



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA

PROPELLER-LESS MULTI-ROTOR DRONE

Author: Ignacio Aguirre Panadero
Director: Luke A. Wendt (University of Illinois at Urbana-Champaign)

Madrid
JUNIO 2016

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Ignacio Aguirre Panadero

DECLARA ser el titular de los derechos de propiedad intelectual de la obra:

Propeller-less Multi-rotor Drone,

que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducir la en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e

intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 21 de junio de 2016

ACEPTA



AGUIRRE

Fdo: Ignacio Aguirre Panadero

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
PROPELLER-LESS MULTI-ROTOR DRONE

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2015/16 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.


Fdo.: Ignacio Aguirre Fecha: 02.06.2016

Autorizada la entrega del proyecto
EL DIRECTOR DEL PROYECTO


Fdo.: Luke A. Wendt Fecha: 3.6.16

Vº Bº del Coordinador de Proyectos

Fdo:

Fecha:/...../.....

Propeller-less Multi-rotor Drone

Resumen

Autor: Ignacio Aguirre Panadero

Director: Luke Wendt

12 June 2016

1 Introducción

Los drones se están convirtiendo en algo cada vez más habitual en todo tipo de industrias y negocios. Por ejemplo, Amazon está comenzando un proyecto que en el se usan drones para distribuir sus productos. Pese a todo esto, las hélices utilizadas en aparatos multi-rotor se rompen con facilidad al entrar en contacto con cualquier objeto además de rasgar la piel de las personas con facilidad. En vez de utilizar las hélices habituales, en este proyecto se pretende utilizar un ventilador centrífugo para obtener propulsión. De esta manera, conseguiremos un drón más seguro que los clásicos con las hélices expuestas o incluso más seguro que aquellos que usan protectores de hélices ya que estas se siguen rompiendo y doblando al chocar. El diseño pretendido en este proyecto tendrá la rueda o turbina con los álabes encerrados en una carcasa o voluta de tal manera que no sea posible que de manera accidental se llegue a tocar el interior donde se encuentran los álabes girando a gran velocidad. Al estar la rueda y los álabes cubiertos por la carcasa será más difícil que se dañen al chocar.

1.1 Objetivos

- Diseñar un drón que vuele sin hélices, partiendo desde cero. Lo que conlleva a un diseño mecánico, eléctrico y de control.
- Diseñar un drón más seguro

1.2 Funciones y Características

Al comienzo del proyecto las expectativas eran muy altas. La idea era construir un drón capaz de volar como los drones comerciales. Esto implicaba un drón capaz de volar al aire libre, que fuese ligero, portátil y equilibrado con bajo consumo y mucha propulsión.

Al final, el diseño cambió hacia un drón de interior debido a la cantidad de nuevas leyes que restringen el uso de drones. También el hecho de volarlo al aire libre implica un problema adicional como son las perturbaciones que provoca el viento sobre el drón. Este cambio hace posible que se pueda diseñar un drón que vuele necesitando menos propulsión. A partir de aquí lo que se pretende es fabricar un drón capaz de levantarse del suelo y que se estabilice.

2 Diseño

2.1 Diseño Mecánico

El diseño final fue primero modelado en 3D a través del programa *SOLIDWORKS*. Usando los ficheros 3D se han podido fabricar y cortar las piezas del drón usando una cortadora láser .

Diseño Rueda dentada:

Se debe de alcanzar un compromiso entre los dos tipos de ruedas dentadas más habituales (radiales y backward inclined) en cualquier nuevo diseño, el cual en este caso es esencialmente una rueda dentada radial donde cada cuchilla está ligeramente inclinada hacia atrás. Este diseño, optimiza la rueda dentada para rangos medios de velocidad. El flujo de aire en este diseño es notablemente mayor que el de una rueda radial habitual y además se consiguen altas presiones en el flujo de salida más rápidamente que en una rueda inclinada hacia atrás. En la Figura1 se puede ver el diseño final.



Figura 1: Diseño rueda dentada

Fabricación:

Para la construcción del drón (marco, ruedas dentadas, volutas, etc..) se ha utilizado un material con núcleo de espuma de poliestireno recubierto por papel formando capas por arriba y por abajo del núcleo. Como se ha comentado anteriormente, la fabricación de las piezas se ha llevado a cabo utilizando técnicas de cortado láser. Se decidió finalmente por esta técnica

debido a su precisión y consistencia. También se consideró cortarlo a mano mediante el uso de herramientas de corte habituales, pero la falta de precisión de estos métodos pone en peligro la viabilidad del proyecto.

Rendimiento:

La manera de comprobar la propulsión máxima que puede generar cada ventilador es colocando cada uno hacia abajo sobre una báscula electrónica de tal manera que el ventilador ejerza una fuerza al empujar contra la báscula.

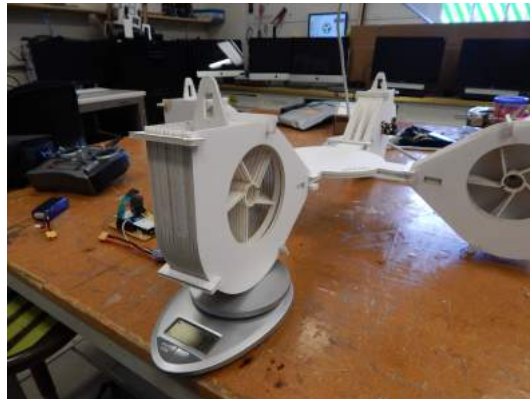


Figura 2: Test de propulsión

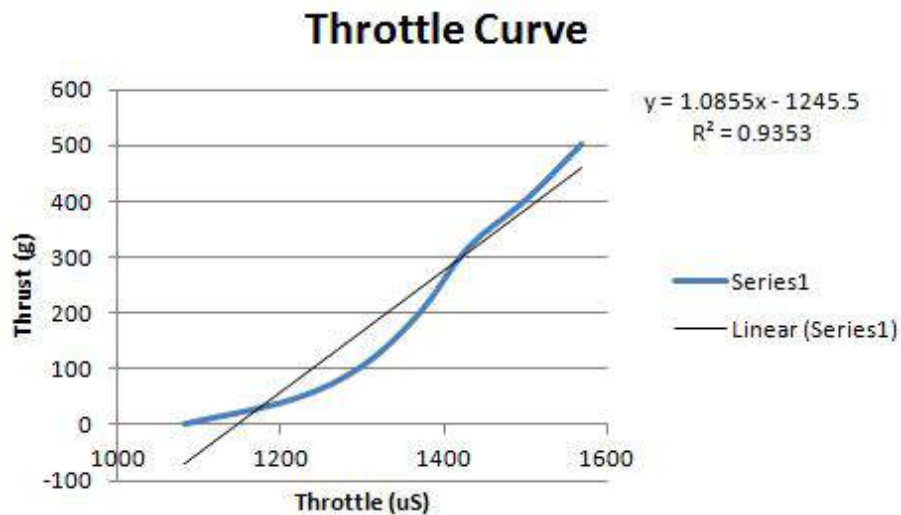


Figura 3: Propulsión en función de la velocidad exigida al motor

El *Throttle* en μS micro segundos representa el ancho de pulso de la señal PWM que controla los ESCs o controladores de velocidad de los motores eléctricos acoplados a la rueda del

ventilador. Nuestro rango de ancho de pulso es 1000-2000 μ S luego aproximadamente al 60 por ciento del rendimiento del motor la propulsión obtenida es de unos 500g lo que es suficiente para los propósitos de este proyecto. Como se puede apreciar en la gráfica a partir del 40 por ciento la propulsión crece de manera lineal.

2.2 Diseño Control

El diagrama de bloques para el movimiento de cabeceo es el siguiente:

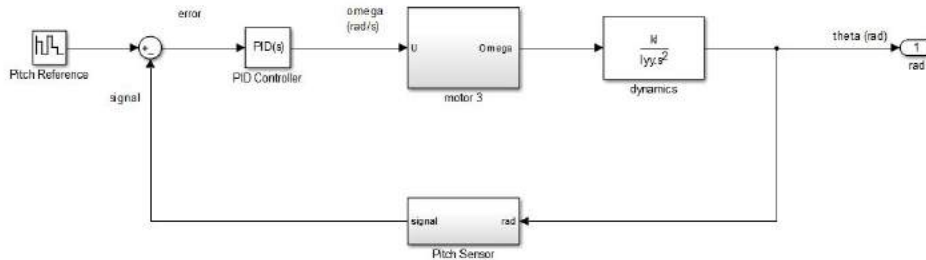


Figura 4: Pitch Control

Tras una investigación se descubrió que en la mayoría de diseños de drones caseros no se tiene en cuenta los parámetros físicos (inercia, rozamiento), de manera que no se conoce la planta. Como es imposible obtener los parámetros físicos de un drón que en el momento del diseño no había sido construido, se decide que los parámetros de los controladores PID se identificarán a partir de tests de estabilización que se realizarán sobre el drón una vez ya finalizado.

2.3 Diseño Código Control

Control y actualización de motores:

Para poder rotar sobre los ángulos de nutación y cabeceo, la idea es cambiar únicamente la velocidad en los motores indicados para conseguir esto. Para obtener elevación vertical basta con mandar la señal de *throttle* o aceleración (proveniente del piloto) a todos los motores por igual. Es decir:

```
m1 = throttle;
m2 = throttle;
m3 = throttle;
m4 = throttle;
```

Sin embargo, para poder mover un drón hacia adelante o atrás es necesario variar el ángulo de cabeceo en este caso. Bastará con cambiar la velocidad de los dos motores en el eje X (motores 1 y 3). Por el contrario si se cambia la velocidad de los motores 2 y 4 (eje Y) se logrará la rotación alrededor del eje X o de nutación. La mejor manera de cambiar la

velocidad de estos motores es sumándole un *offset* en uno y restándoselo en el otro de tal manera que la aceleración vertical sobre el eje permanece constante en media y por tanto la variación angular del cabeceo no afectará a la elevación vertical.

$$Throttle_{13} = \frac{(throttle + pitch_{input}) + (throttle - pitch_{input})}{2} = throttle \quad (1)$$

El pseudo-código está mostrado en la Fig5

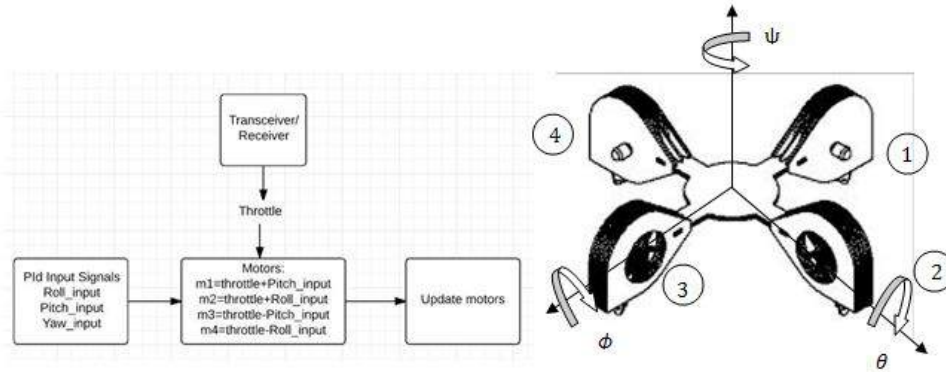


Figura 5: Actualización de motores

Los ventiladores 1 y 3 controlan el cabeceo mientras que el 2 y el 4 controlan la nutación. La rotación sobre su propio eje (Yaw) no es necesaria para el control de estabilización y por lo tanto no ha sido incluida.

3 Conclusiones

En este se proyecto ha sido posible diseñar y construir un drón capaz de volar sin hélices. El diseño utiliza un ventilador centrífugo compuesto por Poliestireno el cual es capaz de generar 550g de propulsión y que es capaz de soportar caídas y golpes sin romperse. Durante el proyecto, se produjo un ensayo en el cual un ventilador falló de tal manera que la rueda de álabes se llegó a desintegrar dentro de la carcasa de tal manera que no causó ningún tipo de daño al contrario que las hélices actuales que pueden ser muy peligrosas. Usando un micro controlador Arduino se puede interpretar la información recibida por los sensores para conseguir la estabilidad angular del drón. Sin embargo, debido a las limitaciones de energía y al tiempo requerido para el ajuste de los parámetros del PID, no se fue capaz de calibrar el control para que asegurase la estabilidad del drón durante vuelo. Otro problema encontrado fue el hecho de que las trazas utilizadas para la placa de circuito impreso no fueron lo suficientemente grandes como para soportar la intensidad requerida por los motores. Si se pudiese continuar este proyecto, se debería optimizar el diseño de los ventiladores centrífugos para que generen mayor fuerza propulsora con un diseño de ventilador más pequeño además

de explorar el uso de diferentes materiales para su construcción. También se debería rediseñar la placa de circuito impreso para que fuera capaz de conducir la corriente de los motores y así conseguir un efectivo sistema de control de corriente. Otro elemento clave a mejorar es el hecho de que los motores a usar deben consumir menos corriente o se debe de conseguir una forma de alimentación capaz de abastecer al drón para un vuelo duradero. Con un mayor tiempo de vuelo, se puede llegar a alcanzar un control de vuelo que garantice la estabilidad.

Propeller-less Multi-rotor Drone project summary

Author: Ignacio Aguirre Panadero

Project summary for ECE 445, Spring 2016

Director: Luke Wendt

12 June 2016

1 Introduction

Drones are becoming very common today in many industries. For example Amazon is starting a project to use drones for delivery. However, propellers on multirotors easily break when it comes into contact with anything hard. It tears skin when it hits flesh. Our project aims to make drones safer. Instead of using propellers, this project will use a centrifugal fan for thrust. Doing so will make the drone safer than a typical drone with expose propellers or even propeller guards on it. The exposed propeller case is obvious since the propellers will have a chance to damage anything that comes into contact with it. This project will be safer than a drone with propeller guards most propeller guards will still flex and during a crash, the propellers will hit the surface resulting in plastic shrapnel. This design will have the impeller enclosed such that people will not be able to accidentally touch the fast spinning impeller. Also, since the impeller is enclosed in the shell, it will be less likely to be damaged and shrapnel when it crashes.

1.1 Objectives

- Design a drone that flies without propellers
- Design a safer drone

1.2 Functions and Features

At the beginning of the project, the expectations were very high. The idea was to build a drone capable of flying like a commercial drone. This means an outdoor drone which would be light, portable and balanced with low power consumption and high thrust. Using state feedback control, the pilot should be able to control attitude and velocity.

At the end, the design changed to an indoor drone due to the amount of new policies and laws that restrict the use of drones outdoors. Also, flying the drone indoors avoids the trouble of wind disturbances and this fact makes possible to fly a drone with less thrust. At this point, it's pretended to build a drone capable of hovering and maintaining attitude.

2 Design

2.1 Mechanical Design

The final design was first designed as a 3D model using the program *SOLIDWORKS*. Using the 3D CAD files, the parts were cut using laser-cutting later.

Impeller Design:

A compromise between the two typical impeller designs (radial bladed and backward inclined) is an air wheel design, which is essentially a radial bladed design where each blade is very slightly inclined rearwards. This optimizes the impeller for mid-range RPM efficiency, increased flow compared to the radial blade impeller, and faster pressure build-up than a fully fledged backwards inclined impeller. The final design is shown in Fig1



Figure 1: Impeller design

Manufacture:

For the construction of the frame and impellers, foam board was chosen. The foam core board is made from polystyrene foam with paper layered on the top and bottom. Manufacturing of the pieces of the drone were done through laser cutting. Laser cutting was chosen among other techniques due to its precision and consistency. Other considerations were hand-cutting the material, but the consistency and precision of such a method would have been greatly detrimental to the projects viability.

Performance:

The way to test the maximum thrust generated by each fan is by pointing each fan down so that they push an electronic scale.

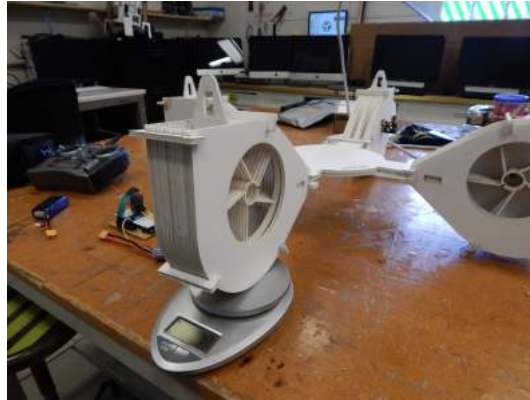


Figure 2: Centrifugal fan with impeller

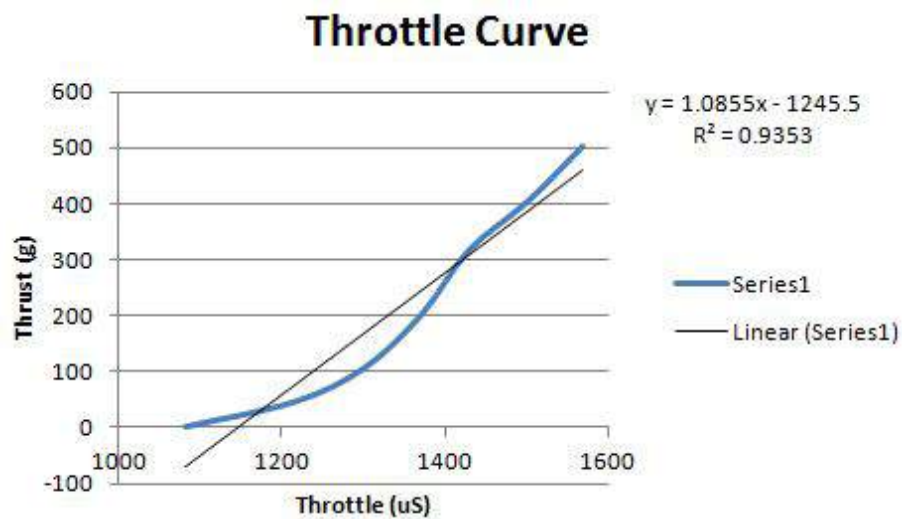


Figure 3: Thrust versus throttle of one centrifugal fan

The throttle in μs microseconds represents the pulse width of the PWM signal which controls the ESCs. Our pulse width range is 1000-2000 μs so at approximately 60 percent of throttle the thrust obtained is about 500g with is enough for the purposes of this project. As it's possible to see in the plot after 40 percent it becomes linear.

2.2 Control Design

The control schematic for the pitch can be modeled as the next one:

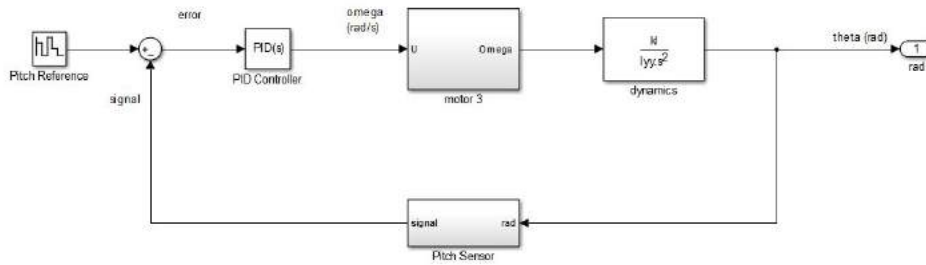


Figure 4: Pitch Control

After more research I have realized that most of the current drone's control are not design taking into account all the physical parameters like inertia, etc. Knowing the exact values of all the physical parameters will make possible to design an accurate control choosing the proper values of the PID's parameters. Due to the fact that this is not possible, we will design a control without paying much attention to the parameters which will be tuned later by doing stabilization tests.

2.3 Control Code Design

Control and motors update

In order to change the pitch and roll angles , the main idea is to change each motors speed so the quad-copter starts bending in the desired direction. Just to obtain vertical lift we just simply redirect the throttle input (coming from a remote the control) to the motors, which will just concretely change the altitude/vertical speed of the drone. The code will be:

```
m1 = throttle ;
m2 = throttle ;
m3 = throttle ;
m4 = throttle ;
```

Now, let's think a bit about how to make the quad copter move forward/ backward. This motion corresponds to the pitch angle. In order to change the pitch, we only need to change the speed of the two motors on the X axis (motors 1 and 3). Indeed, affecting the speed of the motors 2 and 4 (on the Y axis) would just start a rotation around the X axis therefore changing the roll angle.

In order to incline the aircraft there must be a difference in the motors' speed. The best way to get different motors' speed is to add the offset value on one motor output, and to subtract it on the other, so the "overall throttle " of the axis remains constant and therefore

changing the pitch won't affect the quad-copter's "vertical speed" [?].

$$Throttle_{13} = \frac{(throttle + pitch_{input}) + (throttle - pitch_{input})}{2} = throttle \quad (1)$$

The pseudo code is shown in Figure 5

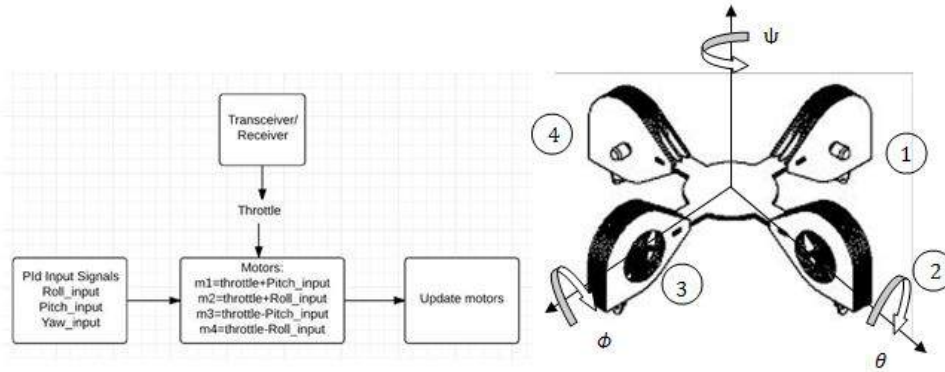


Figure 5: Updating control on motors

Fans 1 and 3 controls pitch while 2 and 4 controls roll. Yaw's control is not needed for drone stabilization and for this reason has not been included in the control.

3 Conclusion

In this project it has been able to design and construct a drone that is capable of flight without the use of propellers. Our design used a centrifugal fan composed of Polystyrene which produced a peak thrust of 550g and was capable of withstanding falls without breaking. Also, when the fan failed, the impeller shattered like safety glass and posed no dangers unlike shattered plastic propellers. Using an Arduino microcontroller as the controller, we were able to interpret sensor data from the IMU and attempted to determine stability. However, due to power supply limitations and parameter change times, we were unable to calibrate the system to achieve stability during flight. Another issue we encountered was the PCB traces not being large enough withstand the current required by the motors.

If we were to continue this project, we would optimize the fan design to output greater thrust with a smaller overall design as well as explore other materials such as composites. We will also redesign our power distribution board to be formed in layers to withstand the high current of the system. Either the motors selected must draw less current or we must get a power supply with a greater capacity for sustained flight. With a longer flight time and the right apparatus, we will be able to improve upon control and achieve stability in flight.



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA

PROPELLER-LESS MULTI-ROTOR DRONE

Author: Ignacio Aguirre Panadero
Director: Luke A. Wendt (University of Illinois at Urbana-Champaign)

Madrid
JUNIO 2016

Abstract

Propellerless Multi-rotor Drone is the idea of three electrical engineers to developed a new system of propulsion focusing on a drone that uses centrifugal fans instead of propellers. Propellers on drones are dangerous due to their high RPM typically greater than 20k RPM. When plastic propellers come into contact with a harder material, the plastic shatters leaving sharp shards and when the propeller comes into contact with a soft material like flesh, it will slice through it. The idea is to design a safer alternative to propellers on drones as well as contribute to the current drone technology. We took inspiration from the "This is NOT a Propeller" [1] video from Sam Shepard of using a centrifugal fan for propulsion. From the design we achieved a peak thrust of 550g from each fan and was able to control the drone using a Arduino microcontroller. When shattering the design broke apart like safety glass into harmless pieces of paper and polystyrene. In conclusion, we were able to explore another method of propulsion for a drone that is safer than propellers.

Acknowledgement

I wish to express my most sincere gratitude and appreciation to future doctor Luke A. Wendt for his guidance, patience and encouragement throughout the development of the project.

I would also like to thank Leo Yamamae and Bree Peng for all their work and dedication in this project.

Faculty in the Electrical Engineering Department for their assistance and help in the development of my exchange program of study at University of Illinois at Urbana-Champaign.

Contents

- 1 Introduction and Project Approach. 2
 - 1.1 Purpose 2
 - 1.2 Objectives 3
 - 1.3 Function and Features 4
- 2 Technology description 6
 - 2.1 Multi-rotor’s Operation 6
 - 2.2 What is a centrifugal fan. 10
 - 2.3 What is different in our system? 11
- 3 General Schematics 14
 - 3.1 Block diagram. 14
 - 3.2 General Circuit 15
- 4 Block descriptions and choice of components 18
 - 4.1 Weight 18
 - 4.2 Motors 19
 - 4.3 Power Supply 20
 - 4.4 Transmitter 21
 - 4.5 Receiver 22
 - 4.6 Electronic Speed Controller 22
 - 4.7 Centrifugal Fan. 23
 - 4.8 Controller 23
 - 4.9 IMU sensor 23
 - 4.10 Actuator 24
- 5 Mechanical Design 26
 - 5.1 A/R Ratio 26
 - 5.2 Materials 30
 - 5.3 Manufacturing. 30
 - 5.4 Prototypes 30
 - 5.5 Final Design Centrifugal Fan 32

6	Code Design	36
6.1	Flow Chart	36
6.2	Control State	37
6.3	Communication.	37
6.4	Motor Control.	38
7	Custom Hardware Electrical Design	40
7.1	Circuit Overview	40
7.2	LED Module	40
7.3	Voltage Alarm.	40
7.4	Current Sensor	41
8	Control design.	44
8.1	Dynamics.	44
8.2	State Equations	47
8.3	Linearization	48
8.4	Equilibrium point.	50
8.5	Obtaining the Laplace transform	51
8.6	SISO system	51
8.7	Control.	52
8.8	PID controller.	54
9	Cost Analysis	58
9.1	Part costs	58
9.2	Labor costs	58
9.3	Grand total cost	59
10	Ethical / Legal Considerations.	62
10.1	IEEE Code of Ethics Considerations.	62
10.2	Legal Obligations	62
10.3	Safety Statement	63
11	Conclusion.	66
	Appendix A Requirements and Verification.	68

Appendix B	Circuit Schematic	74
Appendix C	Code Snippets	80
Appendix D	CAD models and schematics	84
Appendix E	Sources	94

1 Introduction and Project Approach

1.1 Purpose

Drones are becoming very common today and can be seen in different businesses, used for recreational purposes and even for defense sectors in the US. For example Amazon has a project to deploy drones for delivery tasks [2]. One major problem with current drones is its durability and danger. Propellers on drones easily shatter when it collides with a hard surface and propellers can easily tear skin. There are some different ways to combat this issue, for example, one can attach a propeller guard which shields the propellers from the sides but not the top or the bottom as we can see in Figure 1. However, using the guards will not prevent the propellers from shattering when the drone crashes nor will it prevent unlucky people from slicing their fingers off.

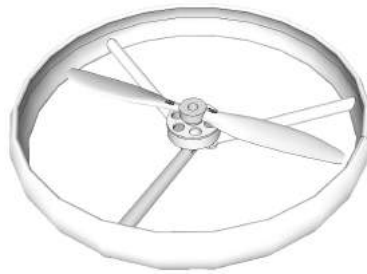


Figure 1: Prop-guard commonly used on drones

The project aims to make drones safer by utilizing centrifugal fans instead of propellers. The use of a centrifugal fan is a different and new method to generate thrust without the use of propellers. By nature, the fan is enclosed in an outer shell that will shield potential victims from the impeller.

This project pretends to make a newfangled drone from scratch assuming all the difficulties that can appear on the design of something completely new. This project comprehends mechanical, electrical and control challenges. The challenges begin with the design of a fan capable of providing the amount of thrust needed, in addition to the difficulties of providing enough energy to power the motors and the complexity of controlling the system to achieve an appropriate flight experience.

1.2 Objectives

The objectives of this project can be separate in three main categories: mechanical, electrical and control.

Mechanical

The objective is to design all the mechanical components needed for the quad-copter in 3D CAD. After that, 3D files are going to be used to build the real components using laser-cutting techniques. The mechanical portion comprehend the next objectives:

- Design a centrifugal fan capable of providing the thrust needed to fly our homemade quad-copter. This includes the housing and impellers which conforms the fan.
- The choice of a material light and strong enough and appropriate for laser cutting.
- Design the lightest drone possible. Taking into account the different parts of a drone like the frame, the electronic components and the fans.

Electrical

The objective is to find the motors and battery required to provide enough thrust and power to fly the homemade drone. Additionally, the inclusion of a PCB design which functions are to provide orientation and control peak currents. The electrical portion comprehend the next objectives:

- The choice of Brush-less motors capable of providing enough speed and torque to compensate the resistant torque caused by the impeller
- The choice of a battery capable of powering 4 motors and the rest of the electronic components during a reasonable time.
- The design and implementation of a PCB hardware which objectives are to provide orientation and to control the maximum currents.

Control

The control must assure a good flight experience to the pilot. For this reason, a study of the dynamics of a quad-copter must be carried out in order to obtain the transfer functions that rules the movement of a drone.

- The study of the dynamics of a drone obtaining a final SISO system.

- To obtain the values of PID's parameters.
- To implement the control code on the Arduino.

1.3 Function and Features

At the beginning of the project, the expectations were very high. The idea was to build a drone capable of flying like a commercial drone. This means an outdoor drone which would be light, portable and balanced with low power consumption and high thrust. Using state feedback control, the pilot should be able to control attitude and velocity.

At the end, the design changed to an indoor drone due to the amount of new policies and laws that restrict the use of drones outdoors. Also, flying the drone indoors avoids the trouble of wind disturbances and this fact makes possible to fly a drone with less thrust. At this point, it's pretended to build a drone capable of hovering and maintaining attitude.

2 Technology description

2.1 Multi-rotor's Operation

A multi-rotor is a device which its thrust is produced by the rotation of multiple equidistant rotors from the geometric center. With different rotation speeds, it's possible to achieve a great maneuverability.

In multi-rotor's group, there are different subgroups depending on the number of rotors. There are from tri-rotors until octo-rotors. Obviously, as the number of rotors increase, the stability improves and therefore the thrust.



(a) Tri-rotor



(b) Octo-rotor

Figure 2: Example of common multi-rotors

An aircraft in flight is free to rotate in three dimensions: pitch, nose up or down about an axis running from wing to wing; yaw, nose left or right about an axis running up and down; and roll, rotation about an axis running from nose to tail. The three angle of rotation are roll, pitch and yaw and those angles in mechanics are known as the Euler-angles[3].

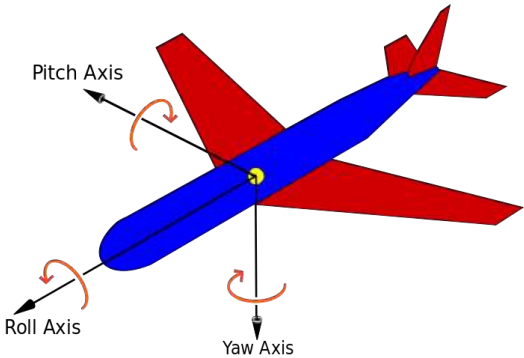


Figure 3: Take off and landing

In our case, this project is going to be focus on the case of quad-copters which flight dynamics are gonna be explain in the next lines[4]:

Vertical movement:

Using a quad-copter, the aircraft must have all four motors turning at the same speed to achieve smooth vertical take-offs and landings assuming that the weight of the quad-copter is evenly distributed across the frame.

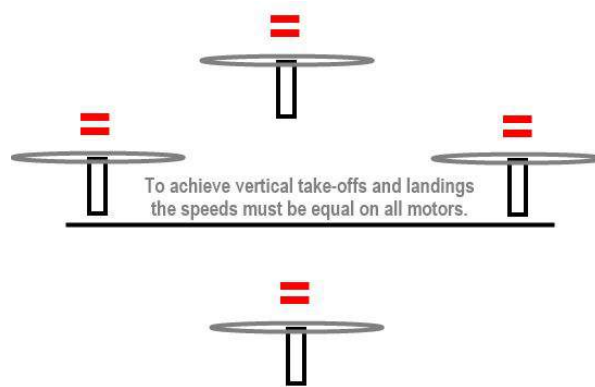


Figure 4: Take off and landing

Roll

Pitch and Roll rotation is controlled by increasing the speed of two motors on one side while slowing down the speed of the motors on the opposite side. For example, to roll left the motors on the right side of the frame must increase in speed and the motors on the left must decrease in speed. To roll to the right the opposite must take place, the motors on the left side of the frame must increase in speed and the motors on the right must decrease at the same time. Doing this we are creating a torque across the roll axis.

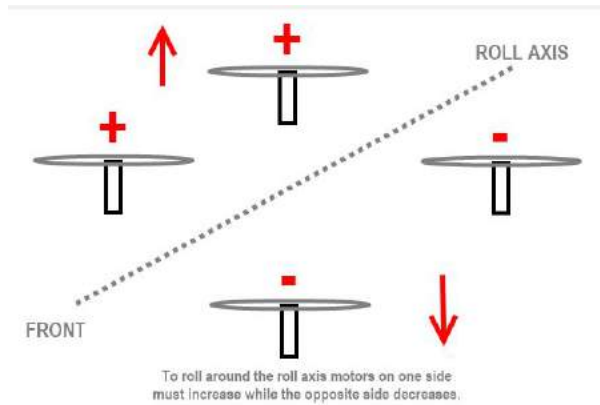


Figure 5: Roll

Pitch

In the same fashion, to pitch forward the two motors at the rear must increase in speed while the two motors at the front decrease and vice-versa for a backward pitch. In order for a quad-copter to fly forward the speed of the motors at the rear of the aircraft must be faster than the speed of the motors at the front which will cause the quad-copter to "lean" forward or roll around the pitch axis. Once in this position the quad-copter will fly forward when the speed of all four motors is increased, increasing the overall thrust. Generally, the greater the "lean" forward, the faster the quad-copter will travel. The same is true for traveling in reverse. To move backwards, the speed on the front motors must increase to cause the aircraft to "lean" backwards. Once the quad-copter is leaning backwards, the speed of all four motors is increased, increasing the overall thrust and the quad-copter flies backwards. Again the greater the "lean" the faster it travels.

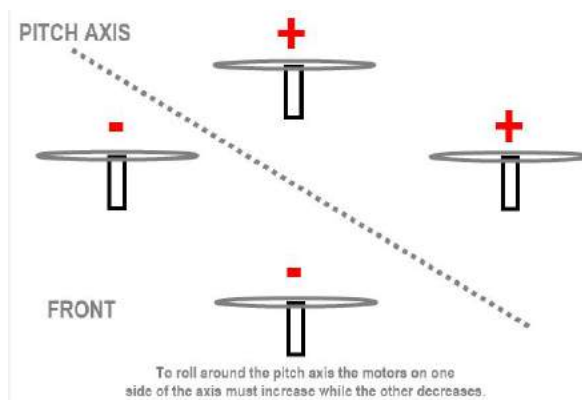


Figure 6: Pitch

Yaw

The quad-copter rotates over himself clockwise or counterclockwise by increasing the speed of two motors that are diagonally across from one another. Doing this, torques are compensated in order to maintain the drone leveled but there is an angular momentum across the yaw axis that produces rotation.

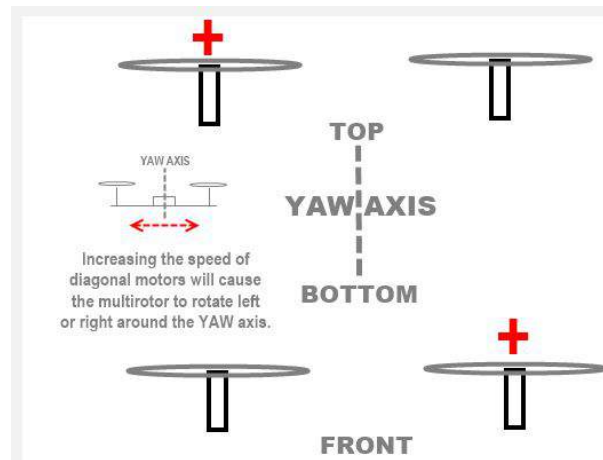


Figure 7: Yaw

On a quad-copter two motors diagonally across from one another must turn in one direction (clockwise) while the other two motors turn in the opposite direction (counterclockwise) in order to fly. This clockwise and counterclockwise is because of the angular momentum of the motors. A good example is a helicopter. It has its huge propellers and it has a tail-rotor. Without the tail-rotor, the helicopter will be flying like a spinning top. The tail-rotor is there to conserve momentum by applying a torque of the helicopter.

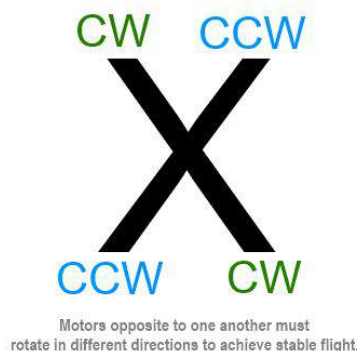


Figure 8: Yaw

2.2 What is a centrifugal fan

A centrifugal fan is a mechanical device for moving air increasing the speed of air stream with rotating impellers. It uses the kinetic energy of the impellers to increase the pressure of the air stream which in turn moves them against the resistance caused by ducts, dampers and other components. Centrifugal fans accelerate air radially, changing the direction, typically by 90° , of the airflow.

It has a fan wheel composed of a number of fan blades, or ribs, mounted around a hub. The hub turns on a drive-shaft that passes through the fan housing. The air enters from the side of the fan wheel, turns 90 degrees and accelerates due to centrifugal force as it flows over the fan blades and exits the fan housing[5].

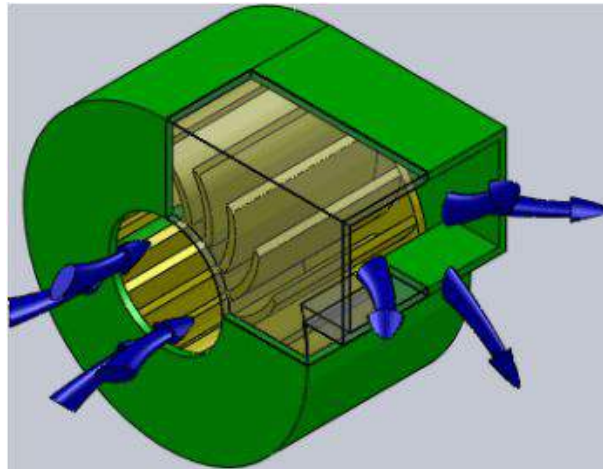


Figure 9: Centrifugal Fan

2.3 What is different in our system?

The idea is to have four centrifugal fans in each corner replacing the motor plus propeller configuration seen in most of the typical drones. You may ask, is this the same as just swapping out methods of propulsion? The answer is no.

The problem that arises from changing to this method of propulsion is that the drone cannot yaw (turn left or right). The traditional drone increases the RPM on two motors spinning the same direction located diagonally from each other and decreases the RPM of the other two. This will result in keeping altitude and turning left or right by angular momentum. A drone also rolls by changing the RPM of two motors but this time on the same side. The issue with drones that uses centrifugal fans is that in order to yaw, the direction of the fan must be changed. Therefore, we will need to use an actuator like a servo to redirect the air flow.

The ultimate goal is to build a drone that uses a centrifugal fan as a method of propulsion and map the controls such that an experienced multi-rotor flyer will be able to pick up the controller and fly it. So it will be a remote-controlled drone that will be using a typical RC transmitter and receiver.

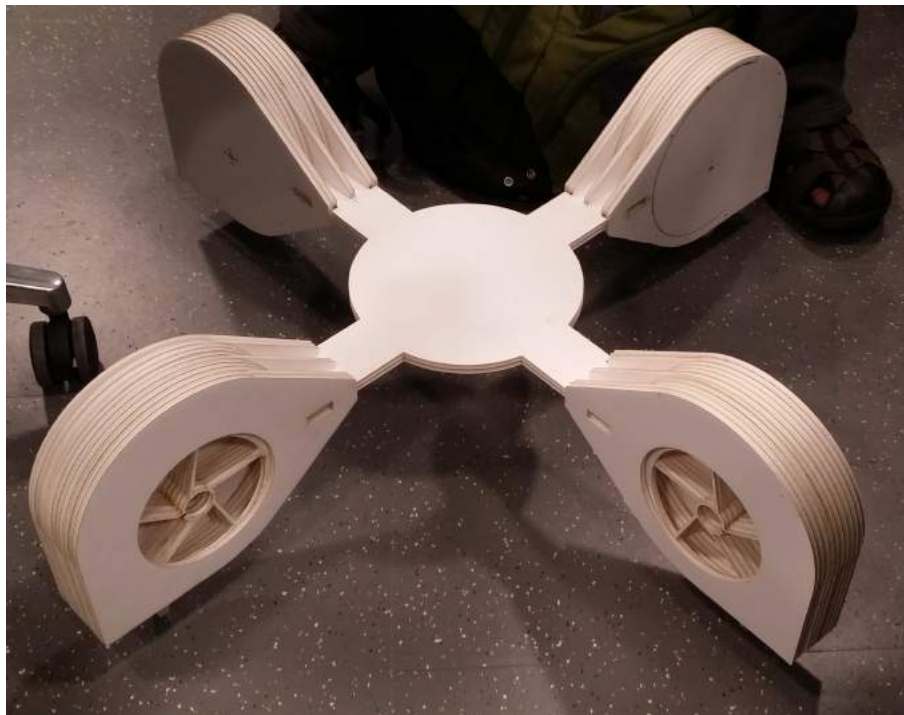


Figure 10: Homemade centrifugal fan quad-copter

3 General Schematics

3.1 Block diagram

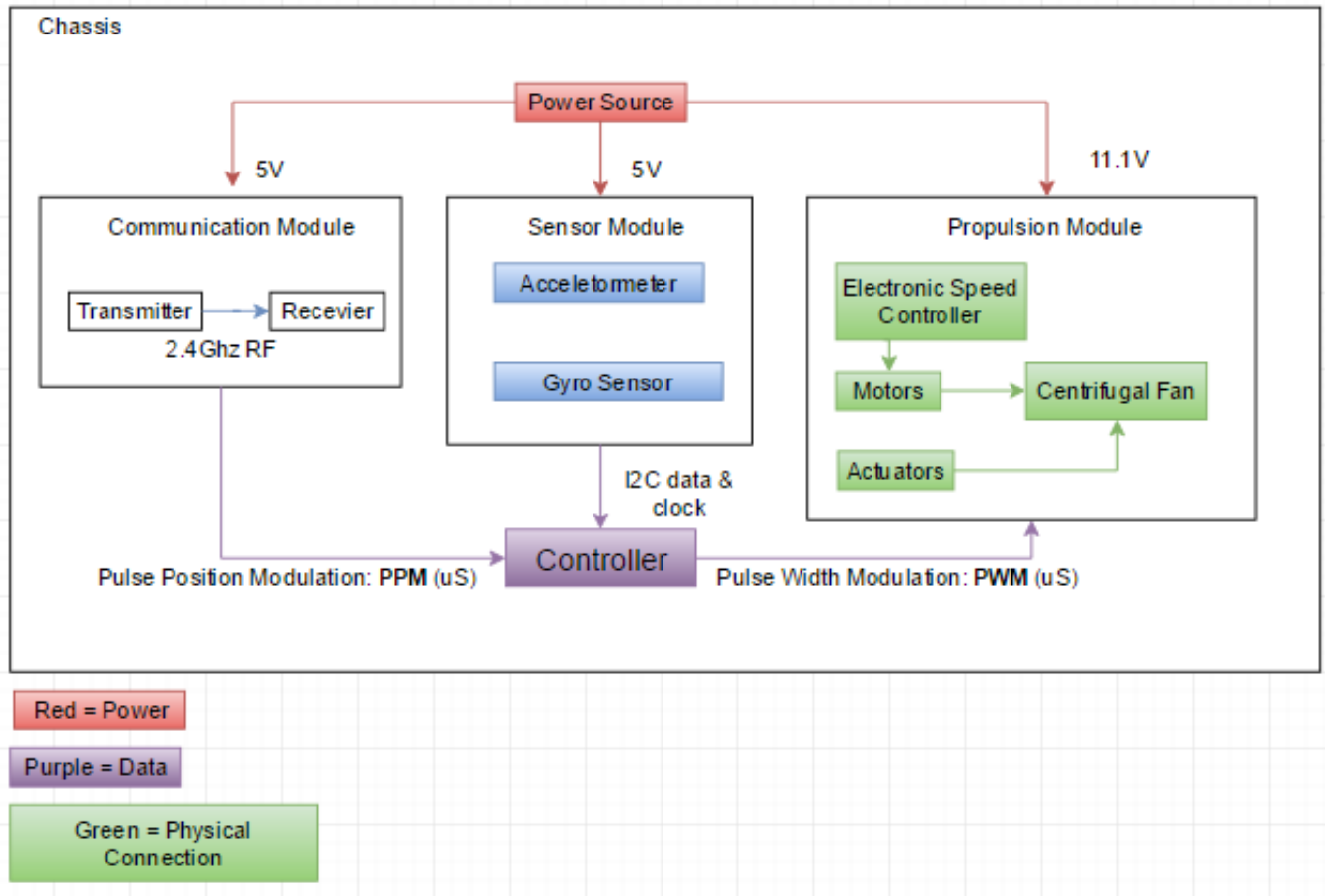


Figure 11: Block Diagram with the modules

The figure shown in Figure 11 is a general block diagram which shows the main modules of the project. The main modules are Power, Communication, Sensor, Propulsion, and Controller.

3.2 General Circuit

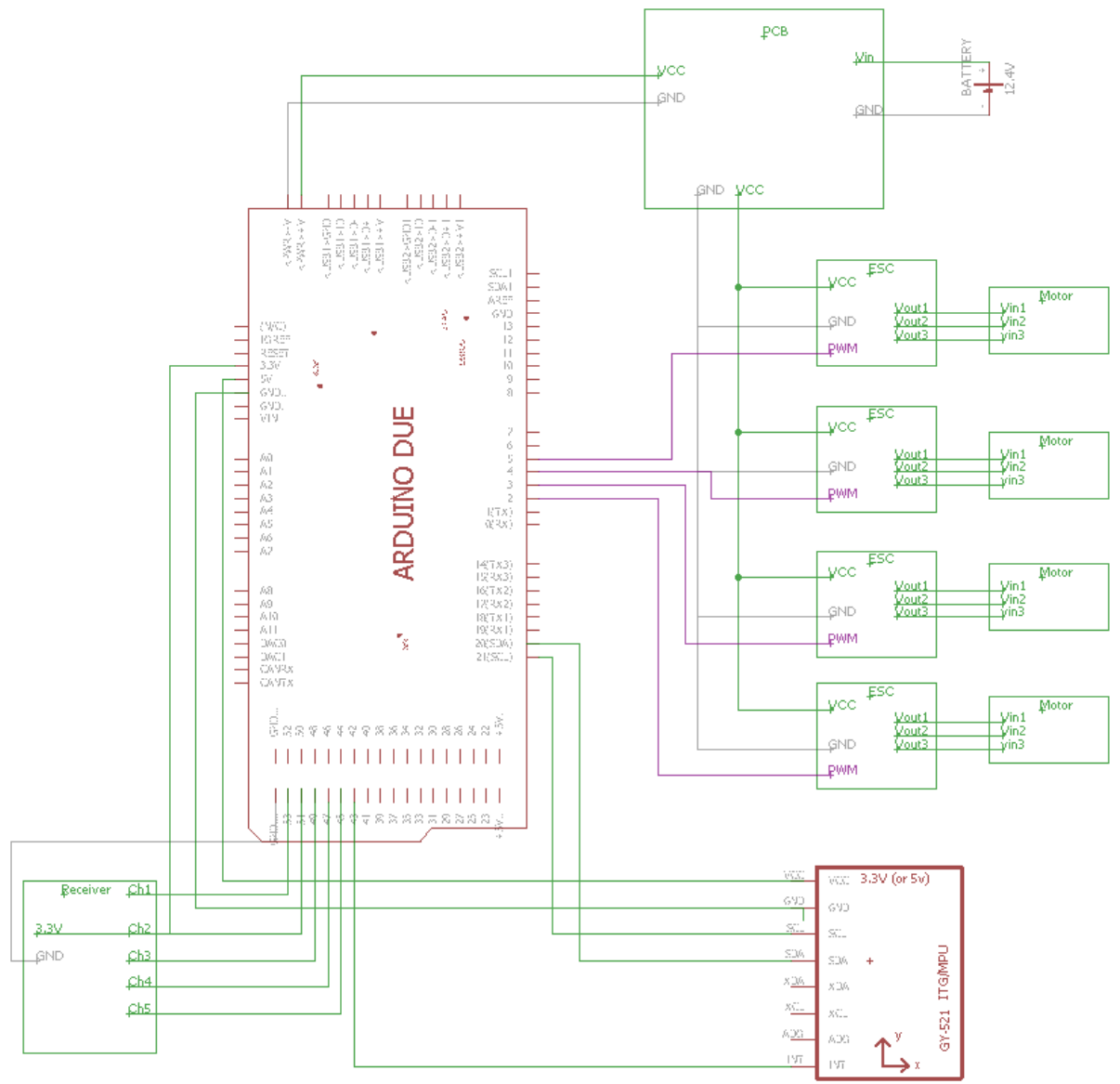


Figure 12: Genera Schematic

The general schematic shown in Figure 12 shows the overview of components that will be connected together.

The digital PWM pins 2, 3, 4 and 5 are connected to the PWM inputs of the ESCs. For the sensor's interrupt whatever digital pin can be used. For I2C bus communications between the Arduino and the sensor pin 20(SDA) and pin 21(SCL) have been used.

4 Block descriptions and choice of components

In order to choose the right components when designing a quad-copter that uses propellers, there is an order to choose the components. Firstly, the weight of the frame must be known in order to choose the motor-propeller couple capable of providing a thrust/weight ratio of at least two (common ratio on drones, it allows a vertical acceleration of one G). Once you know the motor used, the battery can be chosen.

However, our case was a trial and error method as at the beginning we did not know how much thrust our fan could generate. This fact conditionates the size and weight of the drone. For this reason, we have probably oversize the motors in order to make sure that they can provide enough thrust and resist the resistant torque of the impellers.

4.1 Weight

The first principle to success in the design of a drone is to make it as light as possible. Ordinary mid-size drones commonly weight around one kg (with all the components). Due to centrifugal fans are more complex and weightier than propellers, we expected more than one kg. On the other hand, the supposition was that a centrifugal fan could provide more thrust at the same speed than a propeller.

Here it is possible to find a table with the weight of each component and the total weight of the drone:

Table 1: Total Weight

Item	Weight (g)
Frame	250
Housing (x4)	54 (216)
Impeller (x4)	16 (64)
Motor (x4)	97 (388)
ESC (x4)	48 (192)
Perf Board	84
Battery	202
PCB	29
Sensor	5
Total	1430

Let's estimate how much thrust must generate every motor plus fan to obtain lift. For this calculation, we need to know the weight of our final design which is 1430g. In the drone mass, we are including all the components like motors, ESCs or the drone frame.

$$m_{drone} = 1430g$$

T_{motor} = thrust of the motor

$$T_{motor} = \frac{m_{drone}}{4} = \frac{1430}{4} = 357.5g \quad (1)$$

But with this calculation we are just compensating the gravity force so to achieve at least an acceleration like the gravity we have to achieve the double of thrust.

$$T_{motor} = \frac{m_{drone}}{4} 2 = \frac{1430}{4} 2 = 715g \quad (2)$$

This ratio of $\frac{Thrust}{Weight} = 2$ it's just realistic for drones using propellers. As our drone just is pretended to hover (flying close to ground) indoors, it's not necessary such a huge security margin. For these purposes, a ratio of 1.3 should be enough.

$$T_{motor} = \frac{m_{drone}}{4} 1.3 = \frac{1430}{4} 1.3 = 464.75g \quad (3)$$

It will be easier to choose a motor plus propeller if we were using an ordinary drone because the thrust obtained by the ensemble is provided in the manufacture datasheet. In our case, we will be using the data from the prototype tests where we achieved a maximum thrust of near 300g with a 4000kV motor.

4.2 Motors

The motors are the elements that will allow the machine to move. Large brushless out-runner motors with a KV rating of greater than 2670. KV rating is the RPM generated per volt applied to the motor. Therefore, if we apply 11.1V to a 3400kV motor, the motor will be spinning at 44400 RPM. Brushless motors are chosen due to its light weight, lower operating costs, lower friction, higher efficiency, and high power they provide. This motor will need to be able to handle a great amount of torque for it to not burn during operation of the drone. For this reason, the NTM Rotor drive 450 Series 3400kV/500W motor is chosen.



Figure 13: NTM Rotor drive 450 Series 3400kV/500W

This motor pulls from 30A continuous and 46A burst. From these values, we can calculate the amount of current the power supply must provide.

$$Current_{required,nomial} = 4motors * 30A = 120A \quad (4)$$

$$Current_{required,burst} = 4motors * 46A = 184A \quad (5)$$

4.3 Power Supply

The power supply distributes power for all components in the system. A Lithium Polymer Battery (LIPO battery) has been chosen for our project due to its high Power/Weight ratio. Since the motors selected for our project draws a nominal current of 30A, the battery must be able to supply at least 120A for the 4 four motors used in the project. Then we need to choose a battery with a discharge rate that allows run every component at his nominal current.

$$I_{battery} = 4I_{motor}(A) + Components(mA) \approx 4I_{motor}(A) = 120A \quad (6)$$

The specific battery chosen for this project is the Turnigy Bolt 2200mAh 3S 65-130C 25.08Wh Lithium Polymer battery. This battery operates at 3S equivalent to 11.1V with a capacity of 2200mAh and a 65-130C discharge rate.

We can calculate the amount of current this battery can provide by multiplying the capacity by its discharge rate. The following is the calculation performed:

$$Current_{nominal} = 2200mAh * 65C * \frac{1A}{1000mAh} = 143A \quad (7)$$

$$Current_{burst} = 2200mAh * 130C * \frac{1A}{1000mAh} = 286A \quad (8)$$



Figure 14: Turnigy Bolt 2200mAh 3S 11.4V

Now let's calculate how much time would last the battery at maximum load:

$$t_{bat} = \frac{2200mAh}{143A} \frac{1A}{1000mA} \frac{3600sec}{1h} = 55.38sec \quad (9)$$

This is one of our biggest issues, a flight time of 55.38 seconds due to the high current consumption of the motors, is too short. After a market research, there were no better options than the battery chosen because bigger batteries were way too heavy and expensive. One possible solution to this issue is to attach the drone to a constant electric supply through a long wire but this is not an easy solution.

4.4 Transmitter

The transmitter allows an operator to send signals to the receiver using pulse position modulation or PPM for short. For our project, we chose the Spektrum Dx6i as our transmitter. The Spektrum Dx6i transmitter uses the 2.4Ghz band and the DSM2 modulation[6]. The Dx6i supports up to 6-channels where 4 of the 6 is required for the drone movement, and the other 2 channels may be used for expanding the project or for safety features. The Dx6i was chosen for this projects as it provides a reliable communication to the receiver and is a similar type of controller used for typical drone and plane operation.



Figure 15: Transmitter Dx6i

4.5 Receiver

The receiver is the component that will take in the signals from the transmitter. The receiver must be able to receive at least 6 channels of data in order to maximize our drone's potential. For our project, we chose the Spektrum AR600. The AR600 operates on the DSM2 modulation as well as the 2.4GHz Band[7]. The receiver also supports up to 6-channels and is compatible with the Dx6i for our project.

4.6 Electronic Speed Controller

The electronic speed controllers (ESC) are the components that are calibrated to a range of PWM (Pulse Width Modulation) or PPM (Pulse Position Modulation) signals, and uses these to determine the speed at which they should be driving the motors. Each must have current-handling capabilities that include a safety factor in order to properly drive each motor on the drone without jeopardizing the integrity of the overall control system. For this project, we will be using the HobbyKing SS Series 50-60A ESC. This ESC is able to control the speed of a brushless motor and can handle up to 50A continuous and 60A. This ESC is perfect for our project because it has many features included like overheating protection as well as a smooth linear throttle response and a auto-shutoff when signal is lost.



Figure 16: ESC

4.7 Centrifugal Fan

The centrifugal fan is the key component in the project. The centrifugal fan gives the project its uniqueness. This and the use of the servo creates an alternative propulsion method for drones of this size. The centrifugal fan is similar to an automotive intake compressor found in cars equipped with turbines or belt-driven compressors. It has a circular hole on its large face which takes in air and then a spinning circular blade compresses the air and pushes it out the nozzle on the side of the round housing. This part will be custom-made. The part will be made with polystyrene impeller and housing. These materials are chosen mainly due to their weight.

4.8 Controller

The controller is the brain of the drone. The controller will use signals from the receiver and I2C sensor bus to compute the necessary values for the speed controller's inputs. The controller we chose for our project is the Arduino DUE. The Arduino DUE has 12 PWM pins which is used to send signals to the speed controllers and has a clock time of 84MHz.[8].

4.9 IMU sensor

For this project the sensor chosen is the MPU6050 with 6DoF(3 gyroscope axis+3 accelerometer axis) on the breakout board GY-521. The sensor uses the I2C-bus to interface with the Arduino.

To obtain accurate angles a complementary filter algorithm must be used to combine accelerometer and gyroscope measurements. Accelerometer data is noisy on short time scales, and gyroscope data drifts on longer timescales, so the complementary filter combines both for greater accuracy. However, the MPU-6050 contains a Digital Motion Processor (DMP)

which can perform the data fusion on the IMU chip itself and reports the orientation in yaw/pitch/roll or quaternion format. Though the exact details of how the DMP does this calculations are not published, the DMP is still useful because presumably InvenSense has a pretty good fusion algorithm and because the DMP calculations occurs on the MPU-6050 chip, freeing up processor power. For this code we have used the libraries MPU6050.h and I2Cdev.h.

The code is based on Jeff Rowberg work MPU6050 DMP[9].



Figure 17: MPU6050 with 6DoF on the breakout board GY-521

4.10 Actuator

The actuator is the component that is used to angle the centrifugal fans in order to yaw the drone. This is most-likely be a brushless or core-less servo motor due to its power to weight ratio requisites. The servo will take in a PWM signal and 5V from the controller. The servo will be able to turn 125 degrees of rotation and should be able to output 20oz-in of torque. For this project, we will use the Sparkfun generic sub micro-size servo since we will use the servos to turn the shutters on the centrifugal fan's output port to direct the air flow.

5 Mechanical Design

The diagrams and images for the Mechanical Design portion was taken from "Information and Recommendations for the Engineer Fe-2400" by Clarage[10].

5.1 A/R Ratio

An A/R ratio is the ratio of the cross sectional area of the path that air travels versus the radius of the centerline of its path. As the cross sectional area of the compressor housing increases, the total volume of air that can flow through that area also increases. As the radius of the housing increases, the total volume of air contained along the flow path increases as well.

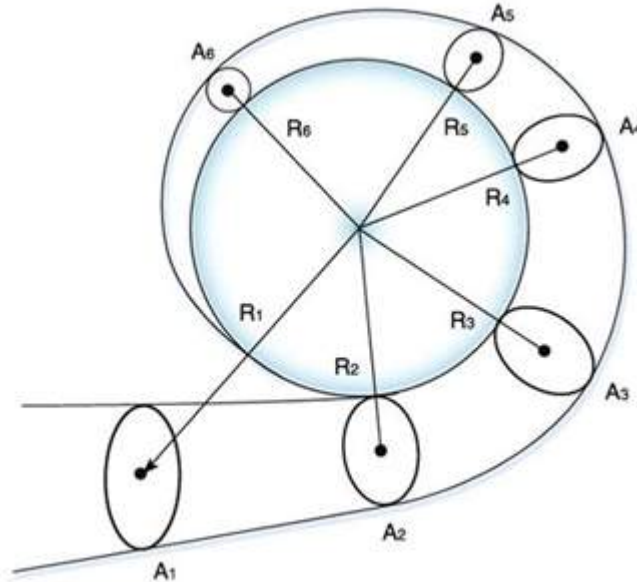


Figure 18: Image of Cross sectional Area vs Radius

Slower response in the centrifugal fan design directly correlates with the size of the fan. Large impellers with greater mass accelerate at a slower rate than smaller impellers with low mass and therefore lower inertia. Larger impellers have a higher air flow than smaller impellers, however they require larger housings, which takes longer to pressurize due to the increased air volume of the housing. Thrust in a centrifugal fan design correlates with flow rate of the fan. A high flow rate means greater thrust. However, it is important to consider

the responsiveness of the fan, as a fan that is too greatly optimized for maximum thrust will be very slow to respond. The size increase radially of a centrifugal impeller will also increase load and energy requirements exponentially as rotational kinetic energy is governed by

$$KE_{rotational} = \frac{1}{2}I\omega^2 \quad (10)$$

where I is the moment of inertia of the impeller, and ω is the angular velocity of the impeller. The equation shows that as the angular velocity of the impeller increases, the kinetic energy of the impeller increases by a 2nd order, and as mass of the impeller increases, the energy requirements increase proportionally as well.

The impeller blade profile greatly affects the characteristics of a centrifugal fan.

A radial bladed centrifugal fan will have peak efficiency earlier, and will reach its pressure and flow peak sooner than a backwards inclined impeller, however will exhibit severe diminishing returns with increases in RPM. This is due to the blunt blade profile with respect to its direction of motion. The flat, perpendicular surface does not impart deflection momentum on the air, but relies on the pressure difference of the fast-moving air at the blade tips versus the slow moving air in the center of the impeller. This design, however, generates a lot of turbulent flow, which in turn, increases its aerodynamic drag significantly.

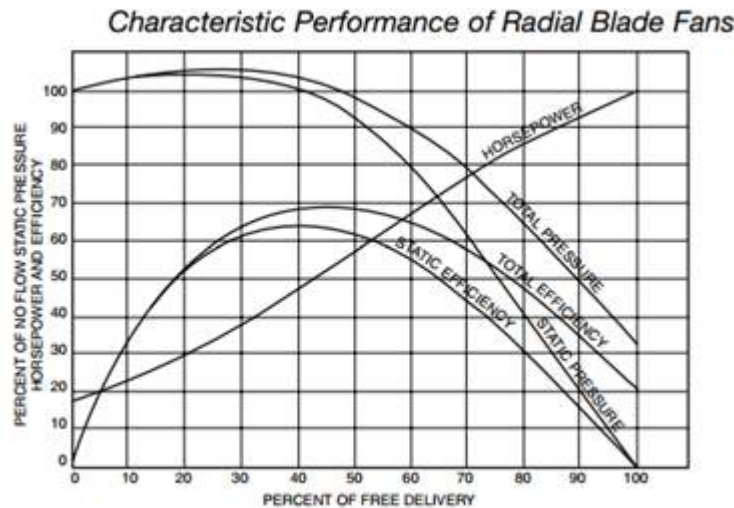


Figure 19: Performance of Radial Blade Fans in graph form

A backward inclined centrifugal fan will have peak efficiency at a higher RPM, but will reach its pressure peak slower than a radial fan. The slanted blade profile imparts deflection momentum to air particles as well as relying on pressure differences. However due to the deflection, the air particles at the tips of the blades leave the rotation of the impeller before reaching high velocity, so the pressure difference is less effective in moving air outwards. Thus, as the RPM increases, more of this pressure difference comes into effect in addition to the deflection, resulting in high flow, and an overall higher RPM peak efficiency. The downside to this, is that it is energetically draining to operate at higher RPM. This is especially important to consider because as the flow velocity of air increases, the force of drag on the impeller will also increase exponentially.

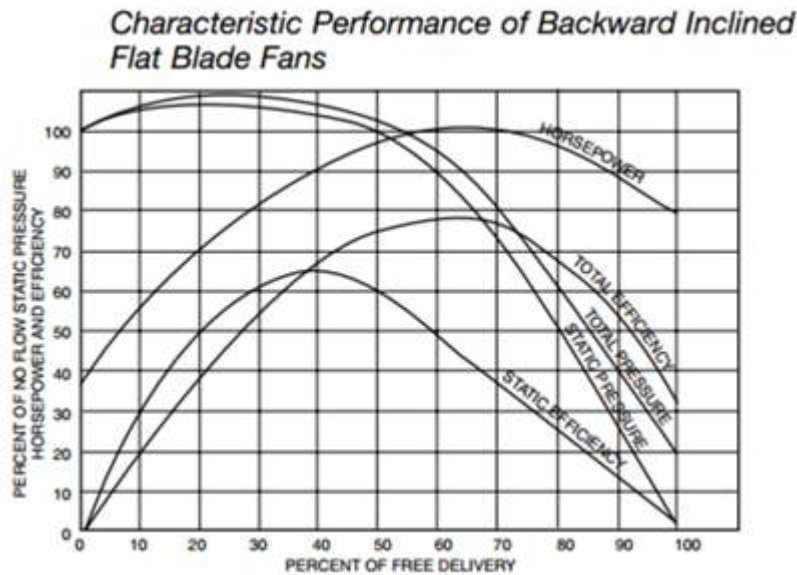


Figure 20: Performance of Backward Inclined Flat Blade Fans in graph form

$$F_D = \frac{1}{2} \rho u^2 C_D A \quad (11)$$

Where u is the flow velocity relative to the object, C_D is the coefficient of drag, A is the cross sectional area of the object, And ρ is the density of air.

A compromise between these two designs is an air wheel design, which is essentially a radial bladed design where each blade is very slightly inclined rearwards. This optimizes the

impeller for mid-range RPM efficiency, increased flow compared to the radial blade impeller, and faster pressure build-up than a fully fledged backwards inclined impeller.

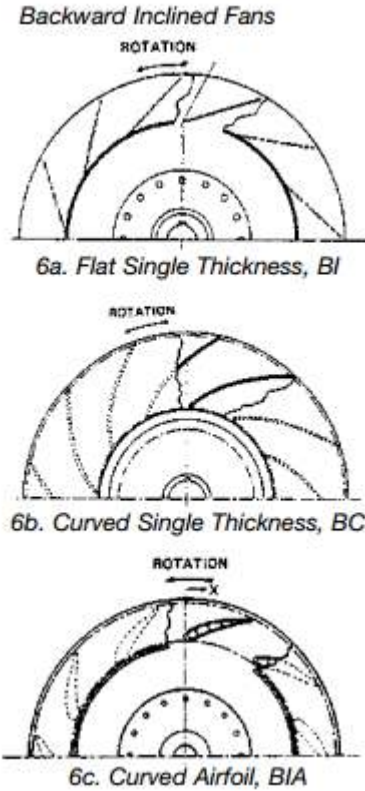


Figure 21: Images of Backward Inclined Fans of different types

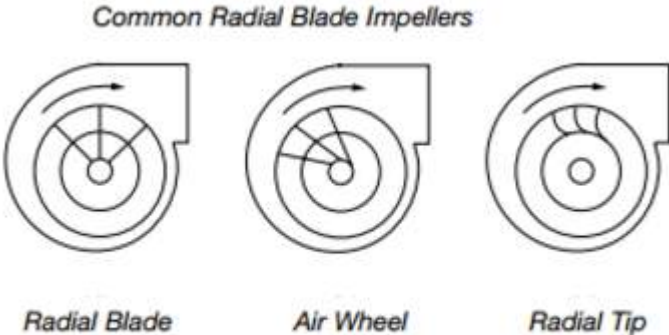


Figure 22: Radial Blade Fan Designs

5.2 Materials

For the construction of the frame and impellers, foam board was chosen. We considered corrugated paper board and composites, but neither was appropriate for the application. Corrugated paper board did not have the stiffness desired, and composites were much too expensive. The foam core board is made from polystyrene foam with paper layered on the top and bottom. This creates a stiff, low-density layered material. It is very lightweight. The material exhibits brittle fracture characteristics which are desirable when considering impact with human appendages or other foreign material for safety purposes. The material undergoes fragmentation when dispersing impact energy, and this fragmentation does not require a high impact to occur. The material is moderately inexpensive, and easy to shape. Corrugated paperboard is a common material for boxes. It is less stiff than foam core, and it is tougher as it does not undergo brittle fracture or fragmentation. However, the bending of the material in response to stress beyond its elastic deformation point increases the amount of time it can apply force to a foreign object, and diminishes the energy dispersal of the impact. This leads to an increase in risk of injury if an impeller comes in contact with an appendage while spinning at high rpm. This material is very inexpensive, and about as easy to shape as the foam core. Composites such as fiberglass, carbon fiber, and kevlar are very stiff as well as very lightweight. They also exhibit brittle fracture, though they do not exhibit fragmentation as foam core does. They have a very high force threshold for elastic deformation, which is likely to guarantee injury should an appendage come in contact with an impeller of this material spinning at high rpm. Additionally, the expense of procuring, manufacturing, and finishing parts utilizing these materials is too great of a burden for our proof-of-concept application.

5.3 Manufacturing

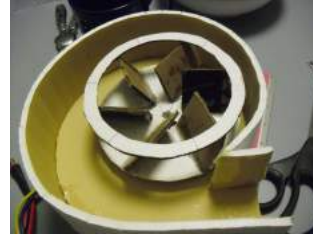
Manufacturing of the pieces of the drone were done through laser cutting. We decided on laser cutting due to its precision and consistency. Other considerations were hand-cutting the material, but the consistency and precision of such a method would have been greatly detrimental to the projects viability.

5.4 Prototypes

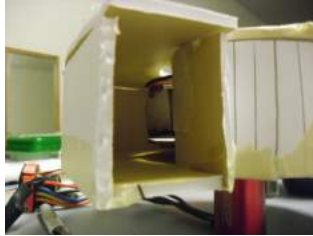
Before starting with the CAD 3D model, some homemade prototypes were made to check which configuration performed the best. The best performance was obtained with a prototype made with polystyrene foam.



(a) Overall view



(b) Impeller and housing detail



(c) Output detail



(d) Motor junction detail

Figure 23: Centrifugal Fan Prototype

Performance:

In Figure 24 it is possible to see the comparative in thrust performance between a propeller and our prototype using different batteries and motors. Propellers and centrifugal fan can not use same motors due to the high resistant torque of the impellers. In fact the motor used for the fan is a helicopter's motor instead of a drone's motor.

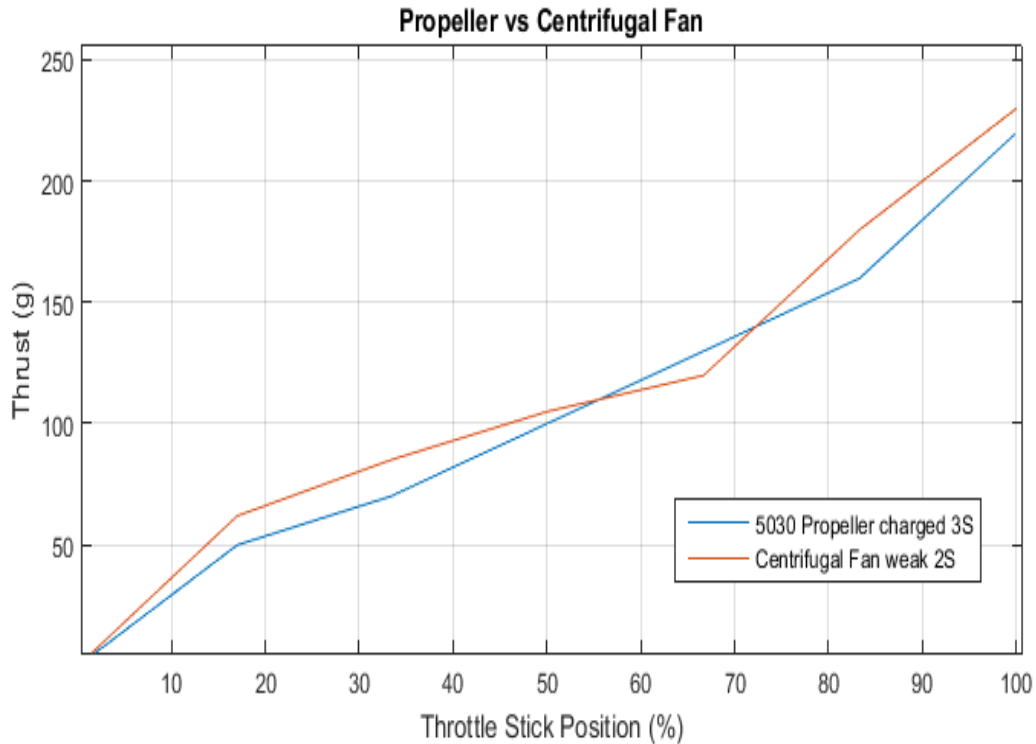


Figure 24: plot for throttle stick vs thrust for the original 5030 propeller

Despite of the fact that the motor used in the fan is much more powerful, it can be observed in the plot that around 75 percent of maximum throttle the fan generates more thrust than the propeller using a battery that is not fully charged. This was the proof to trust in the viability of this project.

5.5 Final Design Centrifugal Fan

The final design was first designed as a 3D model using the program *SOLIDWORKS*. Using the 3D CAD files, the parts were cut using laser-cutting later. The 3D models and 2D figure plans can be found on the Appendix D



Figure 25: Impeller

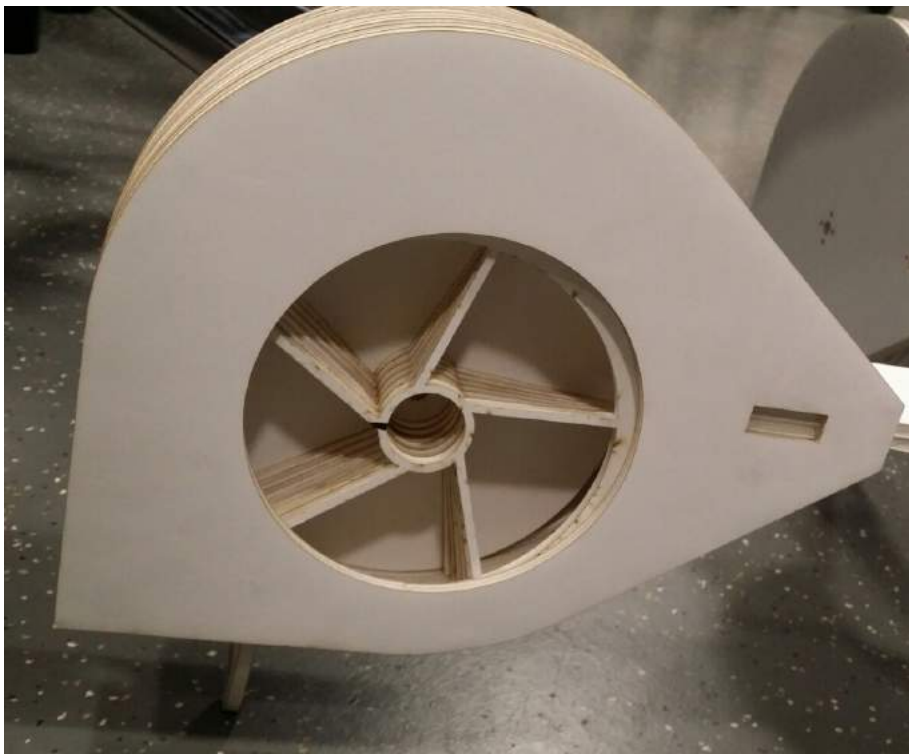


Figure 26: Centrifugal fan with impeller

Performance:

The way to test the maximum thrust generated by each fan is by pointing each fan down so that they push an electronic scale. In Figure 27 is shown this method.



Figure 27: Centrifugal fan with impeller

The next plot shows the maximum thrust of one fan versus the throttle.

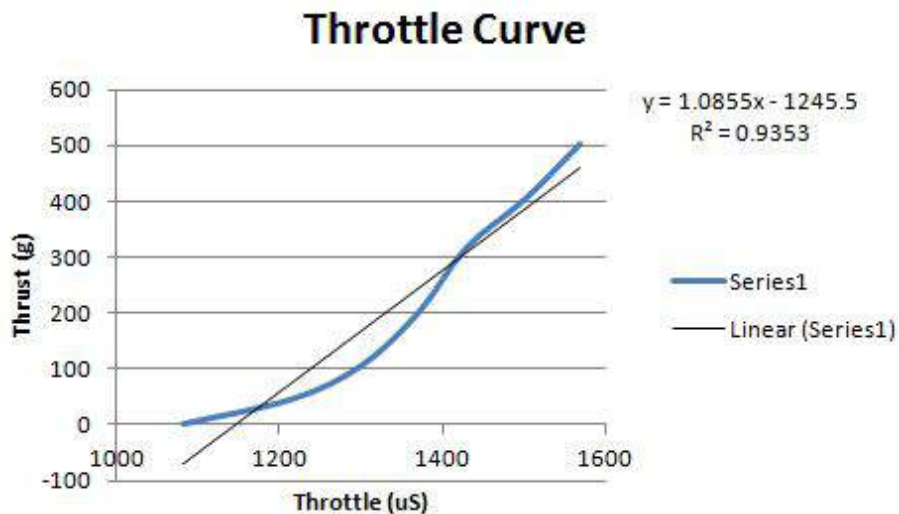


Figure 28: Thrust versus throttle of one centrifugal fan

The throttle in μS microseconds represents the pulse width of the PWM signal which controls the ESCs. Our pulse width range is 1000-2000 μS so at approximately 60 percent of throttle the thrust obtained is about 500g with is enough for our purposes. As it's possible to see in the plot after 40 percent it becomes linear.

6 Code Design

The code of the drone was done on the Arduino IDE platform in C/C++. This platform was chosen for our project because of the many available resources on the open-source Arduino as well as many libraries what was available. We will go over the general flow of the software and the main libraries we wrote: control state, communication, and motor control. The important parts of the code can be seen in Appendix C.

6.1 Flow Chart

The flow chart shown in Figure 29 explains how the software will act generally. The drone first checks if it is given the signal to arm itself and does so. Once the drone is armed, the drone will continue its previous instructions while waiting for any new commands. The instructions will update the motor controllers and while checking for any failsafes. Once a failsafe is detected, the drone will begin its descend.

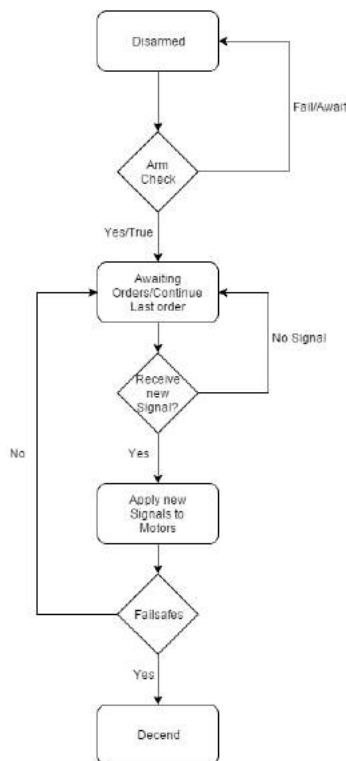


Figure 29: Flowchart for drone

6.2 Control State

The control state, as its name implies holds the states of the drone. This includes the constant values, transceiver values, arm state, alarm state, and the sensor values. We chose to create a control state library as well as other libraries in order to minimize the length of the code in the main .ino file as well as increase compilation speed. By using a control state, we are able to keep local copies of values like the calculated throttle and the previous throttle. By doing this, we have a back up of the data just in case something goes wrong.

A great advantage of using a control state is the ability to use and update variables defined in another library. If we were to approach this issue by creating one large program file, it will look unprofessional as well as prove difficult to debug modules.

The code can be seen in Appendix C Figure 42

6.3 Communication

The communication library deals with the reading of signals of the transmitter. There are generally 2 methods to read PPM signals from a receiver. One is using the function pulseIn() and the other is using interrupts. The pulseIn() method is the easiest to implement as it is already provided in the Arduino library. However, the signal reading are very noisy where the values we want and the one we get from the method is different. This is because the system is affected by the other signals and cannot filter the signals correctly.

The interrupt method is how we dealt with the communication for our project. This method is harder to implement but produces values much cleaner compared to the pulseIn() method. The interrupt method works by first detecting a signal HIGH on one of the channels and starts a timer in microseconds. When the signal is LOW, signifying the end of the pulse, the duration of the pulse is calculated using the timer started earlier. After this, a bit flag is raised to show that there was a change in signal in one of the channels. When it is time to feed the new signals into the control state, the interrupt is turned off to take local copies of the values and update them according to which flag was raised. After all of the new values are implemented, the bit flag is set back to 0 to signify no changes and interrupts is turn back on. Using this method allows us to avoid any blocking code that will cause noise in the signal we read. This portion of the code was implemented based on the tutorial from the website RCArduino [11].

How we calculated the values can be seen in the Appendix C Figures 43 and 44.

6.4 Motor Control

The motor control library handled the writing to the motors as well as arming and disarming the motors. The arming/disarming was done by passing the minimum or zero speed to the speed controller for a duration of 1 second which is translated into disarm and arming the drone. Another task the library handled was writing the values to the motors using the function `writeMicroseconds()` which as it sounds like, writes a PWM signal with an input parameter of μS . This was a function that came from the servo library in the Arduino platform. When writing the code, the main procedure is to pass in the values we want, map the value to the limits of the speed controller and then use the `writeMicroseconds` function.

Parts of the code can be seen in the Appendix C Figure: 45 and 46.

7 Custom Hardware Electrical Design

7.1 Circuit Overview

For our custom hardware, we designed a custom power distribution board with integrated safety features for a drone. The power distribution board has 3 main modules: LEDs, a voltage alarm circuit and a current sensor circuit.

7.2 LED Module

The LED module was designed to show orientation of the drone. LEDs of the colors: white, red and green were used to show the front, left and the right of the drone respectively. LEDs were used instead of only coloring the drone chassis since there are instances where drones are flown in dark areas or far away. It is important to know the drone's orientation so that the operator can compensate for wind and fly properly. The module was designed using the following equation.

$$I = \frac{V_B - V_A}{R} \quad (12)$$

Using this equation, it is possible to determine the resistor necessary for the module. After research, it has been determined that red and green LED's consumes 2V and draws 20mA while a White LED consumes 4V and draws 20mA[12]. With this information, it is possible to calculate the resistance required for the red/green and white LEDs (150Ω and 70Ω respectively). These value were calculated using 5V as the input voltage through a 7805 Linear voltage regulator so that we may adjust the LED brightness by changing the resistor or the regulator.

7.3 Voltage Alarm

The voltage alarm is a system to protect the battery from over discharging. Lithium Polymer batteries can be damaged when overdischarged. In order to prevent over discharging the battery, it was designed a voltage shunt that will trigger when the battery bias is below a set value. It has been found very similar circuit from the Texas Instruments' data sheet for the TL431 Precision Programmable Reference[13]. The shown circuit in Appendix B uses two TL431 chips to turn an LED on when the input voltage is within a threshold. Threshold is determined using four resistors and the following equations from the datasheet:

$$LowLim = (1 + \frac{R_{1B}}{R_{2B}})V_{REF} \quad (13)$$

$$U_{ppLim} = (1 + \frac{R_{1A}}{R_{2A}})V_{REF} \quad (14)$$

V_{REF} was given in the data sheet as 2.5V and we wanted the upper limit to be 10.6V and the lower limit as 10V. These values comes from taking the maximum voltage of a lipo cell of 4.2V and using 80% of it as a safe value. With these values and the provided equations, we chose the resistor values of 100Ω and 300Ω for R_{2B} , R_{1B} and 100Ω and 330Ω for R_{2A} , R_{1A}

7.4 Current Sensor

The current sensor module ensures that the motor will not draw more current than it can handle due to the large load of the impeller. For this part of the circuit, we decided to used an ACS709 Hall effect based current sensor and a voltage divider. The used of the ACS709 was inspired by Hacktronics’s arduino current sensor project [14].Even though the circuit was inspired from their project, our design differs significantly since the current it’s being measuring is massive compared to that of Hacktornic’s circuit. it’s also needed to use a voltage divider circuit to bring down the output voltage to a safe range for our Arduino DUE since the Arduino DUE’s I/O pins can only tolerate 3.3V[15]. For since it’s known that output voltage of ACS709 is typically $V_{CC} * 0.5$ [16], it can be assumed that the output voltage is about 5V since the input is about 10V. Therefore, a voltage divider can be used which formula is shown below to choose the correct resistances.

$$V_2 = \frac{R_2}{R_1 + R_2} * V_1 \quad (15)$$

Using the voltage divider formula, it can be found the following ratio for the resistors.

$$R_1 = 0.4R_2 \quad (16)$$

With this in mind, commonly available resistor values would be 3.9kΩ resistor for R_1 and 5.6kΩ resistor for R_2 . This gives an output voltage of 2.95V with an input of 5V which is a safe range for our microconroller.

The circuit schematics as well as the PCB design may be viewed in Appendix B

8 Control design

8.1 Dynamics

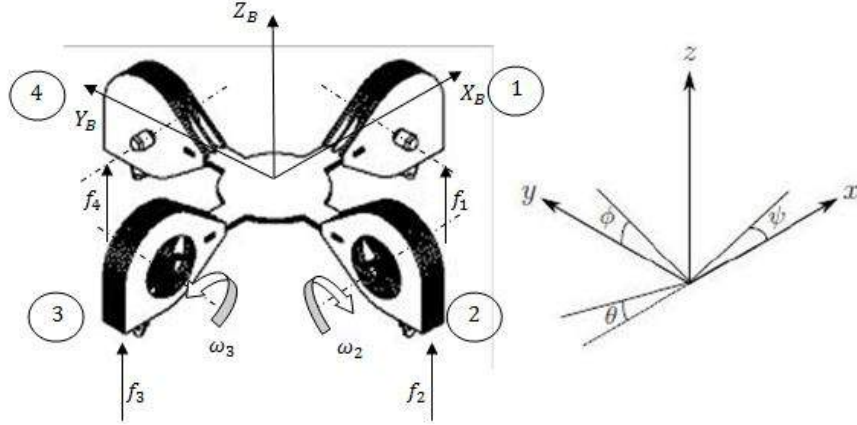


Figure 30: The inertial and body frames of a quadcopter

The first step to derive the quad-copter's dynamics is to introduce the two frames in which will operate. The inertial frame is defined by the ground, with gravity pointing in the negative z direction. The body frame is defined by the orientation of the quadcopter, with the frame pointing in the positive z direction and the arms pointing in the x and y directions as the rotors do.

The absolute linear position of the quadcopter is defined in the inertial frame x, y, z axes with ζ . The attitude, i.e. the angular position, is defined in the inertial frame with three Euler angles χ . Pitch angle θ determines the rotation of the quadcopter around the y -axis. Roll angle ϕ determines the rotation around the x -axis and yaw angle ψ around the z -axis[17].

$$\chi = \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} \quad \zeta = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (17)$$

The rotation matrix from the body frame to the inertial frame is:

$$R = \begin{pmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\theta S_\phi S_\psi + C_\phi C_\psi & S_\psi S_\theta C_\phi - C_\psi C_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{pmatrix} \quad (18)$$

In which $S_x = \sin(x)$ and $C_x = \cos(x)$.

This matrix is obtained from the fundamental rotation matrixes across ϕ, θ and ψ

$$R = R_1(\psi)R_2(\theta)R_3(\phi) \quad (19)$$

Where:

$$R_1(\psi) = \begin{pmatrix} C_\psi & -S_\psi & 0 \\ S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} R_2(\theta) = \begin{pmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{pmatrix} R_3(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi & C_\phi \end{pmatrix} \quad (20)$$

The origin of the body frame is in the center of mass of the quadcopter. In the body frame, the linear velocities are determined by V_B and the angular velocities by ω

$$V_B = \begin{pmatrix} v_{x,B} \\ v_{y,B} \\ v_{z,B} \end{pmatrix} \omega = \begin{pmatrix} \omega_{x,B} \\ \omega_{y,B} \\ \omega_{z,B} \end{pmatrix} \quad (21)$$

Meanwhile in the inertial frame, $\dot{\zeta}$ represents the linear velocity and $\dot{\chi}$ the angular velocities.

$$\dot{\zeta} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \dot{\chi} = \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (22)$$

The transformation matrix for angular velocities from the inertial frame to the body frame is W_χ

$$\omega = W_\chi \dot{\chi}, \quad \begin{pmatrix} \omega_{x,B} \\ \omega_{y,B} \\ \omega_{z,B} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (23)$$

The quadcopter is assumed to have symmetric structure with the four arms aligned with the body x- and y-axes. Thus, the inertia matrix is diagonal matrix I in which $I_{xx} = I_{yy}$

$$I = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \quad (24)$$

The angular velocity of rotor i , denoted with Ω_i , creates force f_i in the direction of the z

axis. The angular velocity and acceleration of the rotor also create torque τ_{Mi} around the rotor axis[17].

$$f_i = k\Omega_i^2, \tau_{Mi} = b\Omega_i^2 + I_M\dot{\Omega}_i \quad (25)$$

in which the lift constant is k, the drag constant is b and the inertia moment of the rotor is I_M . Usually the effect of $\dot{\Omega}_i$ is considered small and thus it is omitted.

The combined forces of rotors create thrust T in the direction of the body z-axis. Torque τ_B consists of the torques τ_ϕ , τ_θ and τ_ψ in the direction of the corresponding body frame angles.

$$T = k \sum_{i=1}^4 \Omega_i^2, T^B = \begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} \quad (26)$$

$$\tau_B = \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} lk(\Omega_4^2 - \Omega_2^2) \\ lk(\Omega_3^2 - \Omega_1^2) \\ b(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2) \end{pmatrix} \quad (27)$$

in which l is the distance between the rotor and the center of mass of the quadcopter. [17].

This is the case of an ordinary propeller. Due to the design of our fans, yaw movement is achieved by directing the thrust force in order to create a torque in the z axis. Torques matrix will be the next:

$$\tau_B = \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} lk(\Omega_4^2 - \Omega_2^2) \\ lk(\Omega_3^2 - \Omega_1^2) \\ lk \sin \gamma (\Omega_1^2 + \Omega_3^2) \end{pmatrix} \quad (28)$$

where γ is the angle between f_i and the vertical axis. For our system we have neglected de the torque produced by $b\Omega_i^2$ because it is much smaller than the torque produced by $lk\Omega_i^2$

The equations of motion are:

$$m\ddot{x} = \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} + RT_B \quad (29)$$

$$\tau = I\dot{w} + w \times (Iw) + \Gamma \quad (30)$$

Where Γ are the gyroscope forces.

From last equation it's possible to obtain \dot{w} :

$$\begin{pmatrix} \dot{\omega}_{x,B} \\ \dot{\omega}_{y,B} \\ \dot{\omega}_{z,B} \end{pmatrix} = \begin{pmatrix} (I_{yy} - I_{zz})\omega_{y,B}\omega_{z,B}/I_{xx} \\ (I_{zz} - I_{xx})\omega_{x,B}\omega_{z,B}/I_{yy} \\ (I_{xx} - I_{yy})\omega_{x,B}\omega_{y,B}/I_{zz} \end{pmatrix} + J_R \begin{pmatrix} \omega_{y,B}/I_{xx} \\ -\omega_{x,B}/I_{yy} \\ 0 \end{pmatrix} \sum_{i=1}^4 \Omega_i + J_R \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \sum_{i=1}^4 \alpha_i + \begin{pmatrix} \tau_\phi/I_{xx} \\ \tau_\theta/I_{yy} \\ \tau_\psi/I_{zz} \end{pmatrix} \quad (31)$$

J_R motor inertia.

Due to the difficulty of our system it has been decided to do a study of the dynamics of a regular drone which uses propellers.

8.2 State Equations

The state vector is:

$$x(t) = (x, y, z, u, v, w, \phi, \theta, \psi, \omega_{x,B}, \omega_{y,B}, \omega_{z,B})^T \quad (32)$$

Our system have eight control's inputs:

$$u(t) = (\Omega_1, \Omega_2, \Omega_3, \Omega_4, \alpha_1, \alpha_2, \alpha_3, \alpha_4)^T \quad (33)$$

And the output vector:

$$y(t) = (x, y, z, \phi, \theta, \psi)^T \quad (34)$$

Finally, I have considered that angular accelerations are inputs that have not relationship with the speeds and for this reason can be put out of the model.

We have not take into account any drag force due to our indoor purposes The system

equations are[18]:

$$\begin{aligned}
\dot{x} &= u, \\
\dot{y} &= v, \\
\dot{z} &= w, \\
\dot{u} &= \frac{k}{m}(\cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi)) \sum_{i=1}^4 \Omega_i^2, \\
\dot{v} &= \frac{k}{m}(\sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi)) \sum_{i=1}^4 \Omega_i^2, \\
\dot{w} &= -g + \frac{k}{m}\cos(\theta)\cos(\phi) \sum_{i=1}^4 \Omega_i^2, \\
\dot{\phi} &= \omega_{x,B} + \omega_{y,B}\sin(\phi)\tan(\theta) + \omega_{z,B}\cos(\phi)\tan(\theta), \\
\dot{\theta} &= \omega_{y,B}\cos(\phi) - \omega_{z,B}\sin(\phi), \\
\dot{\psi} &= \sec(\theta)(\omega_{z,B}\cos(\phi) - \omega_{y,B}\sin(\phi)), \\
\dot{\omega}_{x,B} &= \frac{I_{yy} - I_{zz}}{I_{xx}}\omega_{y,B}\omega_{z,B} + \frac{J_R}{I_{xx}}\omega_{y,B} \sum_{i=1}^4 \Omega_i + \frac{lk(\Omega_4^2 - \Omega_2^2)}{I_{xx}} \\
\dot{\omega}_{y,B} &= \frac{I_{zz} - I_{xx}}{I_{yy}}\omega_{x,B}\omega_{z,B} - \frac{J_R}{I_{yy}}\omega_{x,B} \sum_{i=1}^4 \Omega_i + \frac{lk(\Omega_3^2 - \Omega_1^2)}{I_{yy}} \\
\dot{\omega}_{z,B} &= \frac{I_{xx} - I_{yy}}{I_{zz}}\omega_{x,B}\omega_{y,B} - \frac{J_R}{I_{zz}} \sum_{i=1}^4 \alpha_i + \frac{b(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2)}{I_{zz}}
\end{aligned} \tag{35}$$

8.3 Lineralization

The general equation of a nonlinear system as we have are:

$$\begin{aligned}
\dot{x}(t) &= f[x(t), u(t), t] \\
y(t) &= g[x(t), u(t), t]
\end{aligned} \tag{36}$$

But to obtain the dynamic transfer function of our drone is necessary to have a linear model as:

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t)\end{aligned}\tag{37}$$

For these reasons we have to linearize our system around an equilibrium point where considering small displacements:

$$\begin{aligned}x(t) &= x_0(t) + \delta x(t) \\ u(t) &= u_0(t) + \delta u(t)\end{aligned}\tag{38}$$

so:

$$\begin{aligned}f[x(t), u(t), t] &= f[x_0(t), u_0(t), t] + f_x \delta x(t) + f_u \delta u(t) \\ g[x(t), u(t), t] &= g[x_0(t), u_0(t), t] + g_x \delta x(t) + g_u \delta u(t)\end{aligned}\tag{39}$$

In the previous equation x , u , g , f are vectors and the term $f_x(t)$, $f_u(t)$, $g_x(t)$ and $g_u(t)$ are jacobian matrix.

$$f_x(t) = \left. \frac{\partial f}{\partial x} \right|_{x_0, u_0} \quad f_u(t) = \left. \frac{\partial f}{\partial u} \right|_{x_0, u_0} \quad g_x(t) = \left. \frac{\partial g}{\partial x} \right|_{x_0, u_0} \quad g_u(t) = \left. \frac{\partial g}{\partial u} \right|_{x_0, u_0}$$

so the linear equations are:

$$\begin{aligned}\delta\dot{x} &= A\delta x + B\delta u \\ \delta\dot{y} &= C\delta x + D\delta u\end{aligned}\tag{40}$$

Where A(t) has a dimension of 12x12 (12 states), B(t) has a dimension of 12x8 (8 inputs) and C(t) has a dimension of 6x12 (6 outputs)[18].

8.4 Equilibrium point

To obtain the equilibrium points of the system, we have to look for the values that make zero the state vector. So we make the state vector equal to zero and then we solve the equation. The equilibrium point is the next[18]:

$$\begin{aligned}x(t) &= (x, y, z, u, v, w, \phi, \theta, \psi, \omega_{x,B}, \omega_{y,B}, \omega_{z,B})^T \\ x_0(t) &= (x, y, z, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T\end{aligned}\tag{41}$$

Therefore, the inputs that gives these solutions are[18]:

$$\begin{aligned}\Omega &= \frac{1}{2}\sqrt{\frac{gm}{k}} \\ \alpha &= 0\end{aligned}\tag{42}$$

8.5 Obtaining the Laplace transform

Once the system has been linearized to obtain the Laplace transform:

$$\begin{aligned} sX(s) - x(0) &= AX(s) + BU(s), \\ Y(s) &= CX(s) \end{aligned} \tag{43}$$

Therefore:

$$\frac{Y(s)}{U(s)} = P(s) = C(sI - A)^{-1}B \tag{44}$$

This will result in a MIMO system (Multiple Inputs Multiple Outputs) which after being uncoupled will result in a SISO system.

8.6 SISO system

The final SISO system is[18]:

$$\begin{pmatrix} x(s) \\ y(s) \\ z(s) \\ \phi(s) \\ \theta(s) \\ \psi(s) \end{pmatrix} = \frac{1}{s^2} \begin{pmatrix} 0 & 0 & \frac{gkl}{I_{yy}s^3} & 0 \\ 0 & -\frac{gkl}{I_{xx}s^3} & 0 & 0 \\ \frac{k}{ms} & 0 & 0 & 0 \\ 0 & \frac{kl}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{kl}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{b}{I_{zz}} \end{pmatrix} \begin{pmatrix} U_1(s) \\ U_2(s) \\ U_3(s) \\ U_4(s) \end{pmatrix} \tag{45}$$

where $U_1(s)$, $U_2(s)$, $U_3(s)$, $U_4(s)$ are:

Vertical Thrust

$$U_1(s) = \Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2 \tag{46}$$

Roll:

$$U_2(s) = \Omega_4^2 - \Omega_2^2 \quad (47)$$

Pitch:

$$U_3(s) = \Omega_3^2 - \Omega_1^2 \quad (48)$$

Yaw:

$$U_4(s) = \Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2 \quad (49)$$

Thanks to the axis of movement chosen, the system can be decoupled. Roll and Pitch depends just of two motors while when Yaw is active Pitch is also affected due to the position of the servomotors.

8.7 Control

The control schematic for the pitch can be modeled as the next one:

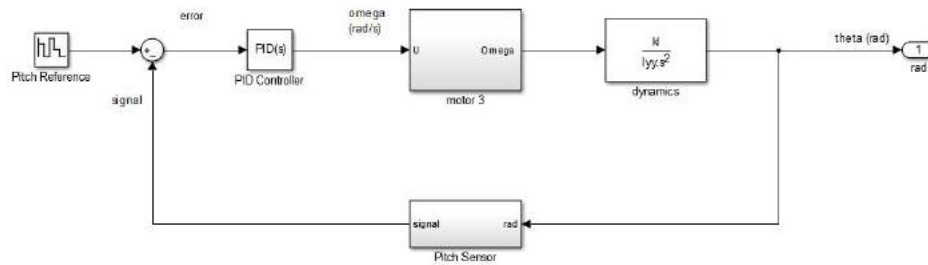


Figure 31: Pitch Control

After more research I have realized that most of the current drone's control are not design taking into account all the physical parameters like inertia, etc. Knowing the exact values of all the physical parameters will make possible to design an accurate control choosing the proper values of the PID's parameters. Due to the fact that this is not possible, we will design a control without paying much attention to the parameters which will be tunned later by doing stabilization tests.

Control compute

To compute the PID values we have use the Arduino library *PIDV1.h*:

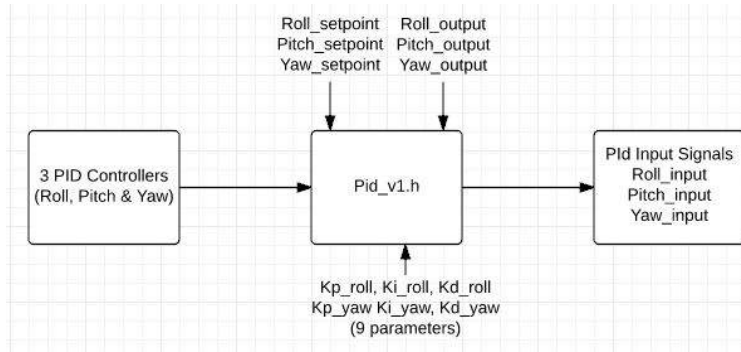


Figure 32: Control block

This library needs the three setpoint values obtained from the stick response of the pilot, the three current angles of the drone (pitch, roll and yaw) from the sensor readings and the nine parameters of the three PIDs (K_p , K_i and K_d of every axis of movement). In real there are just 6 different parameters because roll and pitch have the same numbers. The implementation of this code can be found in the appendix G.

The input signals generated by the control block are the width of the pulses that must be added or subtracted to the throttle pulse of every motor. The range of our pulse modulation is $1500\text{-}2000\mu\text{s}$ (minimum and maximum width), for this reason establish a maximum control output (input signal of the block diagram, U) of $\pm 200\mu\text{s}$ should be enough to stabilize the drone.

Control and motors update

In order to change the pitch and roll angles, the main idea is to change each motor's speed so the quad-copter starts bending in the desired direction. Just to obtain vertical lift we just simply redirect the throttle input (coming from a remote control) to the motors, which will just concretely change the altitude/vertical speed of the drone. The code will be:

```
m1 = throttle;
m2 = throttle;
m3 = throttle;
m4 = throttle;
```

Now, let's think a bit about how to make the quad copter move forward/ backward. This motion corresponds to the pitch angle. In order to change the pitch, we only need to change the speed of the two motors on the X axis (motors 1 and 3). Indeed, affecting the speed of the motors 2 and 4 (on the Y axis) would just start a rotation around the X axis therefore

changing the roll angle.

In order to incline the aircraft there must be a difference in the motors' speed. The best way to get different motors' speed is to add the offset value on one motor output, and to subtract it on the other, so the "overall throttle" of the axis remains constant and therefore changing the pitch won't affect the quad-copter's "vertical speed" [19].

$$Throttle_{13} = \frac{(throttle + pitch_{input}) + (throttle - pitch_{input})}{2} = throttle \quad (50)$$

The pseudo code is shown in Figure 33

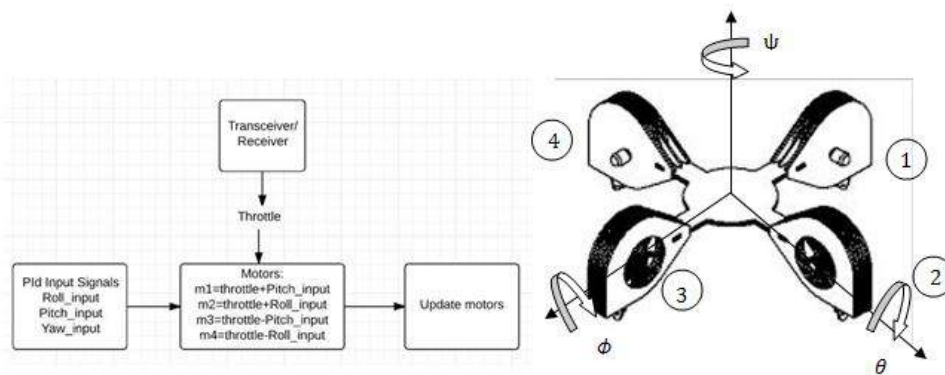


Figure 33: Updating control on motors

Fans 1 and 3 control pitch while 2 and 4 control roll. Yaw's control is not needed for drone stabilization and for this reason has not been included in the control. The control updating can be found on appendix G figure 30.

8.8 PID controller

The PID controller of each axis is the key to have a stable flight experience. Three separate constant parameters for each angular control (roll and pitch shares parameters), K_P depends on the present error, K_I depends on the accumulation of past errors and K_D is a prediction of future errors.

K_P is the main value to worry about, the stability of the system depends on this parameter. The higher the P is, the fastest the system will respond. If P is too high overshoots will be big until the system becomes unstable.

The effect of K_I is keeping the last stick position. It's integrating the error over time, the longer the error persist, the greater the force is applied. This parameter set the steady state error to zero.

K_D changes the forced applied to correct position error, when it sees a decrease or increase in the position error. So you will see a softer movement in the quadcopter when D is used.

PID tuning process

For the purpose of adjust the PID parameters, we have to fix one axis of movement for testing, e.g roll, so only one axis of rotation is active at time. During the this test roll, pitch and yaw are set to 0 (setpoint=0) from the stick position in the receiver. The only signal different to zero is throttle. The idea is trying to tilt the drone by hand, motors must counteract this effect producing enough torque. A proper set of parameters would make very hard to move the drone by hand over time.



Figure 34: Setup, stabilization test

Let's describe the steps to adjust PID parameters[20]:

1. Set Yaw PID parameters to zero and roll/pitch I-D gain to zero. Let's increase P-gain.
 - 1.1. If P-gain is too low is easy to turn by hand (slow and imprecise to stick responses when a different to zero setpoint is set.).
 - 1.2. Increase P-gain until oscillations appears. This means that the system has become unstable. The proper P-gain must be below this ultimate P.
 - 1.3. A good P-gain must assure that it is hard to turn by hand and sharp stick response.
2. Now let's tune I-gain:
 - 2.1. A low I-gain makes easy to tilt the drone by hand over long time.

- 2.2. When I-gain is too high the drone oscillates with a lower frequency than with a too high-P gain.
 - 2.3. A good I-gain makes very hard to move the drone by hand over long time. It stays in position when stick is released (in case you set a setpoint different to zero).
3. Finally, let's tune D-gain:
 - 3.1. For having a soft flight experience, increase D-gain slowly until shaking finishes.

9 Cost Analysis

9.1 Part costs

Table 2: Parts Costs

Part	Manufacturer	Model	Quantity	Cost/unit (\$)
Brushless Motor	NTM	NTM Prop Drive 450	4	22.99
Battery	Turnigy	Bolt 3S	1	29.08
Microcontroller	Arudino	Due	1	45.00
Gyro + Accelerometer	Adafruit	Breakout Board	1	6.88
Transceiver	Spektrum	Dx6i/AR600	1	120
Actuator	Sparkfun	Generic Sub-micro	2	10.00
ESC	Hobby King	Hobby King 50A ESC	4	20.00
Chassis Frame	-	-	1	92.69
PCB Parts	Panasonic	-	1	36.54
Custom Shield	-	-	1	2.16
Misc Parts	-	-	-	5.00
Total				\$ 517.15

9.2 Labor costs

Table 3: Labor Costs

Personnel	Salary per Hour (\$)	Hours Worked	Total = Salary per Hour * 2.5 * Hours Worked (\$)
Bree Peng	37.5	180	16,875
Ignacio Aguirre	37.5	180	16,875
Leo Yamamae	37.5	180	16,875
Total			\$50,625

9.3 Grand total cost

Table 4: Grand Total Cost

Item	Cost (\$)
1 Drone Part Cost	517.15
Labor Cost	\$50,625.00
Grand Total	\$51,142.15

10 Ethical / Legal Considerations

10.1 IEEE Code of Ethics Considerations

Since this drone will be airborne, this will involve the first code of the IEEE Code of Ethics where we must consider the decisions that we make which will involve the safety, health, and welfare of the public [21]. This is because even if the drone is safer than those that uses propellers, it may still fall from the sky and cause injury or property damage, cause injury or property damage from flight motions, catch fire if overheated, or catch fire if the battery is punctured. This is a concern that regarding the clause 9 in the IEEE code of ethics to avoid injuring others, their property, reputation, or employment by false or malicious action. Because since the drone will be capable of carrying and delivering a payload, we must deter its usage for malicious intent through its specifications.

10.2 Legal Obligations

A legal consideration that we must consider is the FAA regulations on Model Aircraft Operations. The FAA recently released the guidelines on flying for hobby or recreation, which restricts recreational model flight to below 400 feet, away from other people or stadiums, et cetera. The regulation that most applies to this drone states that one may not fly within 5 miles of an airport without clearance from the airport and control tower before flying [22]. This applies to our drone because the University of Illinois at Urbana-Champaign is within the 5-mile radius of the University of Illinois Urbana-Champaign Willard Airport. This means that before any testing is being done on the drone, we must contact the control tower at the Willard Airport for clearance. In addition, the University of Illinois has also placed regulations and restrictions on drone use on campus. Another legal issue that concerns our project is the registration of our drone. FAA has made it mandatory for anyone who has a unmanned aircraft that is heavier than 250g but lighter than 25kg to register their drone on their website to fly outdoors[22]. Our drone will weigh close to 1kg which falls into this category.

However, to be able to test the system without having to go out to an airfield every time, it will be tied to ground. By doing this, the project will not be technically a drone and should be safe enough to be tested from yaw, pitch, roll, stability even indoors.

10.3 Safety Statement

In this project, there are many safety considerations that must be taken into mind. This is because the project is a drone which upon completion will be airborne. During the build phase, there are many step that pertains safety concerns.

The fact that the project is a drone is one safety concern that we have to address according to the first and second IEEE code of ethics. The first statement is to to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment.[21] This is a factor we need to consider because even though our drone will be safer than a drone that uses propellers, the drone could still damage unsuspecting people or property when it falls from a height.

Also the drone will be powered by a Lithium-Polymer battery which by itself is a safety concern. This is because a Li-Po has many factors that could cause it to explode releasing harmful chemicals. Some events that could cause the battery to explode include but is not limited to charging using an incorrect battery charger, shorting wires, puncturing cells, and overcharging[23]. This is a great concern as fires indoors is not a thing to be taken lightly especially when there are harsh chemicals released.

The second statement in the IEEE code of ethics needs to be considered because as we are flying a motorized object within 5-miles of an airport, we must let the airport control tower know that we will be flying a drone[24]. Even though from past experience, the control tower does not mind the drone being flown at our altitude, there are major concerns of drones interrupting plane and helicopter flight.

Another concern with our project is the safety of the new propulsion method. If the impeller shatters on impact and creates more harm than a normal propeller, the project has failed. From testing conducted, we saw that the Impeller shreds into a confetti or safety glass shards after exploding in a safe direction. From this, we were able to see that the drone propulsion method is safer and a typical propeller as nothing was cut when the impeller shattered.

11 Conclusion

In this project we were able to design and construct a drone that is capable of flight without the use of propellers. Our design used a centrifugal fan composed of Polystyrene which produced a peak thrust of 550g and was capable of withstanding falls without breaking. Also, when the fan failed, the impeller shattered like safety glass and posed no dangers unlike shattered plastic propellers. Using an Arduino microcontroller as the controller, we were able to interpret sensor data from the IMU and attempted to determine stability. However, due to power supply limitations and parameter change times, we were unable to calibrate the system to achieve stability during flight. Another issue we encountered was the PCB traces not being large enough withstand the current required by the motors.

If we were to continue this project, we would optimize the fan design to output greater thrust with a smaller overall design as well as explore other materials such as composites. We will also redesign our power distribution board to be formed in layers to withstand the high current of the system. Either the motors selected must draw less current or we must get a power supply with a greater capacity for sustained flight. With a longer flight time and the right apparatus, we will be able to improve upon control and achieve stability in flight.

Appendix A Requirements and Verification

Table 5: System Requirements and Verifications

Requirement	Verification	Point Allocation
<p>1. Power Supply</p> <p>(a) Capable of handling 286A burst output*</p> <p>(b) Supply $12.6V \pm 0.25V$ when fully charged</p>	<p>1. Power Supply</p> <p>(a) Measure voltage difference across power source using a Digital multimeter placed in parallel with the power supply. voltage should read $12.6V \pm 0.25V$</p> <p>(b) Measure current difference across power source and maximum load using a Digital multimeter placed in series with the power source and load that would draw 143A. The current should read $143A \pm 0.25A$ while maintaining the first tests temperature goals.</p>	<p>1. Power Supply</p> <p>(a) ____/1</p> <p>(b) ____/1</p> <p>Total Points:____/2</p>
<p>2. Transceiver</p> <p>(a) Capable of transmitting clean signal to receiver within 10 feet. Values $\pm 100\mu S$</p> <p>(b) Capable of actuating motors</p>	<p>2. Transceiver</p> <p>(a) Turn on transmitter and check the RX values on the serial monitor.</p> <p>(b) Turn on transmitter and spin the motor using throttle's stick</p>	<p>1. Transceiver</p> <p>(a) ____/2</p> <p>(b) ____/2</p> <p>Total Points:____/4</p>
Continued on next page		

Table 5 – continued from previous page

Requirement	Verification	Point Allocation
<p>3. Microcontroller</p> <p>(a) Capable of receiving 6 signals from the transmitter using a receiver</p> <p>(b) Capable of outputting PWM signals to the ESC</p>	<p>3. Microcontroller</p> <p>(a) Display on Serial monitor that 6 channels are being read</p> <p>(b) Hook up PWM outputs to oscilloscope and test if output signal looks right, and then connect this signal to an ESC/-motor to check that it correctly modulates the motors speed.</p>	<p>1. Microcontroller</p> <p>(a) ____/2</p> <p>(b) ____/3</p> <p>Total Points:____/5</p>
<p>4. IMU module (accelerometer + gyro)</p> <p>(a) Capable of providing accurate angle measures in degrees with an accuracy of $\pm 10^\circ$</p> <p>(b) Capable of responding fast to changes of drone’s position. Reading values at 100kHz \pm 5kHz</p>	<p>4. IMU module (accelerometer + gyro)</p> <p>(a) The test for angles consists of tilting one of the sensor’s axis (e.g. roll) while maintaining the other axis fixed (e.g. pitch). Tilt 45 ° using a triangle to check the angle. Check the measurement obtained by the sensor on Arduino’s Serial Monitor.</p> <p>(b) Defining in the code a I2C clock rate of 100kHz. Additionally, doing a simulation using the program called ”Procesing” where is possible to simulate how well responds the drone to changes in angle.</p>	<p>1. IMU module</p> <p>(a) ____/4</p> <p>(b) ____/3</p> <p>Total Points:____/7</p>
Continued on next page		

Table 5 – continued from previous page

Requirement	Verification	Point Allocation
<p>5. Actuator</p> <p>(a) It must be able to be controlled by the transmitter and adjust an angle of $45^\circ \pm 5^\circ$</p>	<p>5. Actuator</p> <p>(a) Give the servo the signal for its 45° position, measure with precision angle measurement tool, then feed the servo the signal for its -45° position. Measure position again with the same tool</p>	<p>1. Actuator</p> <p>(a) ____/2</p> <p>Total Points:____/2</p>
<p>6. Centrifugal Fans</p> <p>(a) Must generate a thrust of 455g (1.4 thrust/weight) $\pm 10g$ in conjunction with its motor running at 100% power.</p> <p>(b) Must weigh (housing + impeller)less than 100g $\pm 5g$</p> <p>(c) Each of the centrifugal fans must weigh within ± 10 grams of each other</p>	<p>6. Centrifugal Fan</p> <p>(a) Attach motors to centrifugal compressors and determine thrust by measuring force (in Newtons) of each individual compressor-motor pair on a stationary force sensor pad with the compressor thrusting itself against the pad.</p> <p>(b) Weigh module on a scale to determine mass meets requirements</p> <p>(c) Compare mass measurements between each module</p>	<p>1. Centrifugal Fan</p> <p>(a) ____/7</p> <p>(b) ____/2</p> <p>(c) ____/1</p> <p>Total Points:____/4</p>
Continued on next page		

Table 5 – continued from previous page

Requirement	Verification	Point Allocation
<p>7. Flight Controller</p> <p>(a) Maintaining 0 ± 5 set-point (roll, pitch leveled) and changing sensor values, PID outputs must change</p> <p>(b) Control must update values at $100\text{Hz} \pm 20\text{Hz}$ (sensor speed)</p> <p>(c) A good tuning of PID parameters must maintain the drone leveled parallel to the floor at $0^\circ \pm 10^\circ$</p>	<p>7. Electronic Speed Controller</p> <p>(a) Check in the Serial Monitor that PID outputs changes depending on the sensor readings</p> <p>(b) Check in the Serial Monitor that PID outputs are updating at the same speed that sensors do</p> <p>(c) Make the drone fly leveled after pitching or rolling</p>	<p>1. Flight Controller</p> <p>(a) ____/2</p> <p>(b) ____/2</p> <p>(c) ____/4</p> <p>Total Points:____/8</p>
<p>8. Motors</p> <p>(a) Capable of providing $455\text{g} \pm 10\text{g}$ of overall lift in junction with centrifugal fan</p>	<p>8. Motors</p> <p>(a) Attach motors to centrifugal compressors and determine thrust by measuring force (in grams) of each individual compressor-motor pair on a stationary force sensor pad with the compressor thrusting itself against the pad.</p>	<p>1. Motors</p> <p>(a) ____/5</p> <p>Total Points:____/5</p>
<p>9. Chassis Frame</p> <p>(a) Capable of withstanding a fall of $1\text{m} \pm 0.1\text{m}$</p> <p>(b) Must weigh in overall (including electronics and fans) less than $1.4\text{kg} \pm 0.1\text{kg}$</p>	<p>9. Chassis Frame</p> <p>(a) Drop drone from 1.1m and repeat at 0.9m</p> <p>(b) Weigh drone on a scale. It must weigh less than $1.4\text{kg} \pm 0.1\text{kg}$</p>	<p>1. Chassis Frame</p> <p>(a) ____/1</p> <p>(b) ____/3</p> <p>Total Points:____/8</p>
Continued on next page		

Table 5 – continued from previous page

Requirement	Verification	Point Allocation
<p>9. PCB</p> <p>(a) Output signals biased from $0 \pm 0.25V$ to $3 \pm 0.3V^*$</p> <p>(b) Voltage alarm rings and lights up LED when the input voltage is less than $10.6 \pm 0.25V$</p>	<p>9. PCB</p> <p>(a) Hook up power supply and use multimeter to measure voltage at output pins</p> <p>(b) Supply the PCB with 10.8V with power source and see if LED lights up and alarm rings</p>	<p>1. PCB</p> <p>(a) ____/1</p> <p>(b) ____/2</p> <p>Total Points:____/3</p>
		<p>Total Points</p> <p>____/50</p>

Appendix B Circuit Schematic

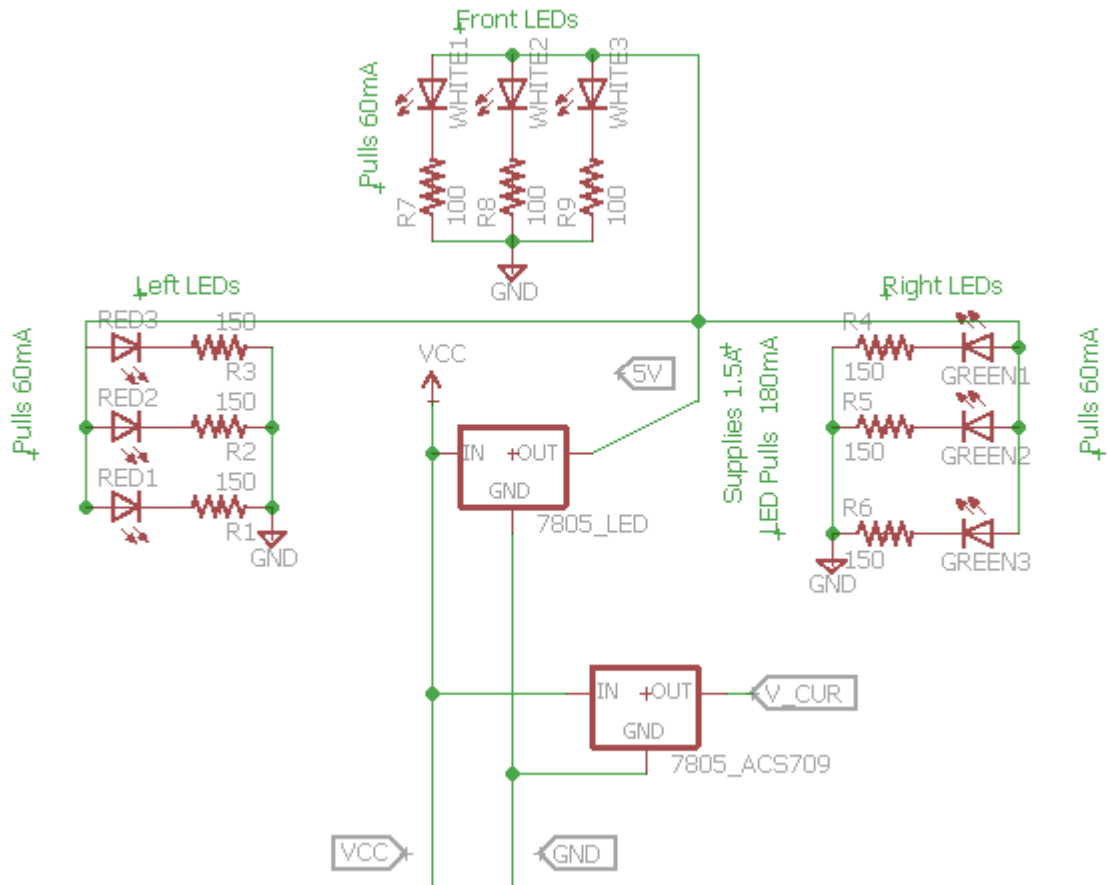


Figure 35: LED module to display orientation of drone

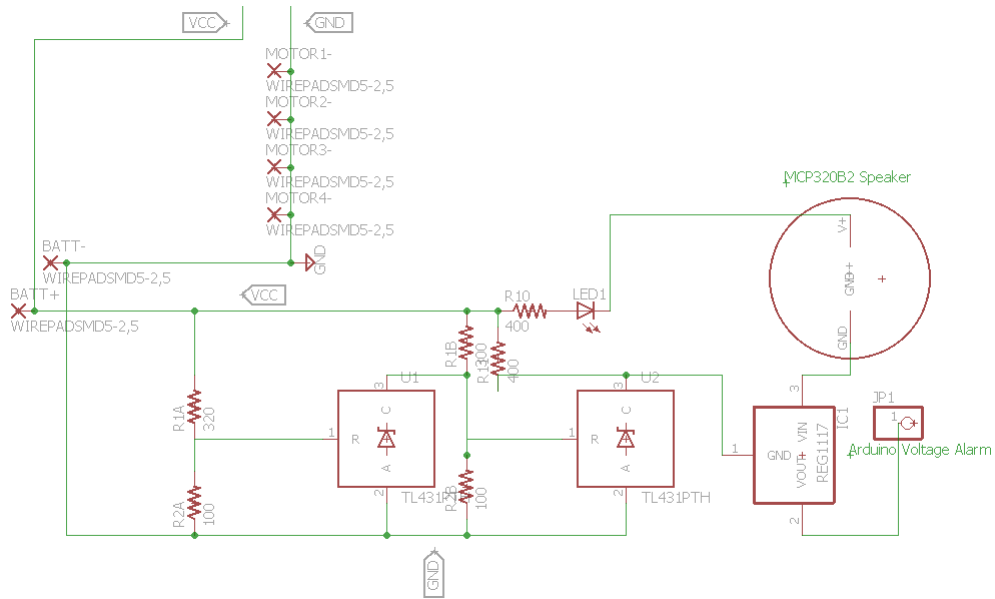


Figure 36: Voltage Alarm module and Mounting pads for motors

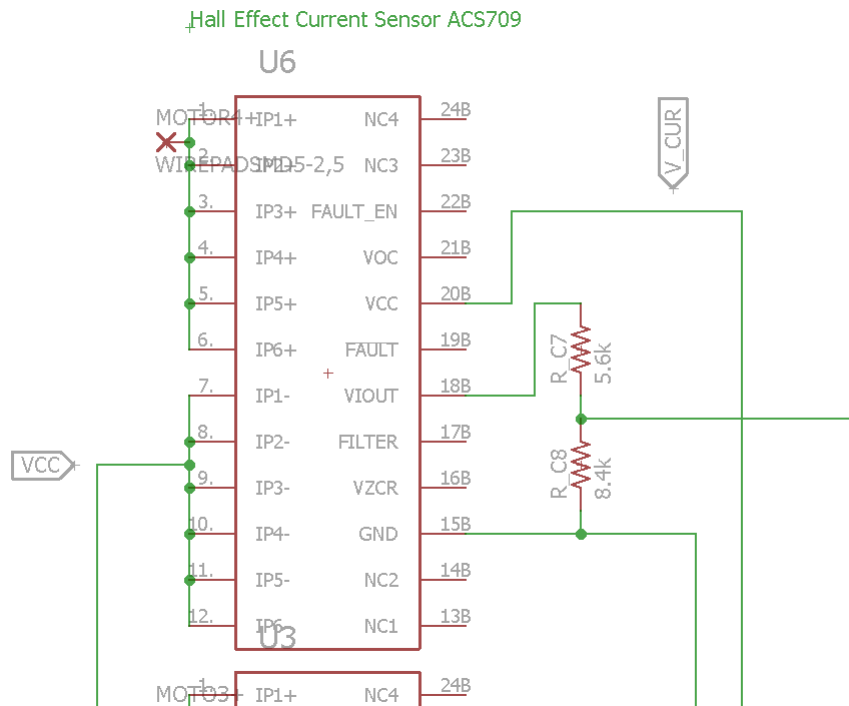


Figure 37: One of the four ACS709 Chips used for sensing current

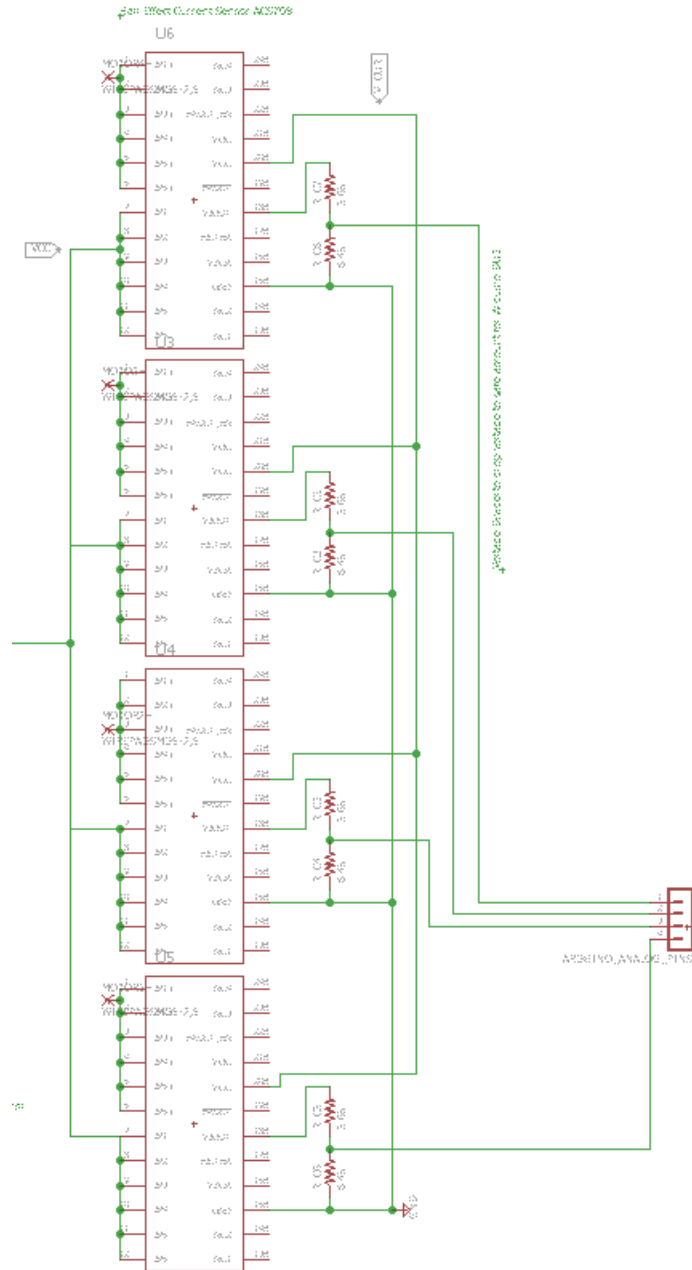


Figure 38: All four of the ACS709 Chips used for sensing current

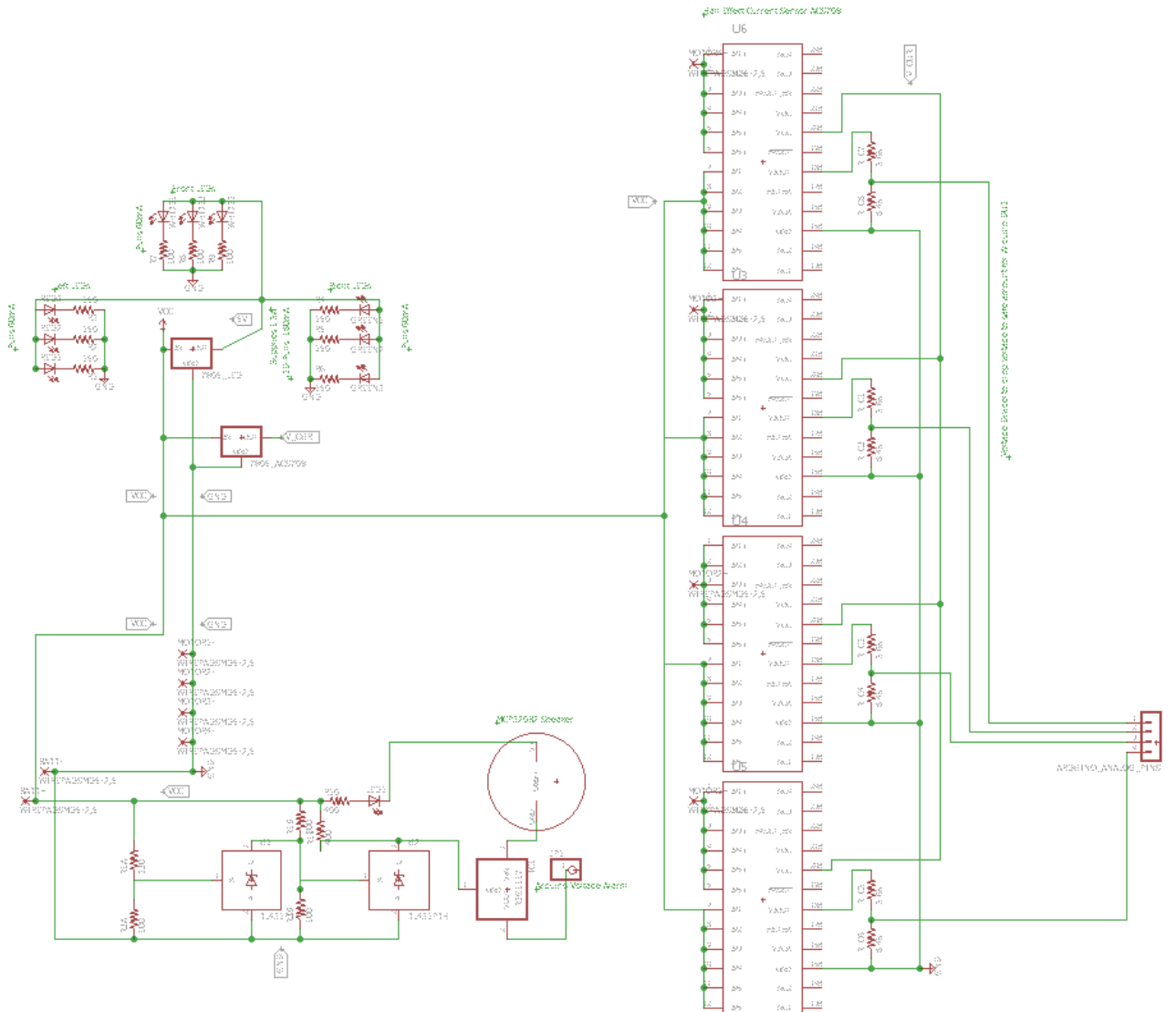


Figure 39: Complete Circuit

PCB Design

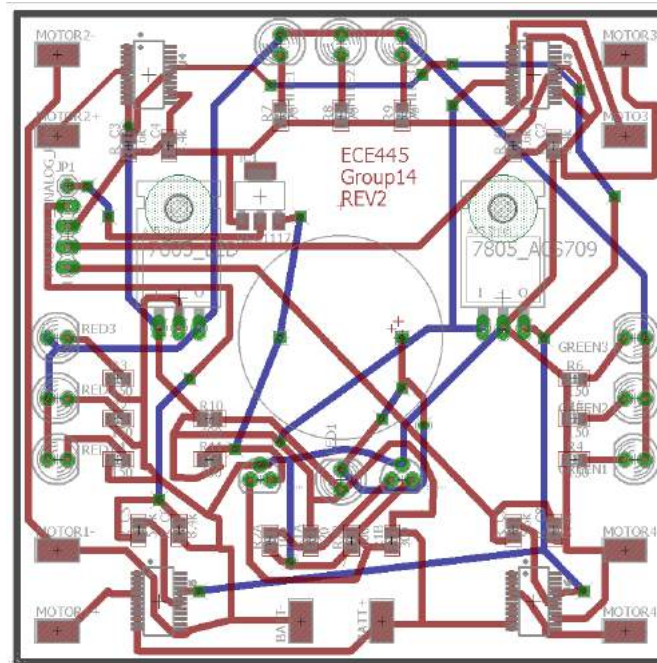


Figure 40: PCB design from EAGLE CAD

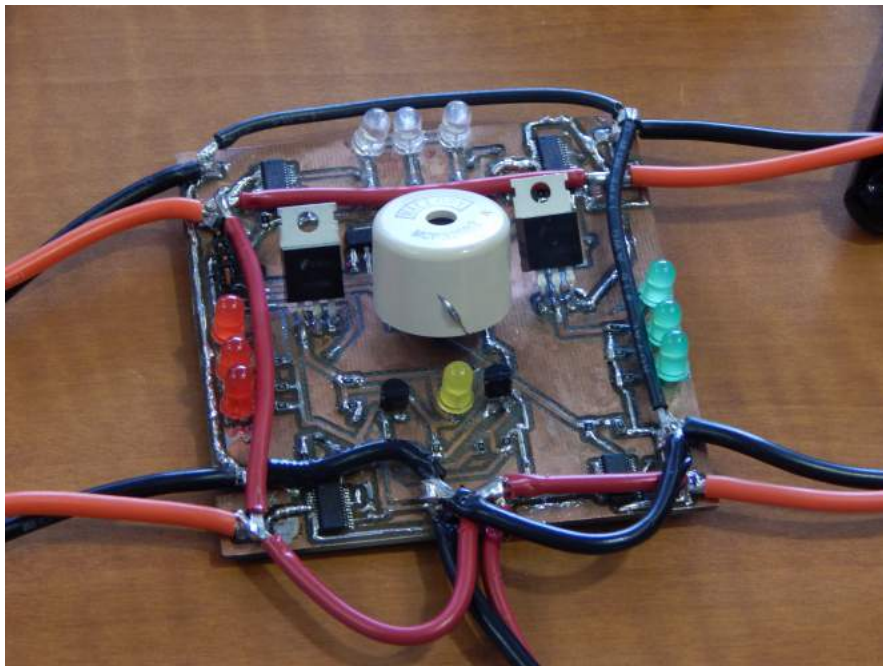


Figure 41: Completed PCB

Appendix C Code Snippets

```
typedef struct control_state{
    //holds the updated bit flags
    volatile unsigned char bUpdateFlagsShared;
    volatile unsigned short unThrottleInShared;
    volatile unsigned short unRudderInShared;
    volatile unsigned short unElevatorInShared;
    volatile unsigned short unAileronInShared;
    volatile unsigned short unGearInShared;
    unsigned char bUpdateFlags;
    unsigned int ulThrottleStart=0;
    unsigned int ulGearStart=0;
    unsigned int ulRudderStart=0;
    unsigned int ulElevatorStart=0;
    unsigned int ulAileronStart=0;

    int throttle;
    int rudder;
    int elevator;
    int aileron ;
    int gear;

    //Boolean to see if drone is armed
    bool armed = false;
    //Boolean to see if signal is good
    bool signalGood = false;
    //PID values
    double pid_roll_in, pid_roll_out, pid_roll_setpoint = 0;
    double pid_pitch_in, pid_pitch_out, pid_pitch_setpoint = 0;
    double pid_yaw_in, pid_yaw_out, pid_yaw_setpoint = 0;
    //Angles values for the sensor
    double angleX = 0;
    double angleY = 0;
    double angleZ = 0;
}control_state_t;
```

Figure 42: Control State Code

```

//Function to read in the signals
uint8_t COMMS::readSignals(control_state *ctrl)
{
    if(ctrl->bUpdateFlagsShared)
    {
        //turn off interrupts to take local copies of the sharead variables
        noInterrupts();
        //take local copy of which channels were updated
        ctrl->bUpdateFlags = ctrl->bUpdateFlagsShared;
        if(ctrl->bUpdateFlags & THROTTLE_FLAG)
        {
            ctrl->throttle = ctrl->unThrottleInShared;
        }
        if(ctrl->bUpdateFlags & AILERON_FLAG)
        {
            ctrl->aileron = ctrl->unAileronInShared;
        }
        if(ctrl->bUpdateFlags & ELEVATOR_FLAG)
        {
            ctrl->elevator = ctrl->unElevatorInShared;
        }
        if(ctrl->bUpdateFlags & RUDDER_FLAG)
        {
            ctrl->rudder = ctrl->unRudderInShared;
        }
        ctrl->bUpdateFlagsShared = 0;
        interrupts();//Turn interrupts back on
    }
    ctrl->bUpdateFlags = 0;
}

```

Figure 43: Communication Code - Read Signals

```

//Function to calculate the value read by the receiver
void calcThrottle()
{
    if (digitalRead(THROTTLE_PIN) == HIGH)
    {
        my_state.ulThrottleStart = micros();
    }
    else
    {
        my_state.unThrottleInShared = (short)(micros() - my_state.ulThrottleStart);
        my_state.bUpdateFlagsShared |= THROTTLE_FLAG;
    }
}

```

Figure 44: Communication Code - Calculate Signal

```

//Set motor velocities in terms of milliseconds
uint8_t MOTORS::updateMotors(int front, int back, int right, int left)
{
  /*
   * m0 = front left = throttle + pid_roll_out
   * m1 = front right= throttle + pitch
   * m2 = back right = throttle - pid_roll
   * m3 = back left = throttle - pid_pitch
   */
  ESC1.writeMicroseconds(front);
  ESC2.writeMicroseconds(back);
  ESC3.writeMicroseconds(right);
  ESC4.writeMicroseconds(left);
}

```

Figure 45: Motor Control Code

```

//Function to update the motor values
void updateCont()
{
  int m0, m1, m2, m3;
  int throttle_value;
  setpoint_update();
  pid_update();
  pid_compute();
  //Mapping values to useable range
  throttle_value = map(my_state.throttle, SIGNAL_MIN, SIGNAL_MAX, MOTOR_MIN, MOTOR_MAX);
  m0 = throttle_value + my_state.pid_roll_out;
  m1 = throttle_value + my_state.pid_pitch_out;
  m2 = throttle_value-50 - my_state.pid_roll_out;
  m3 = throttle_value - my_state.pid_pitch_out;
  my_motors.updateMotors(m0, m1, m2, m3);
  //Serial.println(m0);
  my_motors.updateServos(my_state.rudder);
}

```

Figure 46: Updating motors with controller

Appendix D CAD models and schematics

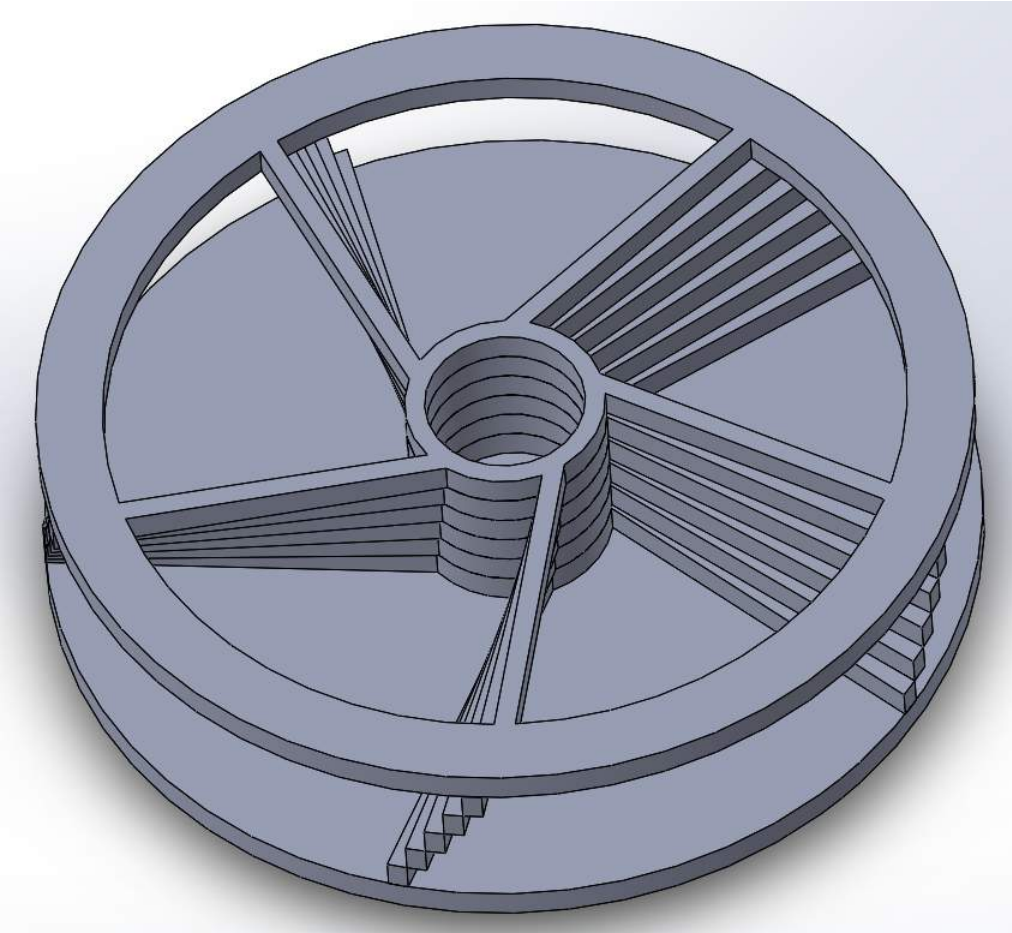


Figure 47: Counter-clockwise impeller



Figure 49: Counter-clockwise housing

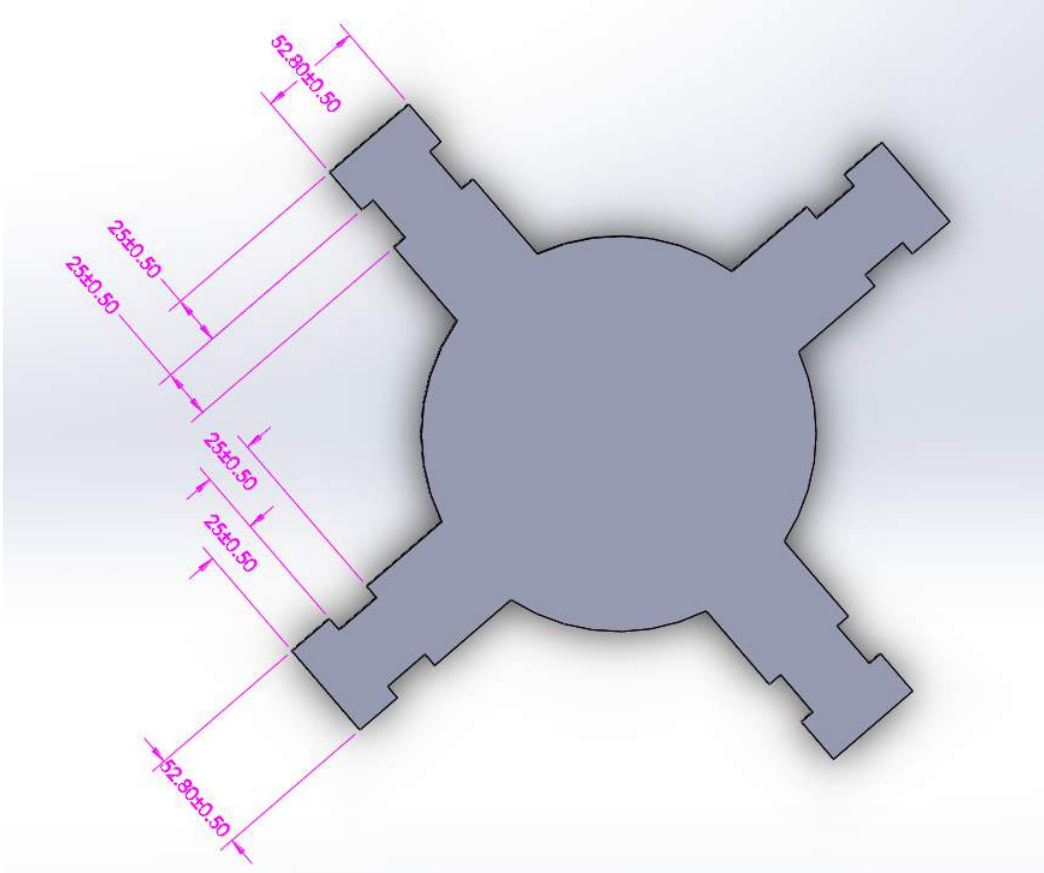


Figure 51: Frame

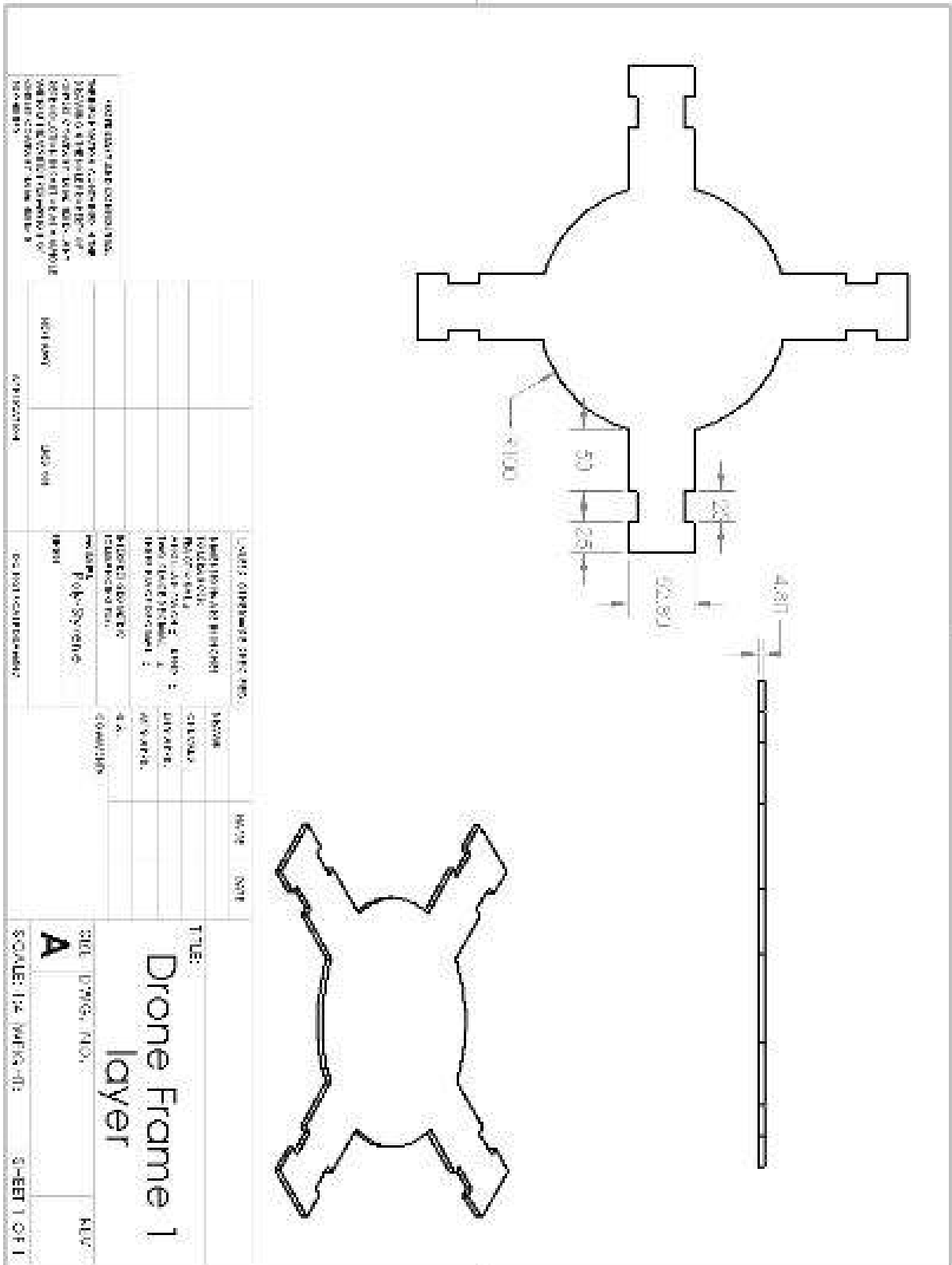


Figure 52: Frame

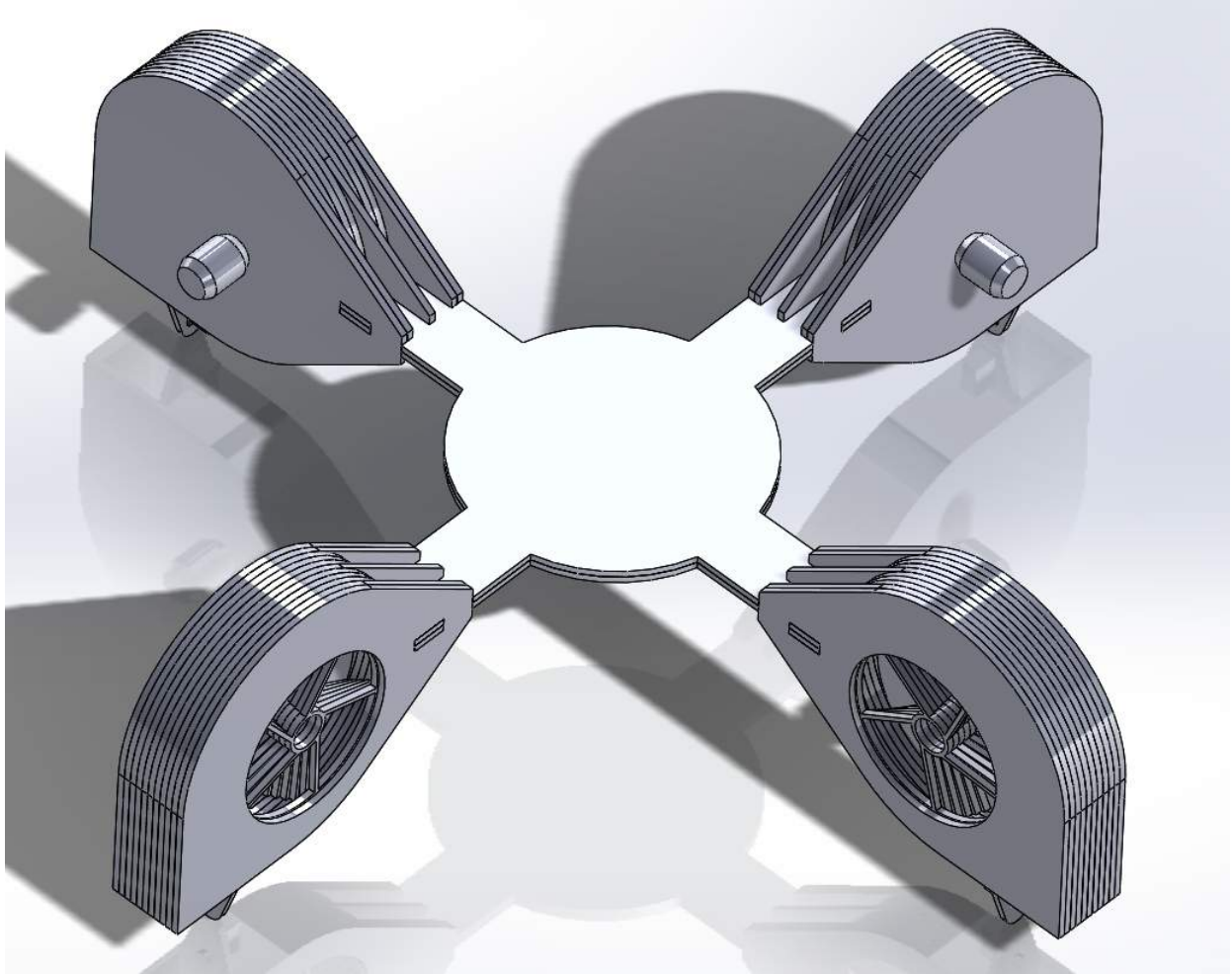


Figure 53: Isometric view of Drone CAD

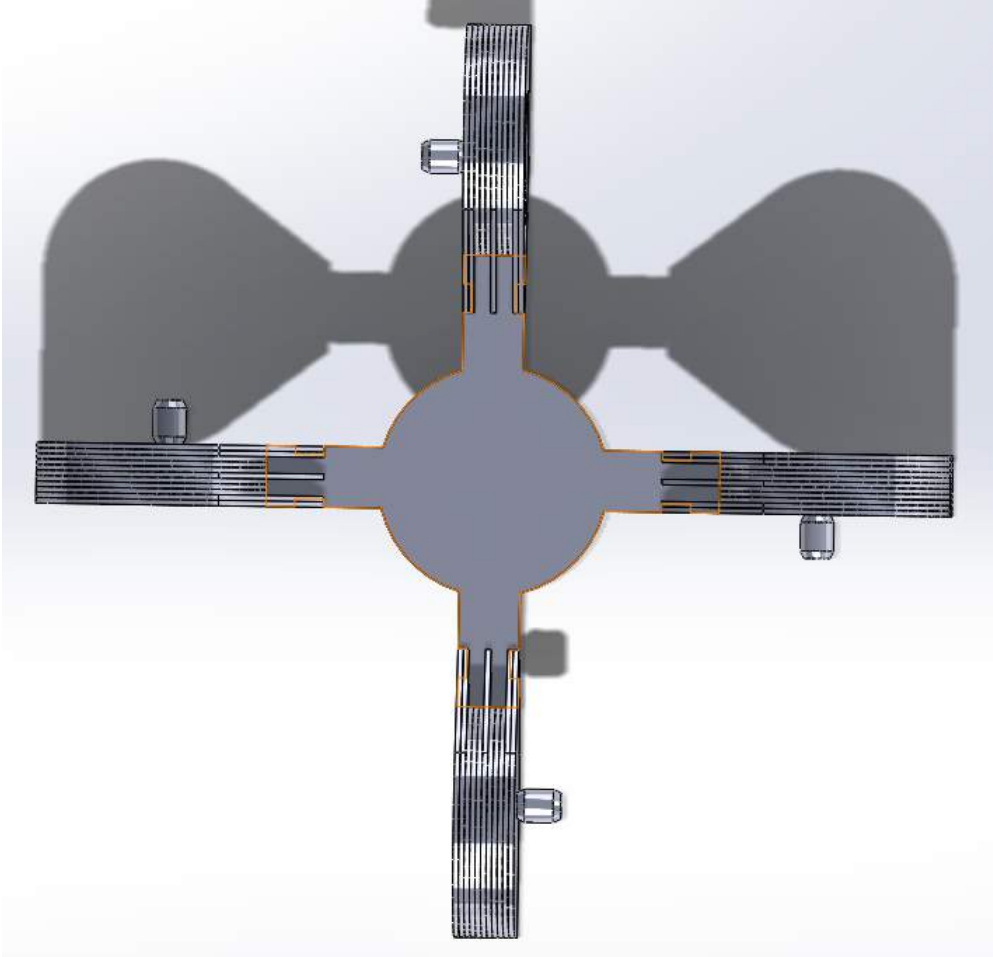


Figure 54: Top View of Drone CAD

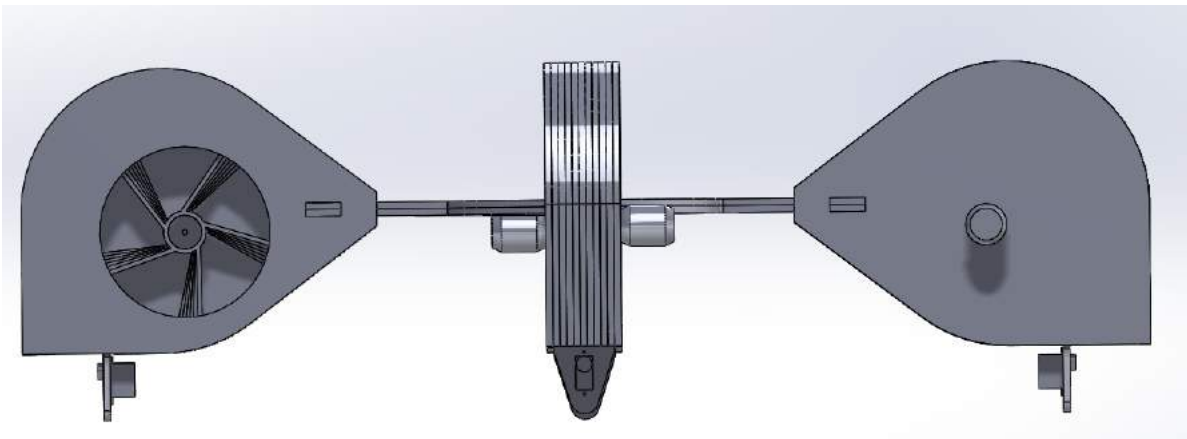


Figure 55: Side View of Drone CAD

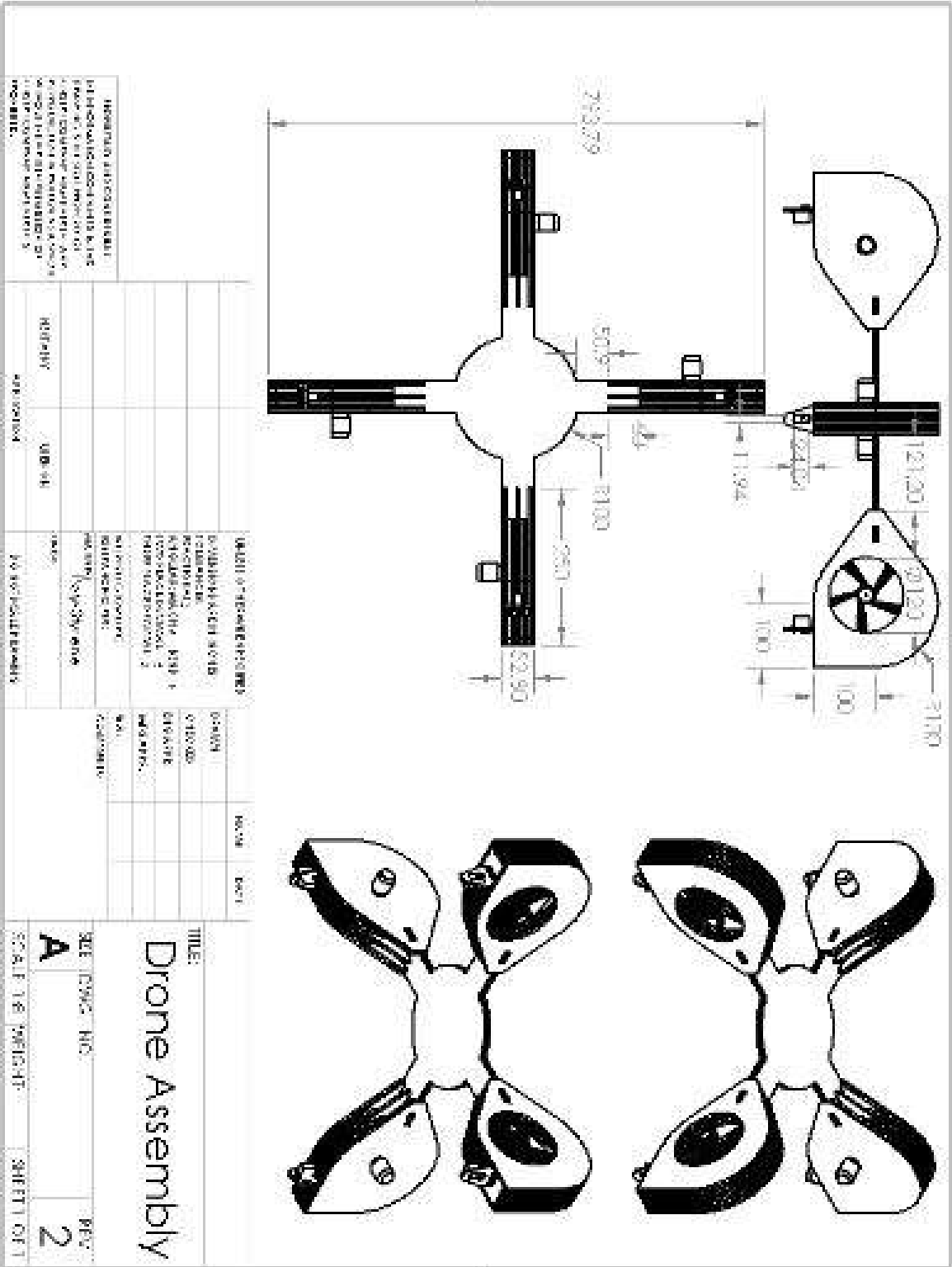


Figure 56: Drone assembly

Appendix E Sources

References

- [1] Shepard, Sam. "This is NOT a Propeller", YouTube, 2016. [Online]. Available: <https://www.youtube.com/watch?v=wg8ZbiC9IBw>. [Accessed: 22- Mar-2016].
- [2] Amazon.com, "Amazon Price Air", 2016. [Online]. Available: <http://www.amazon.com/b?node=8037720011>. [Accessed: 03-Feb-2016].
- [3] "Aircraft principal axes", 2016. [Online]. Available: https://en.wikipedia.org/wiki/Aircraft_principal_axes
- [4] "How do quadcopters and multicopters fly", 2016 [Online]. Available: <http://learnrobotix.com/uavs/quadcopter-basics/how-do-quadcopters-fly.html>
- [5] "Centrifugal fan", 2016 [Online]. Available: https://en.wikipedia.org/wiki/Centrifugal_fan
- [6] Spektrumrc.com, "DX6i DSMX 6-Channel Full-Range without Servo MD2 (SPM6610): Spektrum - The Leader in Spread Spectrum Technology", 2016. [Online]. Available: <http://www.spektrumrc.com/Products/Default.aspx?ProdID=SPM6610>. [Accessed: 28- Feb- 2016].
- [7] Spektrumrc.com, "AR600 6-Ch Sport DSMX Receiver (SPMAR600): Spektrum - The Leader in Spread Spectrum Technology", 2016. [Online]. Available: <http://www.spektrumrc.com/Products/Default.aspx?ProdId=SPMAR600>. [Accessed: 28- Feb- 2016].
- [8] Arduino.cc, "Arduino - Compare", 2016. [Online]. Available: <https://www.arduino.cc/en/Products/Compare>. [Accessed: 28- Feb-2016].
- [9] "MPU6050 DMP" Jeff Rowberg [Online] Available: <https://github.com/jrowberg/i2cdevlib>

- [10] Clarage.com, 2016. [Online]. Available: <http://www.clarage.com/docs/fan-engineering-letters/fan-performance-characteristics-of-centrifugal-fans—fe-2400.pdf?sfvrsn=2>. [Accessed: 05- May- 2016].
- [11] "RCArduino: How To Read Multiple RC Channels", Rcarduino.blogspot.com, 2012. [Online]. Available: <http://rcarduino.blogspot.com/2012/04/how-to-read-multiple-rc-channels-draft.html>. [Accessed: 04- May- 2016].
- [12] "Technical LED Color Chart", Oksolar.com, 2016. [Online]. Available: [http://www.oksolar.com/led/led color chart.htm](http://www.oksolar.com/led/led%20color%20chart.htm). [Accessed: 22- Mar- 2016].
- [13] "TL43xx Precision Programmable Reference", Texas Instruments, 2016. [Online]. Available: <http://www.ti.com/lit/ds/symlink/tl431.pdf>. [Accessed: 22- Mar- 2016].
- [14] "Arduino Current Sensor", Hacktronics.com, 2016. [Online]. Available: <http://www.hacktronics.com/Tutorials/arduino-current-sensor.html>. [Accessed: 22- Mar- 2016].
- [15] Arduino.cc, "Arduino - ArduinoBoardDue", 2016. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardDue>. [Accessed: 16- Feb- 2016].
- [16] "ACS709-Datasheet", Allegromicro.com, 2016. [Online]. Available: [http://www.allegromicro.com/ /media/Files/Datasheets/ACS709-Datasheet.ashx](http://www.allegromicro.com/media/Files/Datasheets/ACS709-Datasheet.ashx). [Accessed: 29- Mar- 2016].
- [17] "Modelling and control of quadcopter", Teppo Luukkonen, school of Science Aalto University, August 22, 2011 [Online]. Available: <http://sal.aalto.fi/publications/pdf-files/eluu11public.pdf>
- [18] "MODELADO, DISEÑO, CONSTRUCCIÓN Y CONTROL BORROSO DE UN HELICOPTERO QUADROTOR", Carlos Mikel Esparza Martinez de Luco, Jorge Elso Torralba, Maria Jose Perez-Ilzarbe Serano,Pamplona, 24 de febrero de 2012 [Online]
- [19] "The quad-copter control the orientation" [Online]. Available: <http://theboredengineers.com/2012/05/the-quadcopter-basics/>
- [20] "PID", multiwii.com, 2016 [Online]. Available:<http://www.multiwii.com/wiki/?title=PID>

- [21] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online].
Available: <http://www.ieee.org/about/corporate/governance/p7-8.html>.
[Accessed: 05- Feb- 2016].
- [22] Faa.gov, "Model Aircraft Operations", 2016. [Online].
Available: [https://www.faa.gov/uas/model aircraft /](https://www.faa.gov/uas/model%20aircraft/). [Accessed: 05- Feb- 2016].
- [23] Horizon Hobby, Lithium Polymer Batteries, 2016. [Online].
Available: <https://www.horizonhobby.com/pdf/EFL-LiPoSafetyWarnings.pdf>. [Accessed: 15-Feb-2016]
- [24] Faa.gov, "Unmanned Aircraft Systems (UAS) Registration", 2016. [Online]. Available: <https://www.faa.gov/uas/registration/>. [Accessed: 28-Feb- 2016].