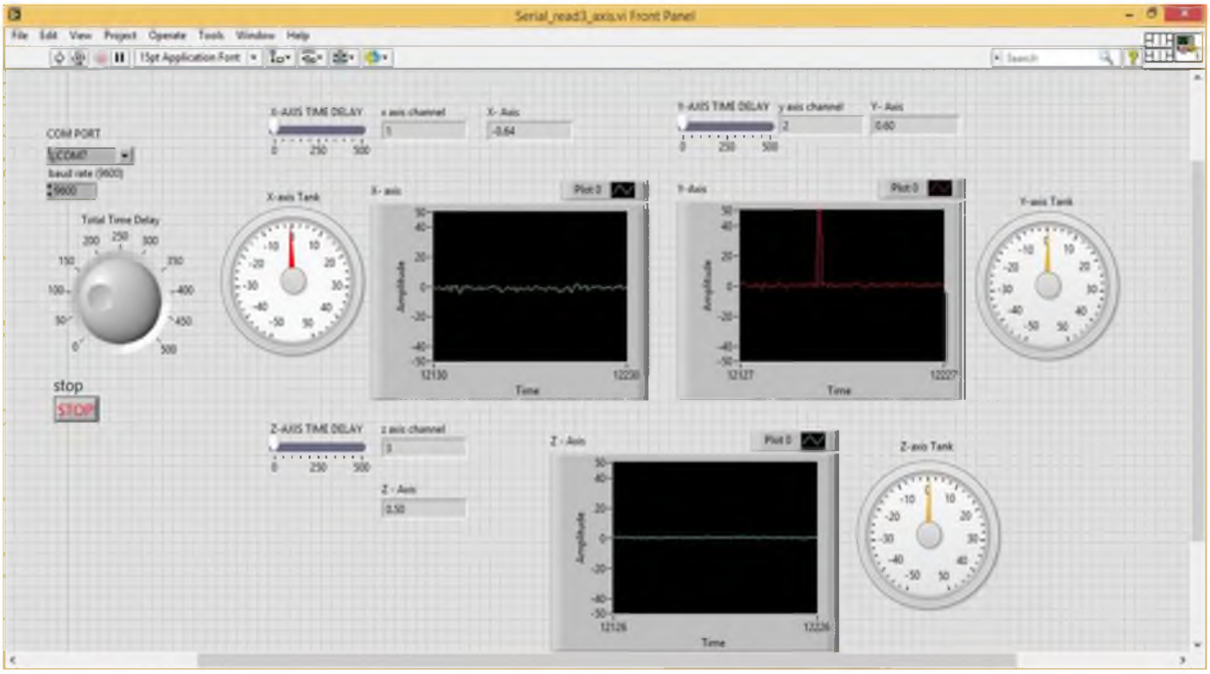


Figure : Software Interface of Air-Table

Final Completed project:

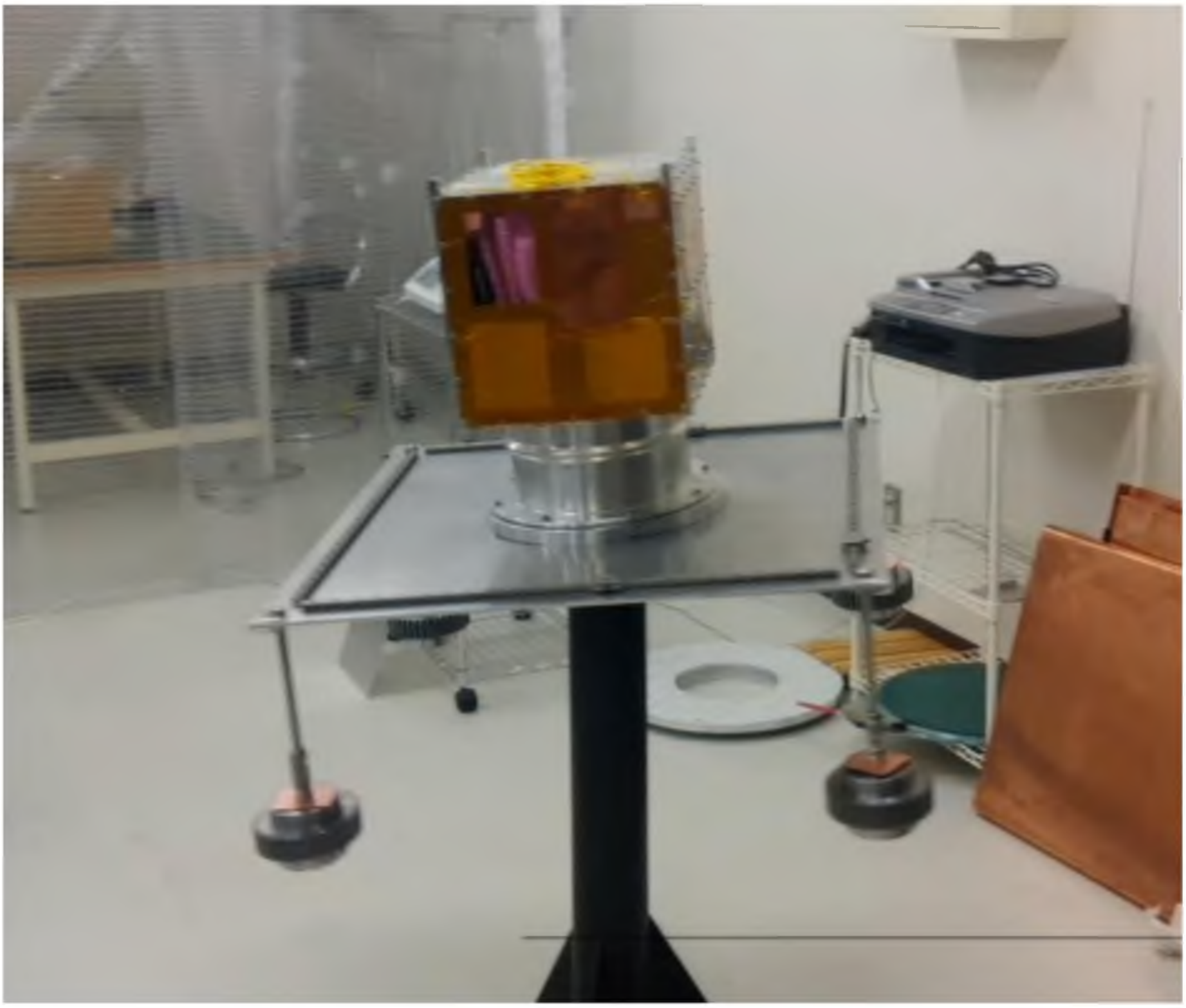


Figure : Final Air Table

E. Turn Table

Definition

Turn Table or Rate Table is a test platform which Rotates with a constant angular velocity according to user input equipped with a self-correcting feedback system.

Purpose

Gyro sensors are very useful to understand orientation of the attached platform in free space. A three axis gyroscope can give us three dimensional orientation of a body in free space. Therefore enabling us to implement various control various control application.

MEMS fabricated Gyro sensors are excellent due to their small form factor and low power consumption. Due to its many advantages, it is being used almost anywhere space and power consumption limitation is crucial. Therefore in is used in Smartphone, indoor drone, Robotics and Nano-Satellites. However, MEMS fabricated gyros have one drawback. It generates random values during its course of operation. This random value or noise has a definite pattern. Unfortunately, the magnitude of noise changes each time the sensor is restarted. In order to eliminate the effect of noises, an algorithm has to be devised.

The purpose of the turntable would be to test the accuracy of gyro sensors, calibrated through software in laboratory test condition. If the gyro mounted device could be rotated with a known angular velocity, relative position of the table can be calculated from time of operation. Hence, the purpose of Turntable is to calibrate gyro sensors.

Design Constraints

According to instructions from Prof. Mengu Cho, following parameters were set as the design constraints.

1. Minimum rate of rotation : 0.0008 deg / sec or 0.000133 rpm
2. Maximum rate of rotation : 36 deg / sec or 6 rpm
3. Maximum payload capacity : 50 Kg
4. Maximum dimension of payload : 50 cm x 50 cm x 50 cm

Design Considerations

Motor selection

To rotate the shaft of turntable, following motor technologies were considered.

1. DC Motor
2. Stepper Motor
3. Induction Motor

a. DC motor

b. Stepper motor

c. Induction motor

DC motor has a high starting torque and most easy to control. DC motor can be easily controlled in both direction using DC motor controller. Control signal requires only a PWM signal.

Stepper motor is an ideal choice for this kind of application. But the precision of stepper motor can be as low as 0.17 deg. Therefore, a gear train is still required. Controlling stepper motor requires a well defined sequence of pulses.

Induction motor are mostly used in industrial and high power hungry solutions. Range of torque and speed is typically higher than DC motors. Controlling an induction motor requires a Variable Frequency Drive (VFD).

Considering all of the facts, DC motor was considered to be the best choice for this project.

Gear train selection

Minimum rate of rotation being too low, it is evident that no motor is capable of rotation is such a slow speed. (At this speed, more than 5 days is required to compile a complete revolution.) A robust gear train system had to custom made to fulfil requirements.

To construct required custom gear train system, following 3 types of gear train system were considered.

1. Spur Gear system
2. Helical Gear System
3. Planetary or epicyclic gear system

a. Spur gear

b. Epicyclic gear

c. Helical gear

Epicyclic gear has most efficient torque converting ratio, which is highly suitable for this purpose. However, building a epicyclic gear train in general, non specialized mechanical lab is not feasible. Also, specialized skill is required to build such gear train.

Helical or double helical gear are only used in fixed gear train. Since this project required a variable gear train, helical gear are not suitable.

Spur gear is the most basic type of gear. Although the gear ratio needed for this project is significantly high, simplicity of the entire project out weighed the complexity of other type of gear system. Therefore, spur gear were selected for this project.

Gear Ratio Calculation:

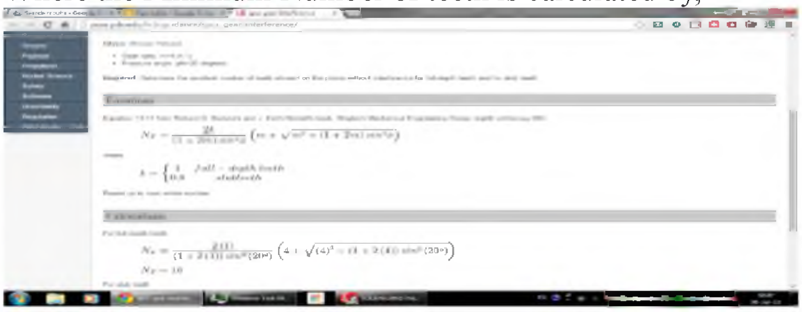
In order to achieve smooth rotation of our turntable from the lowest RPM to the highest RPM, a variable gear train has to be designed. Considering that the motor is a 6 RPM motor and using PWM control, minimum applicable PWM is 40%, the following table[1] of values were calculated.

Effective driving gear RPM	effective gear attenuation	min rpm of table	max rpm of table	min (pwm)	intermediate ratio	minimum teeth (t1)	t1 (input)	temp	t2 (output)
is	to	match	pattern						
6	45112.78195	0.000133	0.0003325	0.4	6.09375	15.96322207	16	97.5	98
6	7403.123193	0.0003241875	0.00081046875	0.4	2.4375	14.5861247	15	36.5625	37
6	3037.178746	0.0007902070313	0.001975517578	0.4	2.4375	14.5861247	15	36.5625	37
6	1246.02205	0.001926129639	0.004815324097	0.4	2.4375	14.5861247	15	36.5625	37
6	511.1885331	0.004694940994	0.01173735249	0.4	2.4375	14.5861247	15	36.5625	37
6	209.7183726	0.01144391867	0.02860979668	0.4	2.4375	14.5861247	15	36.5625	37
6	86.03830669	0.02789455177	0.06973637942	0.4	2.4375	14.5861247	15	36.5625	37
6	35.29776685	0.06799296993	0.1699824248	0.4	2.4375	14.5861247	15	36.5625	37
6	14.48113512	0.1657328642	0.4143321605	0.4	2.4375	14.5861247	15	36.5625	37
6	5.94097851	0.4039738565	1.009934641	0.4	2.4375	14.5861247	15	36.5625	37
6	2.437324517	0.9846862752	2.461715688	0.4	2.437324517	14.58597867	15	36.55986775	37
6	0.9999280069	2.400172796	6.00043199	0.4				0	0
connection gear (output shaft)			6.00043199						25

Output Driven Gear			6.00043199	0.4					37
Tabletop			6.00043199						
Satellite			6.00043199						

[Detailed table can be found at https://docs.google.com/spreadsheets/d/1Z9YStg5fb3d3rOviYSG2ZNva4gHGFl3Ak0XOCQnn5eY/edit?usp=sharing]

Where the Minimum Number of teeth is calculated by,



From the calculation, it is found out that the design constraints can be met by a 13 speed gear box.

Simulation Results

Based on computational data, a 3D model was made in SolidWorks to validate the complete concept of our build.

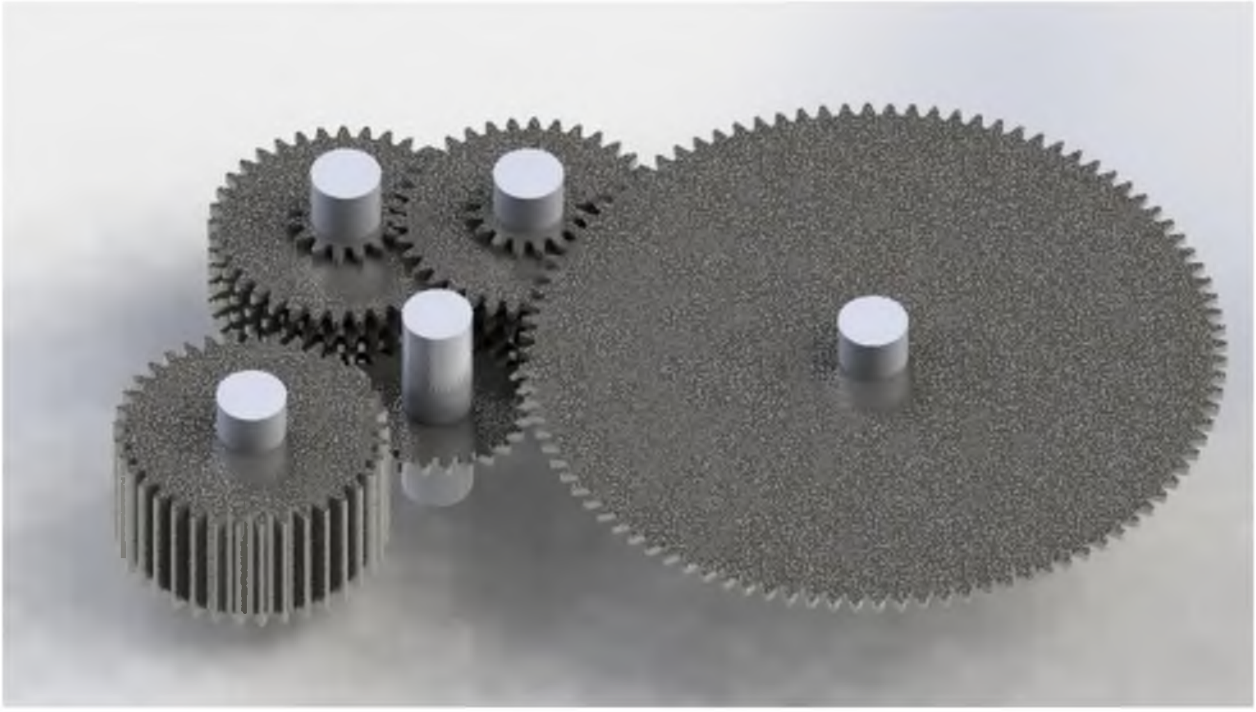


Figure: 13 speed gear box (top view)

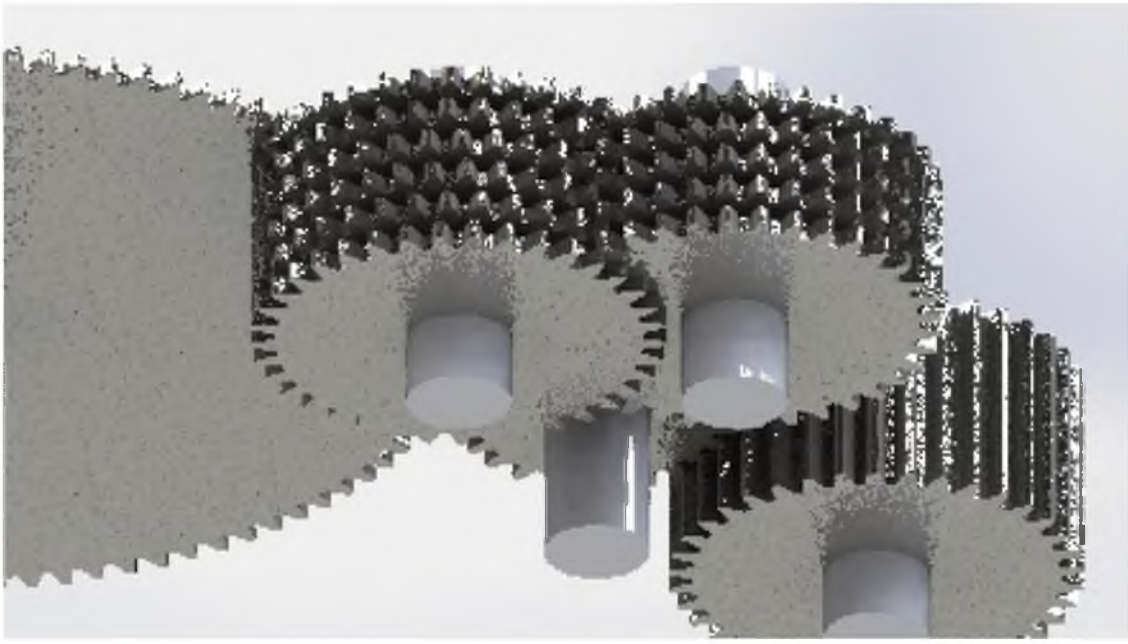


Figure: 13 speed gear box (bottom view)

Figure: Isometric view of Turn Table

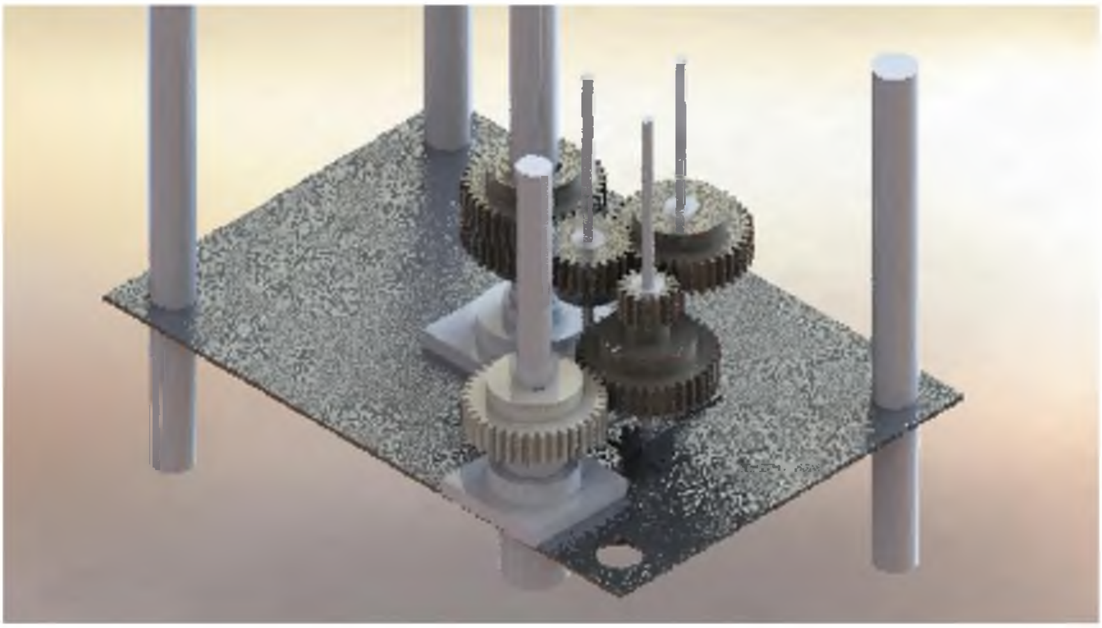


Figure: Close up view of Gear Train (simplified)

Build

After 3D simulation was perfected, with the approval of Prof. Cho and Dr. Polansky, parts were ordered for manufacturing.
Assembly of the table was completed except motor control. Since the frictional loss couldn't be calculated, the power of the motor couldn't be determined.

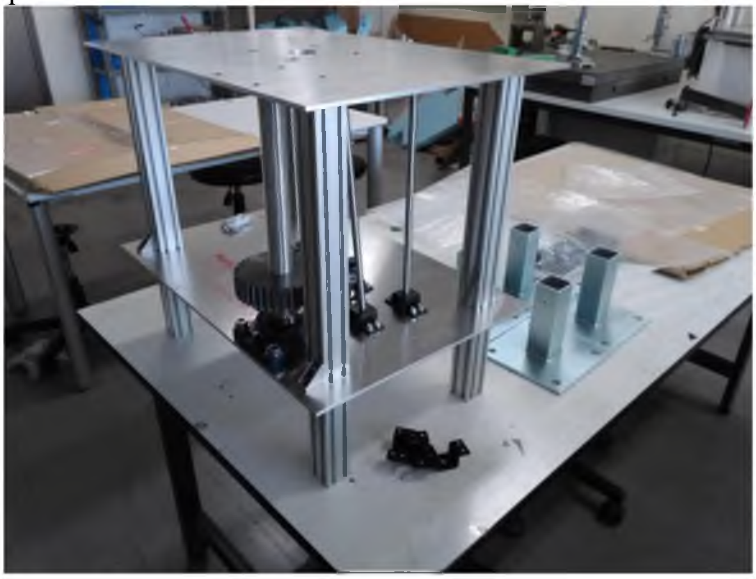


Figure: Partial assembly of rate table.

Additional Files

Additional files related turntable can be found online in the following stated path.

Description	File path
Gear train calculation table	https://docs.google.com/spreadsheets/d/1Z9YStg5fb3d3rOviYSG2ZNva4gHGFl3Ak0XOCOnn5eY/edit?usp=sharing
Rate Table Report	https://docs.google.com/document/d/1PNDeq56LqcbhtGVAT_5YWcgk3IzC44qdcdhJXWCfoJ4/edit?usp=sharing
Presentation file	https://docs.google.com/presentation/d/1RSq3WZE9nqJFLrk_1M_UsXWKlIevLyv5d0Y6ATDFkGw/edit?usp=sharing
Weight of the components	https://docs.google.com/spreadsheets/d/1u_9Z4Zl7DpJ2PIgVUZl-M3taFaV82g7N826pF_2JzIU/edit?usp=sharing
Housing plate (bottom) [CAD file]	https://drive.google.com/file/d/0B8ed5jmYlINAZ0h5dGxVai1obVU/view?usp=sharing
Housing plate (top) [CAD file]	https://drive.google.com/file/d/0B8ed5jmYlINAR1pXcHM4RlVwRW8/view?usp=sharing
Motor shaft [CAD file]	https://drive.google.com/file/d/0B8ed5jmYlINAS0V0RHk0dGwxbk0/view?usp=sharing
Output shaft [CAD file]	https://drive.google.com/file/d/0B8ed5jmYlINAZ01wMDF5MWNMNlk/view?usp=sharing
Top plate [CAD file]	https://drive.google.com/file/d/0B8ed5jmYlINAS1Z5TzBiRWxjLWs/view?usp=sharing
Gear change system	https://www.youtube.com/watch?v=SKNOuBbau5o
Sub-assembly	https://www.youtube.com/watch?v=LcZWby0jhYs