UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Implementación y desarrollo de plataforma tipo gimbal para pruebas de control de drones de mediana potencia

Trabajo de graduación presentado por Steven Josué Castillo Lou para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

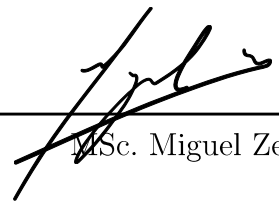
Implementación y desarrollo de plataforma tipo gimbal para pruebas de control de drones de mediana potencia

Trabajo de graduación presentado por Steven Josué Castillo Lou para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

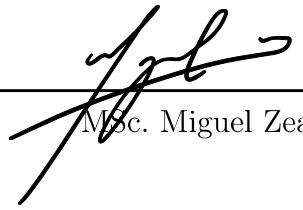
2022

Vo.Bo.:

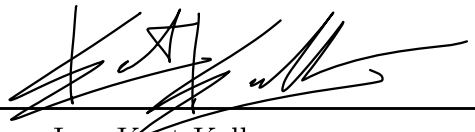
(f) 

MSc. Miguel Zea

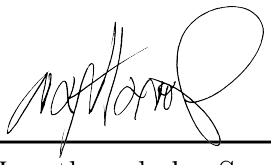
Tribunal Examinador:

(f) 

MSc. Miguel Zea

(f) 

Ing. Kurt Kellner

(f) 

Ing. Jonathan de los Santos

Fecha de aprobación: Guatemala, 5 de enero de 2022.

Índice

Lista de figuras

Lista de cuadros

Resumen

El siguiente trabajo presenta el desarrollo de un cuadricóptero que puede ser acoplado a una plataforma de pruebas en forma de gimball. Dicha plataforma presenta 3 grados de libertad, los cuales corresponden a los ángulos de vuelo (roll, pitch, yaw) así como también esta plataforma posee la capacidad de medir el rendimiento del comportamiento del vehículo abordo por medio de encoders rotativos incrementales.

Para lograr esto, lo primero que se realizó fue el diseño de un marco para el dron el cual fue diseñado en el software CAD de Autodesk Inventor. A lo largo del proyecto se realizaron 3 iteraciones para este diseño. En cada una de esas iteraciones se consideraron los límites físicos de la estructura, el tamaño de los motores sin escobillas que se utilizaron, los cuales fueron los Hitec de $390Kv$ y las hélices para dichos motores fueron de 8 pulgadas de largo (8045). La iteración final de este diseño corresponde al marco del dron F366, el cual tiene una dimensión longitudinal entre los ejes del motor de 36.6 centímetros. Cabe mencionar que, los 4 brazos del dron fueron fabricados por medio de impresión 3D con plástico ABS y las bases superior e inferior fueron fabricadas en acrílico. Por medio de un experimento se determinó que la fuerza de empuje de los motores *brushless* con la hélice de 8 pulgadas fue de 261 gramos(fuerza).

Se logró encontrar que el controlador de vuelo automático Pixhawk 1, fue adecuado para la aplicación principal de este proyecto, ya que permitía su programación en un herramienta de Matlab/Simulink, la cual permitió de manera rápida y eficiente el cambio de velocidad de los motores y la implementación de controladores dentro de la plataforma. Por último se realizó el análisis de esfuerzos tanto teóricamente como por medio de simulaciones, únicamente para las piezas impresas en 3D. Con los resultados obtenidos de dichos análisis se obtuvo un factor de seguridad para los brazos en ABS de aproximadamente 11. Por lo tanto, se concluye que se logró adaptar un controlador de vuelo comercial abordo de un marco acorde a la plataforma.

Abstract

The following work presents the development of a quadcopter that can be mounted to a gimball test platform. This platform has 3 degrees of freedom, which correspond to the flight angles (roll, pitch, yaw) as well this platform has the ability to measure the performance of the vehicle's behavior on board by rotary incremental encoders.

To achieve this, the first thing that was done was the design of a frame for the drone which was designed in Autodesk Inventor CAD software. Throughout the project, 3 iterations were carried out for this design. In each of these iterations the physical limits of the structure were considered, the size of the brushless motors used, which were the Hitec of 390 Kv and the propellers for these motors were 8 inches long (8045). The final iteration of this design corresponds to the F366 drone frame, which has a longitudinal dimension between the motor axes of 36.6 centimeters. It is worth mentioning that the 4 arms of the drone were manufactured by 3D printing with ABS plastic and the upper and lower bases were manufactured in acrylic. Through an experiment it was determined that the thrust force of the brushless motors with the 8-inch propeller it was 261 grams(force).

It was found that the Pixhawk 1 automatic flight controller was suitable for the main application of this project, since it allowed its programming in a Matlab/Simulink tool, which quickly and efficiently allowed the speed change of the engines. and the implementation of controllers within the platform. Finally, the stress analysis was performed both theoretically and through simulations, only for the 3D printed parts. With the results obtained from these analyzes, a safety factor for the ABS arms of approximately 11. It is concluded that it was possible to adapt a commercial flight controller on board a frame according to the platform.

CAPÍTULO 1

Introducción

Los vehículos aéreos multi-rotores no tripulados poseen una alta capacidad de maniobra durante su vuelo y se caracterizan por poder planear, despegar, volar y aterrizar verticalmente en áreas pequeñas de manera muy ágil. Por consiguiente las plataformas de dichos vehículos suelen ser complejas y extremadamente caras debido a su particular forma de vuelo, ya que mientras mas compleja sea la tarea o misión que se le asigne al dron, mayor será la dificultad para controlarlo de manera segura. Esto da a lugar a la necesidad de plataformas en las cuales se puedan experimentar con dichos vehículos dentro de un ambiente seguro, en donde no exista posibilidad alguna de daños físicos tanto al usuario como al vehículo. Para evitar accidentes catastróficos, se deben realizar algunas pruebas antes del vuelo y para lograr que este sea estable, se deben ajustar los parámetros de control adecuados [1].

El presente trabajo es la continuación de la fase uno del proyecto de graduación "Plataforma de pruebas y ajustes de sistemas de control para vehículos multirrotores". En la fase inicial se realizó el diseño y la fabricación (en aluminio) de la plataforma de tres grados de libertad, la cuál fue pensada y diseñada para utilizarla con un cuadricóptero "Parrot AR 2.0". Dicho multirrotor presentó dificultades y limitaciones de movimiento al momento de ser acoplado a la plataforma debido a la baja potencia de los motores a bordo. Debido a esas limitaciones la parte del controlador de vuelo no se trabajó.

El propósito de esta fase consistió en la implementación de un dron y un conjunto de herramientas de software acorde a la plataforma de suspensión tipo cardán de tres grados de libertad. Dicha plataforma, junto con la implementación del autopiloto, será utilizada para para diseñar, configurar y evaluar diferentes técnicas y/o algoritmos de sistemas de control enfocado al manejo de la orientación de un cuadricóptero. Por lo tanto, este proyecto se dividió en 2 módulos principales: el primero de ellos corresponde al hardware de la estructura (diseño y manufactura del marco del dron), el segundo es sobre el software utilizado y este se basa en cómo se trabajó la implementación del controlador de vuelo utilizado.

El módulo de hardware contempló varios componentes que conformaron al dron dentro de la estructura de acuerdo a las limitaciones físicas de la plataforma, como por ejemplo el pandeo que esta presentó al momento de montar el dron debido al material de aluminio de los ejes de movimiento y los límites de las distancias máximas que se tenían dentro de dichos ejes, para considerar que largos máximos de los brazos del marco del dron se podían trabajar y la selección de hélice mas óptima de acuerdo a los motores sin escobillas seleccionados y dadas esas restricciones de longitudes límite.

Para mejorar la respuesta del movimiento del dron dentro de la plataforma considerando los problemas que se tuvieron en la primer fase, se decidió partir con la selección de un motor que posea una potencia suficiente en cuanto a la fuerza de empuje que estos pueden generar. Los motores sin escobillas seleccionados dada la disponibilidad dentro del campus y sus principales ventajas fueron los motores Hitec 4108 de 390kv. La hélice mas óptima que se pudo seleccionar, dadas las restricciones ya mencionadas fue una de ocho pulgadas (8045). Dicho motor con la hélice a su máxima velocidad pudo generar una fuerza de empuje de aproximadamente 261 gramos fuerza.

La fuerza de empuje generada por los motores seleccionados considerando los pesos de los demás componentes del dron y teniendo en cuenta que este no necesita volar solo poder maniobrar dentro de la plataforma fue suficiente para hacerlo girar dentro de la plataforma de manera óptima, dados los resultados de la sección correspondiente. Cabe mencionar que para la obtención de dicha fuerza se fabricó una pequeña maqueta de madera en forma de escuadra (mismos largos), ubicando al centro de la hélice en un extremo y el apoyo en el otro extremo. El apoyo del otro extremo donde no va el motor, fue colocado en el centro de una balanza electrónica y al variar la velocidad hasta su máxima capacidad se obtuvo la medición. En cuanto al análisis de esfuerzos de cada brazo para el material ABS del marco se obtuvo un factor de seguridad teórico de 11 dado el esfuerzo de flexión generado de 1.18MPa. El factor de seguridad teórico coincidió con el obtenido en las simulaciones de análisis de elementos finitos de Ansys y con el análisis de estrés de Inventor.

Por otro lado, el consumo nominal de corriente de acuerdo a la carga de la hélice y el marco del dron final fue de aproximadamente 5.10 amperios. Dado ese consumo se trabajó con una fuente ZYLTECH modelo S-360-12 la cual es capaz de entregar hasta 30 amperios con con 12 voltios.

El módulo de software contempló todo lo relacionado a la implementación del controlador de vuelo “Pixhawk 1” seleccionado dada su disponibilidad y su viabilidad de uso del paquete de soporte de PX4 con el entorno de programación visual de Matlab/Simulink.

CAPÍTULO 2

Antecedentes

En los últimos años ha habido un avance importante en el uso de vehículos aéreos multi-rotores no tripulados (UAV por sus siglas en ingles), comúnmente conocidos como drones. Estos vehículos son conocidos por tener características únicas al poder realizar y ejecutar tareas que muchas veces son muy peligrosas para los seres humanos o que requieren de un tiempo muy corto para que la tarea se ejecute eficazmente. Por lo tanto, la demanda de dichos vehículos ha crecido con el paso del tiempo en importantes aplicaciones civiles y de seguridad tales como monitoreo de tráfico, edificios, tuberías, cultivos e incluso en misiones de seguridad y entregas a domicilio [2].

Los avances más recientes en la tecnología han potenciado el desarrollo y la operación de este tipo de vehículos. Cada vez hay mas nuevos sensores, microprocesadores y sistemas de propulsión que son mas ligeros, pequeños y eficientes, pero sobre todo mas capaces que nunca [3].

A continuación se mencionarán algunas de las plataformas existentes a nivel mundial y una a nivel local (UVG), así como también se verán los distintos enfoques que les otorgaron a cada una respectivamente.

2.1. Trabajos previos UVG

2.1.1. Plataforma de pruebas y ajustes de sistemas de control para vehículos multirrotores

Este trabajo fue realizado por Gabriel Martínez [4] en este se describe el proceso de diseño y desarrollo para la implementación de 2 plataformas de pruebas para cuadricópteros, el Parrot AR 2.0 y el mini dron Crazyflie 2.0 de Bitcraze. Para el trabajo actual la información relevante únicamente será la relacionada al Parrot 2.0, ya que este fue el dron de prueba para la plataforma tipo gimbal que se está trabajando. Dicha plataforma debido a la particularidad de su diseño, se caracteriza por permitir control únicamente en 3 grados de libertad (DOF). Dicho de otra manera solo se puede controlar la orientación del dron cuadricóptero, es decir, control de movimientos angulares como lo es alabeo, cabeceo y guiñada o también conocidos como *roll, pitch* y *yaw* en inglés.

El diseño de dicha plataforma de interés se basa en una suspensión tipo "Cardán" la cual está compuesta por 2 aros concéntricos cuyos ejes forman un ángulo recto de 90° además de poseer un último eje de rotación que será la plataforma central en donde se ensamblará el dron para las pruebas de la plataforma. Por otro lado, esta plataforma cuenta también con 3 codificadores rotacionales tipo ópticos LPD3806-600BM con los cuales es posible determinar posición, velocidad y dirección del dron ensamblado. La plataforma también incorpora anillos colectores SENRING M220-12S con el objetivo de distribuir eficazmente señales tanto de energía como de comunicación desde una plataforma estacionaría hacia una rotacional. Cabe mencionar que dichos codificadores rotaciones se utilizarán en cada uno de los 3 ejes de movimiento únicamente con el objetivo de evaluar externamente el desempeño de los sistemas de control de actitud del dron cuadricóptero. Dicho diseño de la plataforma se decidió de esa forma ya que es una de las mejores soluciones en cuanto a la seguridad de los estudiantes y del equipo utilizado.

En cuanto a las delimitaciones de requerimientos de dicha plataforma cabe decir que esta cumple con ser lo suficientemente ligera para drones de mediana potencia sin embargo no está demás verificar que dicho peso de los aros de movimiento sea lo suficientemente ligero para que este no afecte la inercia de cualquier dron que se ensamble y así poder afectar de la menor manera a las diferentes pruebas que se realicen para los algoritmos de vuelo. Posee sensores externos (encoders) con el objetivo de calibrar los sensores a bordo de controlador de vuelo y realizar experimentos con vehículos que no posean una unidad de medición inercial (IMU).

Lo que quedó pendiente de este trabajo fue implementar un dron cuadricóptero fijo a la plataforma distinto con mejores capacidades de vuelo. Este debe ser capaz de tener un controlador de vuelo con código abierto para poder ser programado e implementado eficazmente. Como recomendaciones se menciona el uso de motores en los ejes de movimiento con el objetivo de medir el desempeño de de los sistemas de control propuestos en condiciones de perturbaciones y se recomienda cambiar a codificadores rotaciones absolutos en lugar de incrementales ya que son más exactos [4].

2.2. Trabajos similares actuales

2.2.1. Development of Simulation Technologies for Assessing Multi-Rotor Unmanned Aerial Vehicle Performance in Precision Agriculture Operations

Este trabajo de maestría fue realizado por Riccardo Lazzari [5], dicho trabajo consiste en la implementación de un dron cuadricóptero con un controlador de vuelo lo suficientemente robusto capaz de controlar y manejar adecuadamente un sistema que tendrá que volar automáticamente entre hileras de viñedos bajo perturbaciones como el aire, temperatura y la intensa luz solar. El controlador de vuelo que se seleccionó, fue escogido en base a los requerimientos de producción y costo, este fue el auto-piloto Pixhawk 4. Básicamente este trabajo tiene como objetivo principal explorar las potencialidades de dicho controlador de vuelo basado en control clásico y moderno mediante un control PID y LQR respectivamente. Para la implementación y simulación de un control personalizado para este cudricóptero utilizaron algoritmos basados en "Model-Based Design" gracias al paquete de MathWorks Embedded Coder Support Package for PX4 Autopilots.

La importancia de dicho trabajo radica en darle nuevas posibilidades de aplicaciones reales para una evolución de esta plataforma tipo gimbal utilizando controladores de vuelo de código abierto como lo es el Pixhawk [5].

2.2.2. Development of the Test Platform for Rotary Wing Unmanned Air Vehicle

Este consiste en una plataforma tipo gimbal, que se utilizará para pruebas de drones multirrotores con el objetivo de medir el desempeño de distintos algoritmos de control con la característica de volar en un entorno seguro y poder así evitar la problemática de estrellar el dron y dañarlo durante estas pruebas de vuelo.

Esta plataforma se basa en el diseño de la suspensión cárdan con 3 anillos concéntricos por lo que permite un libre movimiento en los 3 grados de libertad de orientación de un dron (roll, pitch yaw) y 1 último adicional para el movimiento traslacional de la altitud (thrust). La plataforma está retroalimentada mediante los sensores abordo del controlador de vuelo del cuadricóptero, esta información puede ser visualizada mendiante una interfáz gráfica que permite evaluar el desempeño y comportamiento de los sistemas de control establecidos para cada prueba [1].

2.2.3. Experimental Validation of Quadrotors Angular Stability in a Gyroscopic Test Bench

Para este trabajo también utilizan el controlador de vuelo Pixhawk, dicha plataforma se basa en un banco de pruebas tipo giroscópico para vehículos no tripulados o UAV por sus siglas en ingles. El objetivo de dicho trabajo es realizar pruebas de estabilidad con 3 grados de libertad sin el riesgo de daños presentes en las pruebas de campo, o incluso accidentes con lesiones en los usuarios [6].

2.2.4. EUREKA Dynamics - FFT GYRO series 450

Actualmente en el mercado estadounidense se encuentra esta plataforma tipo gimbal con 3 grados de libertad. Esta plataforma es desarrollada por la empresa Eureka Dynamics y establece que es específica para pruebas de drones con la cualidad de proporciona un entorno seguro sin colisiones para diferentes tipos de drones no solo para el 450. Además posee codificadores magnéticos de alta resolución para medir ángulos de alabeo, cabeceo y guiñada con alta precisión. Los anillos deslizantes permiten una rotación libre absoluta en los 3 grados de libertad. Además, los codificadores se pueden reemplazar con motores para simular fuerzas externas.

Dicho plataforma "450 de FFT GYRO" está fabricada de fibra de carbono, material que se caracteriza por ser muy ligero y resistente. Esta ha sido diseñada específicamente para multirrotores de pequeño a mediano tamaño, está basada en el famoso marco de cuadricóptero 450. Este sistema es ideal para entrenar controladores de vuelo, ajustar parámetros de control no lineales, proyectos de postgrado de Robótica y de Vehículos Autónomos, incluso se podría mejorar controladores existentes y mejorar la dinámica de dron. Además de lo mencionado esta plataforma ofrece poder conectarse a Matlab/Simulink para revisar el desempeño del dron a bordo. La desventaja de estas plataformas comerciales es que tienen un precio excesivamente alto, el rango inicial para esta es de $3,900 que son aproximadamente treinta mil quetzales [7].

CAPÍTULO 3

Justificación

Para que los vehículos UAV no sufran ni tampoco generen daños en sus alrededores se deben realizar algunas pruebas antes de despegar del suelo, con el objetivo de asegurar que el vuelo será completamente estable y para ello se deben ajustar los parámetros de control adecuados. Como se mencionó en los antecedentes, la demanda actual de drones en aplicaciones modernas hace que sea necesario y extremadamente útil tener estas plataformas de vuelo para poder realizar pruebas y experimentación con ellos de la manera más segura posible. Sin embargo como se observó en algunos ejemplos de plataformas existentes actuales en los antecedentes, dichas plataformas son demasiado caras, inaccesibles e incluso desactualizadas para conseguirlas en nuestro Guatemala.

Por esta razón, este proyecto propone la continuación de esta plataforma (tipo gimbal) que realizó Martinez [4] en UVG en el año 2019. Es importante mencionar que para dicha plataforma quedó pendiente establecer un dron acorde a las especificaciones de la plataforma para poder hacer uso de la misma y poder así también obtener información que permita hacer una mayor experimentación para futuros trabajos. Dicha plataforma tendrá como objetivo principal ofrecer un ambiente lo suficientemente controlado que permita configurar, probar y experimentar con varios algoritmos de control tanto clásico como moderno, sin que esta sea susceptible a los accidentes de vuelo que podrían llegar a ocurrir si se configurara de manera errónea los controladores de vuelo para el dron que se instale en la plataforma.

Los controladores de vuelo que utilizan algunas de estas plataformas son controladores desactualizados tales como el Ardu-Pilot, por ejemplo. Es por ello que este trabajo tendrá un aporte a esta plataforma que propone utilizar un controlador de vuelo relativamente moderno, barato, potente y de código abierto con se pueda implementar y cargar los algoritmos de control de manera eficaz, con la visión de poder explotar sus capacidades físicas en futuros proyectos de esta índole para aplicaciones más complejas relacionadas a nuestro país, como por ejemplo, el monitoreo/riego de cultivos.

CAPÍTULO 4

Objetivos

4.1. Objetivo general

Implementar y desarrollar una plataforma tipo gimbal para pruebas de control en la orientación de drones con 3 grados de libertad (roll, pitch, yaw).

4.2. Objetivos específicos

- Implementar algoritmos de control de orientación de vuelo para el dron dentro de la plataforma tipo gimbal.
- Adaptar un controlador de vuelo para utilizarlo dentro de una interfaz gráfica automatizada (Matlab/Simulink) y realizar pruebas de control (orientación) dentro de la plataforma tipo gimbal.
- Diseño y fabricación del chasis/PCB (conexiones de potencia) del dron para utilizar dentro de la plataforma.

CAPÍTULO 5

Alcance

El alcance de esta investigación fue lograr implementar un cuadricóptero con las características físicas adecuadas de acuerdo a las limitaciones y los retos de la plataforma donde se iba a ensamblar. Cabe mencionar que dicho alcance no pretendió diseñar un controlador de vuelo desde cero, sino mas bien fue implementar uno existente en el mercado actual. La implementación del controlador de vuelo se realizó por medio de la primera versión de auto-piloto Pixhawk del fabricante PX4. Gracias a la herramienta de soporte para auto-pilotos de PX4 disponible en Matlab/Simulink fue posible acoplar dicho controlador de vuelo a el dron diseñado para su uso dentro de la plataforma tipo gimball.

Debido a la situación mundial que generó la pandemia del COVID-19, se dificultó el trabajo constante presencial que se pudo llegar a realizar en el campus. Esto provocó también limitaciones y atrasos importantes al momento de conseguir los elementos y componentes mas actualizados que eran necesarios para desarrollar un mejor control del multirrotor manufacturado, por esta razón no se trabajó con la última versión del controlador de vuelo de PX4 (Pixhawk 4).

CAPÍTULO 6

Marco teórico

6.1. Generalidades de un cuadricóptero

Un cuadricóptero es un vehículo aéreo no tripulado o UAV por sus siglas en inglés (Unmanned Aerial Vehicle). En particular, este vehículo consta de cuatro brazos los cuales poseen un motor con una hélice al final. El principio de funcionamiento de este multirrotor es similar al de un helicóptero en muchos aspectos, sin embargo, la diferencia entre estos dos radica en que el cuadricóptero realiza todas sus maniobras con el movimiento en sintonía de sus 4 motores a la vez, gracias a esto este tipo de vehículos no tripulados se caracterizan por tener una gran estabilidad.

Figura 1: Cuadricóptero DIY F450 - MiniPixhawk [8].

Las aplicaciones de los UAV's varian mucho dependiendo del tipo de nave que se tenga, por lo tanto para la correcta selección del vehículo no tripulado a utilizar se necesita conocer la aplicación en específico que se le vaya a dar a este. Las aplicaciones mas comunes para estos tipos de vehículos se presentan a continuación [2]:

- Fotografía aérea (filme, video, etc.).
- Agricultura (vigilancia de los cultivos y aspersión).
- Compañías eléctricas (inspección en líneas de alta tensión).
- Autoridades locales (inspección y control de desastres).
- Agencia de tráfico (vigilancia y control del trafico terrestre).
- Servicio oficial de cartografía (proyección de fotografía aérea).

Una de las ventajas mas importantes que poseen los cuadricópteros es su alta capacidad de vuelo estacionario, debido a que sus cuatro rotores le otorgan una gran estabilidad en el aire, incluso con perturbaciones en su entorno de vuelo como por ejemplo el viento. Gracias a esto los cuadricópteros también poseen una alta capacidad de maniobrar rápidamente en el aire. Además, aparte de su habilidad de maniobrabilidad, estos vehículos poseen una capacidad muy buena de aterrizaje vertical (VTOL, por sus siglas en inglés, "Vertical Take Off and Landing"), por ende estas capacidades dotan a los cuadricópteros de características físicas para poder realizar tareas en entornos difíciles de alcanzar para los humanos [2].

En la Figura 2 se muestra un cuadro comparativo sobre algunas de las características mas importantes que se consideran en este tipo de vehículos.

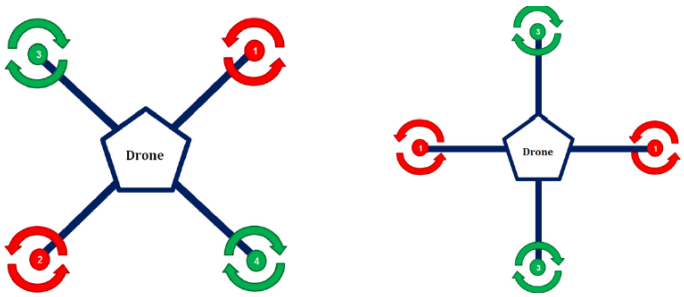
Característica	Helicóptero	Aeroplano	Dirigible	Cuadrirrotor
Capacidad de vuelo estacionario	***		****	***
Velocidad de desplazamiento	***	****	*	**
Maniobrabilidad	***	*	*	****
Autonomía de vuelo	**	***	****	*
Resistencia a perturbaciones externas	**	****	*	**
Auto estabilidad	*	***	****	**
Capacidad de vuelos verticales	****	*	**	****
Capacidad de carga	***	****	*	**
Capacidad de vuelo en interiores	**	*	***	****
Techo de vuelo	**	****	***	*

Figura 2: Característica de los principales tipos de UAV's [2].

Una de las desventajas de los cuadricópteros es la autonomía del tiempo de vuelo. Debido a que este cuenta con cuatro rotores, el requerimiento de energía es bastante elevado para la batería por ende esto hace que se reduzca el tiempo de vuelo. Por otro, lado en adición al peso de los rotores, sensores, circuitos eléctricos y la cámara (si se le coloca), la aeronave tiende a poseer menor capacidad de carga.

6.1.1. Principios de operación de los cuadricópteros

En la actualidad existen diferentes tipos de configuración del modelo para un cuadrirrotor, sin embargo las dos más comunes se caracterizan por colocar de manera simétrica los motores del dron en una configuración en **cruz ("+")** ("Plus configuration" en inglés) o en **equis ("x")** ("Cross configuration" en inglés). En una disposición en cruz la parte delantera del cuadricóptero se alinea directamente con un motor. En la disposición en equis la parte delantera del cuadricóptero se sitúa en medio de los dos motores delanteros, por ende dependiendo de cual de estas dos configuraciones se seleccione para el dron su operación de vuelo cambiará. La configuración en equis es la más utilizada y común, debido a los brazos y las hélices son menos visibles en caso de utilizar una cámara abordo orientada u otros hacia adelantes.



(a) Configuración X (b) Configuración en cruz

Figura 3: Tipos de configuraciones para un cuadrirrotor[2]

6.1.2. Movimientos del cuadricóptero

Para cualquiera de las dos configuraciones de la Figura 3, el cuadricóptero posee seis grados de libertad, tres movimientos de traslación en los ejes cartesianos (x, y, z) y tres movimientos rotacionales alrededor de los ejes principales que corresponden a los ángulos de Euler o mejor conocidos como ángulos de navegación, siendo estos ϕ, θ, ψ representando el **alabeo, cabeceo y la guiñada respectivamente (roll, pitch, yaw en inglés)**. Sin embargo, dos de los tres movimientos tradicionales no se pueden controlar directamente, por lo tanto solo se pueden controlar cuatro de los seis movimientos de forma directa. Estos cuatro movimientos son los tres ángulos de Euler y el movimiento traslación que levanta al dron del suelo conocido como "Throttle" en inglés.

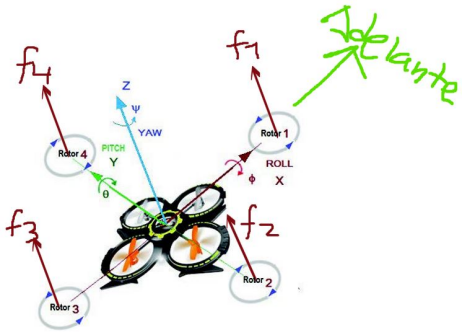


Figura 4: Movimientos principales cuadricóptero configuración "+" [9].

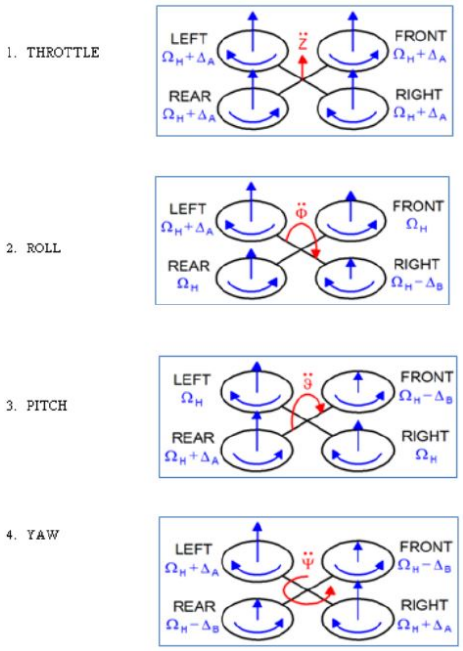


Figura 5: Detalle de movimientos principales cuadricóptero configuración "+" [10].

La descripción detallada de la operación para estos cuatro movimientos del cuadricóptero se presenta a continuación [2]:

1. Altitud (desplazamiento vertical *"throttle"*)

   - Este movimiento se basa en lograr el despegue vertical y llegar a una altura deseada. Esto se logra generando una fuerza de empuje exactamente igual en cada uno de los cuatro rotores. Si estas cuatro fuerzas son en mayor proporción el dron se eleva y si las fuerzas tienen una menor proporción sirve para disminuir su altura y aterrizarlo. Dichas fuerzas se muestran en la Figura 5.

2. Alabeo (giro alrededor del eje x *"roll"*)

   - El alabeo se logra manteniendo la misma velocidad en los rotores uno y tres, y variando la velocidad (fuerza de empuje) de los rotores dos y cuatro, dependiendo hacia donde se quiere dirigir el dron (izquierda o derecha). En la Figura 6 a) y b) se muestra giro hacia la izquierda y derecha respectivamente.

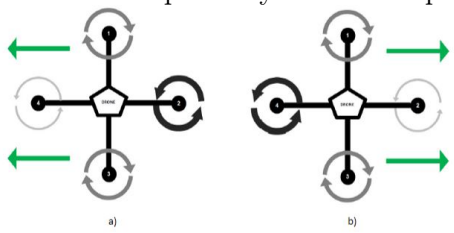


Figura 6: Maniobra de *roll*.

3. Cabeceo (giro alrededor del eje y *"pitch"*)

   - El cabeceo se logra manteniendo la misma velocidad en los rotores dos y cuatro (rotores derecha e izquierda), y en los rotores uno y tres debe de haber una variación diferente de empujes entre ellos, dependiendo hacia donde se quiere dirigir el dron (adelante o atrás), por ejemplo si la fuerza de empuje es mayor en el rotor tres que en el uno, el dron se inclinará hacia adelante 7 a).

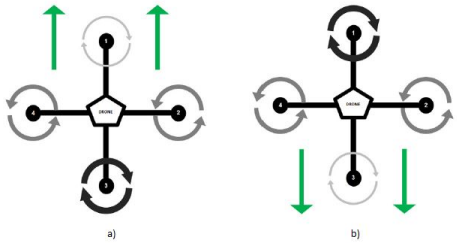


Figura 7: Maniobra de *pitch*.

4. Guiñada (giro alrededor del eje z *"Yaw"*)

   - Este último movimiento se logra efectuando un giro alrededor del eje z, ya sea en sentido horario (CW) o contrario en sentido anti-horario (CCW). En la Figura 8 a) se observa el giro en sentido horario, esto se logra ejerciendo una fuerza de empuje de la misma magnitud en los rotores dos y cuatro y en los rotores uno y tres también con una fuerza de empuje de la misma magnitud pero en mayor proporción que los anteriores, esto hará que el equilibrio en los torques de los 4 rotores en conjunto se pierda (no sea cero) y por ende el dron tenderá a girar. Lo mismo para el caso contrario giro anti-horario [2].

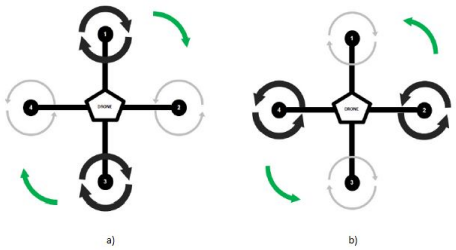


Figura 8: Maniobra de *yaw*.

6.2. Modelado dinámico del dron (control de orientación)

En esta sección se presenta el planteamiento del modelado matemático de la dinámica de un cuadricóptero. La dinámica de un cuerpo se describe estudiando las fuerzas y torques que afectan el movimiento del mismo. Como se ha discutido en secciones anteriores el movimiento de este vehículo en particular se produce por medio de 4 rotores (motores) que se controlan de manera independiente, pero en sintonía de acuerdo a la maniobra que se quiera realizar. El movimiento del quadrotor es el resultado de cambios en la velocidad de los rotores, generando así mayor torques o una mayor fuerza de empuje que haga elevar o descender al cuadricóptero, por ende las variables que se necesitan controlar son las 4 velocidades de dichos motores.

El modelo matemático que establece la dinámica del dron presenta 6 grados de libertad (DOF), dichos grados de libertad relacionan a los posibles movimientos que un dron puede realizar en el espacio, siendo estos los 3 movimientos traslacionales en los ejes principales x, y, z y los 3 ángulos de navegación ϕ, θ, ψ.

Antes de que se muestren las ecuaciones que conforman el modelo dinámico, se presentan las siguientes hipótesis sobre el modelo del dron con el objetivo de tener una mejor descripción del comportamiento del mismo [2]. Los supuestos son los siguientes:

- Las hélices se consideran rígidas.
- La estructura del dron en conjunto se supone rígida y simétrica.
- El origen del centro de referencia de la aeronave coincide con el centro de gravedad.
- El empuje y el arrastre son proporcionales al cuadrado de la velocidad de la hélice.

6.2.1. Modelo dinámico cuadricóptero configuración en "+"

Para obtener las ecuaciones de la dinámica del sistema, es necesario expresar diversas fuerzas y momentos en el sistema de referencia inercial (global), esto debido a que el cuadricóptero se mueve en diferentes transcursos de tiempo, por lo tanto, esto hace que cambie de posición con respecto a un punto de referencia que se supone fijo [9]. Son necesarios un sistema de referencia inercial y un segundo sistema de referencia para el cuerpo.

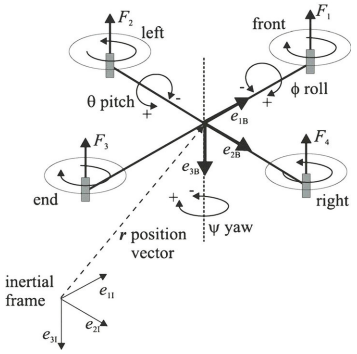


Figura 9: Sistema de referencia del cuadricóptero - configuración "cruz" [11].

Como se observa en la Figura 9, el marco de referencia inercial fijo en tierra (marco global) es $E_I(e_{1I}, e_{2I}, e_{3I})$ y el marco de referencia fijado al cuerpo del dron es $E_B(e_{11}, e_{21}, e_{31})$. Los ángulos de orientación se denominan respectivamente ángulo de guiñada (ψ "yaw" rotación alrededor del eje z), ángulo de cabeceo (θ "pitch" rotación alrededor del eje y) y ángulo de balanceo (ϕ "roll" rotación alrededor del eje x).

Por lo tanto, la posición absoluta del dron se describe mediante el vector de posición lineal $X = [x, y, z]^T$ y posición angular (orientación) $\Theta = [\phi, \theta, \psi]^T$ que corresponden a los ángulos de Euler, entonces denotando al vector de posición absoluta del dron con la letra ξ obtenemos:

$$\xi = [X^I \Theta^I]^T = [x, y, z, \phi, \theta, \psi]^T. \tag{1}$$

De igual manera, el vector de velocidad absoluta v del dron está compuesto por los vectores de velocidad lineal $V^B[m/s]$ y angular $\omega^B[rad/s]$ del cuadrirrotor con respecto al marco-B (cuerpo) [2].

$$v = [V^B \omega^B]^T = [u, v, w, p, q, r]^T. \tag{2}$$

La cinemática de un cuerpo rígido de 6 grados de libertad esta dada por la siguiente ecuación:

$$\dot{\xi} = J_\Theta v \tag{3}$$

donde $\dot{\xi}$ es el vector de velocidad generalizado con respecto al marco-Inercial y J_Θ es la matriz generalizada de conversión. En donde la matriz generalizada está constituida de 4 sub-matrices:

$$J_\Theta = \begin{bmatrix} R(\Theta) & 0_{3X3} \\ 0_{3\times 3} & T(\Theta) \end{bmatrix} \tag{4}$$

donde $R(\Theta)$ es la matriz de rotación compuesta por los ángulos de Euler (convención de giro alrededor de los ejes ZYX) y $T(\Theta)$ es una matriz generalizada de transformación. Dichas matrices se definen a continuación;

definiendo primero las matrices de rotación para el "Roll", "Pitch" y "Yaw" respectivamente.

- Matriz de rotación "roll - eje x" :

$$R(x, \phi) = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\operatorname{sen}(\phi) \\ 0 & \operatorname{sen}(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \tag{5}$$

- Matriz de rotación "pitch - eje y" :

$$R(y,\theta) = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \operatorname{sen}(\theta) \\ 0 & 1 & 0 \\ -\operatorname{sen}(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{6}$$

- Matriz de rotación "yaw - eje z":

$$R(z,\phi) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\operatorname{sen}(\theta) & 0 \\ \operatorname{sen}(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \tag{7}$$

A partir de las rotaciones presentadas anteriormente (7), (6), (5), se define la matriz de rotación completa $R(\Theta)$ del marco inercial I respecto a marco local del dron B, que relaciona las tres rotaciones sucesivas de los ángulos del sistema fijado sobre el cuadricóptero con la rotación respecto al sistema de inercia fijo en la tierra. Dicho de otra forma, a partir de los ángulos de Tait-Bryan, la orientación del dron respecto a su marco local se puede conocer en el marco global o marco inercial.

$$R(\Theta) = R(z,\psi)R(y,\theta)R(x,\phi), \tag{8}$$

$$R(\Theta) = \begin{bmatrix} \cos(\psi) & -\operatorname{sen}(\theta) & 0 \\ \operatorname{sen}(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \operatorname{sen}(\theta) \\ 0 & 0 & 0 \\ -\operatorname{sen}(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\operatorname{sen}(\phi) \\ 0 & \operatorname{sen}(\phi) & \cos(\phi) \end{bmatrix} \tag{9}$$

$$R(\Theta) = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\operatorname{sen}\theta\operatorname{sen}\phi - \operatorname{sen}\psi\cos\phi & \cos\psi\operatorname{sen}\theta\cos\phi + \operatorname{sen}\psi\cos\phi \\ \operatorname{sen}\psi\cos\theta & \operatorname{sen}\psi\operatorname{sen}\theta\operatorname{sen}\phi + \cos\psi\cos\phi & \operatorname{sen}\psi\operatorname{sen}\theta\cos\phi - \cos\psi\operatorname{sen}\phi \\ -\operatorname{sen}\theta & \cos\theta\operatorname{sen}\phi & \cos\theta\cos\phi \end{bmatrix} \tag{10}$$

Sabiendo que $\omega = (p,q,r)^T$ el vector de velocidades angulares respecto al sistema fijado al cuerpo rígido (marco B) y $\Theta = (\phi,\theta,\psi)^T$ el vector de ángulos de Euler. La relación entre $\dot{\Theta}$ y ω según [12] es:

$$\dot{\Theta} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = Tw. \tag{11}$$

Finalmente, la matriz de transformación de Euler según [12] define como:

$$\boldsymbol{T} = \begin{bmatrix} 1 & \operatorname{sen}\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\operatorname{sen}\phi \\ 0 & \operatorname{sen}\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}. \tag{12}$$

Si se sustituye la ec. (12) en ec. (11) se obtiene dicha relación entre los ángulos de Euler y las velocidades angulares.

$$\dot{\Theta} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \operatorname{sen}\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\operatorname{sen}\phi \\ 0 & \operatorname{sen}\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{13}$$

La matriz de transformación entre las velocidades angulares del sistema fijado al cuerpo y la variación de los ángulos de Tait-Bryan viene dado por la inversa de **T**:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\operatorname{sen}\theta \\ 0 & \cos\phi & \operatorname{sen}\phi\cos\theta \\ 0 & -\operatorname{sen}\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{14}$$

La dinámica de un cuerpo rígido regido por fuerzas y torques externos aplicados al centro de masa y expresados en el sistema de coordenadas ligado al cuerpo se puede obtener a través de la formulación de **Newton-Euler** según [12]. Dicha expresión se muestra a continuación:

$$\begin{bmatrix} mI_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \omega^B \end{bmatrix} + \begin{bmatrix} \omega^B \times \left(mV^B\right) \\ \omega^B \times \left(I\omega^B\right) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix}, \tag{15}$$

en donde **m** es la masa total del cuadricóptero, $I_{3\times3}$ es la matriz identidad, $\dot{V}$ es el vector de velocidad lineal del cuerpo del dron respecto al marco B, ω es la velocidad angular del cuerpo respecto al marco B, I es la matriz de inercia respeto a B, F^B y τ^B son las fuerzas y los torques generados por el dron respecto al marco local B.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{16}$$

Para un sistema de coordenadas cuyo origen coincide con el centro de masa del cuerpo, el término $\omega^B \times \left(mV^B\right)$ es igual a cero, por lo tanto la ecuación (15) se simplifica a:

$$\begin{bmatrix} mI_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \omega^B \end{bmatrix} + \begin{bmatrix} 0 \\ \omega^B \times \left(I\omega^B\right) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix}. \tag{17}$$

La fuerza principal aplicada al cuadricóptero es responsable del movimiento vertical o *throttle*, viene modelada como:

$$U_1 = \sum_{i=1}^{4} f_i = \sum_{i=1}^{4} b\Omega_i^2 \tag{18}$$

.

En donde Donde f_i es la fuerza de empuje generada por el i-ésimo rotor, y Ω_i es la velocidad angular del i-ésimo rotor.

Para modelar las demás fuerzas que generan los movimientos de cabeceo, balanceo y guiñada se debe tener en cuenta que el momento aplicado en el cuerpo a lo largo de un eje es causado por la diferencia de empujes entre los rotores ubicados en el eje perpendicular. Por lo tanto, el movimiento de cabeceo (pitch) se obtiene debido a la diferencia de empuje del rotor frontal y el rotor trasero. El movimiento de balanceo (roll) se obtiene debido a la diferencia de empuje entre el rotor de la izquierda y el rotor de la derecha. El movimiento de guiñada (yaw) se obtiene debido a la diferencia de los pares entre los dos rotores que giran en sentido horario y los dos rotores que giran en sentido anti-horario [12].

Por lo tanto los momentos que se aplican a los 3 ejes de orientación del cuadricóptero se expresan con la ecuación (19):

$$M = \begin{bmatrix} l\left(f_4 - f_2\right) \\ l\left(f_3 - f_1\right) \\ \sum_{i=1}^{4} \tau_{M_i} \end{bmatrix} = \begin{bmatrix} lb\left(\Omega_4^2 - \Omega_2^2\right) \\ lb\left(\Omega_3^2 - \Omega_1^2\right) \\ d\left(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2\right) \end{bmatrix} = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix}, \tag{19}$$

donde l es la distancia entre los motores y el centro de gravedad. Las constantes b y d corresponden a los coeficientes de empuje y de arrastre de los cuatro rotores respectivamente.

Juntando las ecuaciones (18) y (19), el modelo completo con todas las velocidades que debe realizar el cuadricóptero para lograr los cuatro movimientos básicos se muestra en la siguiente expresión (20):

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ b\left(\Omega_4^2 - \Omega_2^2\right) \\ b\left(\Omega_3^2 - \Omega_1^2\right) \\ d\left(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2\right) \end{bmatrix}, \tag{20}$$

$$\dot{\Gamma} = \begin{bmatrix} \dot{X} \\ \dot{Z} \end{bmatrix}^T = v, \tag{21}$$

$$m\dot{v} = \boldsymbol{R}F_B, \tag{22}$$

$$\dot{v} = \begin{bmatrix} \ddot{X} & \ddot{Y} & \ddot{Z} \end{bmatrix}^T, \tag{23}$$

$$I\dot{\omega} = -\omega x I\omega + \tau_B. \tag{24}$$

Reemplazando en la ecuación (**??**) las ecuaciones (18) hasta la (24) se obtiene el modelo matemático no lineal del cuadricóptero indicado entre la ecuación (25) hasta la ecuación (33) según [12].

$$\ddot{X} = (\operatorname{sen}\psi \operatorname{sen}\phi + \cos\psi \operatorname{sen}\theta \cos\phi)\frac{U_1}{m}, \tag{25}$$

$$\ddot{Y} = (-\cos\psi \operatorname{sen}\phi + \operatorname{sen}\psi \operatorname{sen}\theta \cos\phi)\frac{U_1}{m}, \tag{26}$$

$$\ddot{Z} = -g + (\cos\theta \cos\phi)\frac{U_1}{m}, \tag{27}$$

$$\dot{\phi} = p + q \operatorname{sen}\phi \tan\theta + r \cos\phi \tan\theta, \tag{28}$$

$$\dot{\theta} = q \cos\phi - r \operatorname{sen}\phi, \tag{29}$$

$$\dot{\psi} = q \operatorname{sen}\phi \sec\theta + r \cos\phi \sec\theta, \tag{30}$$

$$\dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}}qr - \frac{J_{TP}}{I_{XX}}q\Omega + \frac{lU_2}{I_{XX}} = \ddot{\Phi}, \tag{31}$$

$$\dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}}pr + \frac{J_{TP}}{I_{YY}}p\Omega + \frac{lU_3}{I_{YY}} = \ddot{\Theta}, \tag{32}$$

$$\dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}}pq + \frac{U_4}{I_{ZZ}} = \ddot{\Psi}. \tag{33}$$

Con este planteamiento es posible (en teoría) determinar la posición y orientación del cuadricóptero por doble integración de sus aceleraciones lineales y angulares. Para realizar esta operación, sólo basta conocer el punto de partida de las variables internas y las fuerzas sobre los ejes del cuadricóptero a este proceso se le denomina cinemática directa [12].

6.2.2. Simplificación de la dinámica del cuadricóptero

Considerando la implementación de esta dinámica, es importante tener en cuenta que las variables $U1, U2, U3, U4$ de la ec. (20) son combinaciones de fuerzas que generan los cuatro movimientos básicos de un cuadricóptero, dichas fuerzas son producidas por la velocidad de giro de las hélices, las cuales son dependientes de los voltajes que alimentan a los motores. Por lo que el objetivo en la realidad para estabilizar el cuadricóptero se basa en encontrar aquellos valores de voltaje que alimentan a los motores para mantener el cuadricóptero en una cierta posición u orientación requerida en la tarea. A este proceso se conoce como cinemática inversa y a diferencia de la directa esta es mucho mas compleja de calcular, es por eso que se necesita un modelo mas simplificado, para que los algoritmos de control puedan ser implementados.

Según [13], la dinámica del cuadricóptero debe simplificarse considerablemente para poder obtener la cinemática inversa de manera mas sencilla. Las ec. (25) hasta la (33) pueden reorganizarse bajo tres consideraciones según [13]. Dichas consideraciones son las siguientes:

Contribuciones angulares

Estas contribuciones angulares son producidas por productos cruzados de velocidades angulares en el dron (efecto de coriolis y giroscopios) por ende su modelado matemático es bastante complejo. Si supone que el movimiento del cuadricóptero se mantendrá en una condición de vuelo estacionario con pequeños cambios angulares (en especial roll y pitch) los términos de la ec. (31), (32) y (33) se puede aproximar que estos efectos son casi despreciables y por lo tanto son cero.

$$\begin{array}{c} \frac{I_{YY}-I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q\Omega = 0, \\ \frac{I_{ZZ}-I_{XX}}{I_{YY}} pr + \frac{J_{TP}}{I_{YY}} p\Omega = 0, \\ \frac{I_{XX}-I_{YY}}{I_{ZZ}} pq = 0. \end{array} \tag{34}$$

Aceleraciones angulares

Las aceleraciones angulares hacen referencia a los ángulos del cuadricóptero medidos en el sistema de referencia fijo al cuerpo del mismo. La matriz de transferencia T ec. (12) define la relación entre las velocidades angulares en el sistema de referencia inercial en tierra y aquellas velocidades en el sistema de referencia fijo al marco del dron. Dado que en condiciones cercanas a la estabilidad, la matriz es aproximada a la matriz identidad, las ecuaciones de aceleración se referencian directamente a las aceleraciones angulares de Euler.

$$\begin{array}{c} \ddot{\phi} = \dot{p}, \\ \ddot{\theta} = \dot{q}, \\ \ddot{\psi} = \dot{r}. \end{array} \tag{35}$$

Algoritmo de control

El algoritmo calcula las señales adecuadas a los motores. Como se sabe la cantidad de señales que se deben generar para controlar completamente al dron son cuatro. De acuerdo con esta elección, se han eliminado las ecuaciones que describen la posición XY.

$$\ddot{Z} = -g + (\cos\theta\cos\phi)\frac{U_1}{m}, \tag{36}$$

$$\begin{aligned} \ddot{\phi} = \dot{p} &= \frac{lU_2}{I_{XX}}, \\ \ddot{\theta} = \dot{q} &= \frac{lU_3}{I_{YY}}, \\ \ddot{\psi} = \dot{p} &= \frac{U_4}{I_{ZZ}}. \end{aligned} \tag{37}$$

Para este proyecto interesa controlar únicamente la orientación del dron.

6.2.3. Modelo dinámico cuadricóptero configuración en "X"

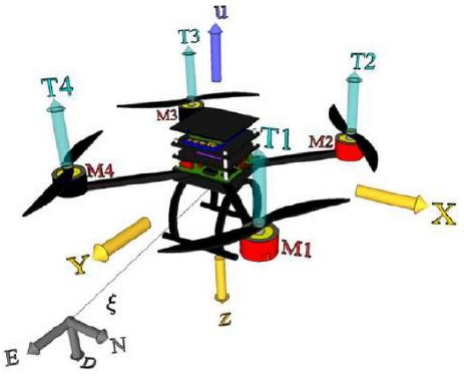


Figura 10: Sistema de referencia del cuadricóptero - configuración "equis" [12].

En esta subsección se presentan las ecuaciones para un cuadricóptero en configuración tipo equis, donde el eje de movimiento frontal del controlador esta justo entre los dos motores delanteros. Este modelo dinámico también está basado en la formulación de Newton-Euler, por lo que dicho proceso es el mismo que el de la sección anterior con la única diferencia que la ec. (20) para este caso cambia a la ec. (38) presentada a continuación:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ b\left({\Omega_4}^2 + {\Omega_3}^2 - {\Omega_1}^2 - {\Omega_2}^2\right) \\ b\left({\Omega_2}^2 + {\Omega_3}^2 - {\Omega_1}^2 - {\Omega_4}^2\right) \\ d\left({\Omega_1}^2 + {\Omega_3}^2 - {\Omega_2}^2 - {\Omega_4}^2\right) \end{bmatrix} \tag{38}$$

Por lo tanto, el modelo no lineal de la dinámica del dron en esta configuración queda como:

$$\begin{aligned} \ddot{X} &= (\operatorname{sen}\psi \operatorname{sen}\phi + \cos\psi \operatorname{sen}\theta \cos\phi)\tfrac{U_1}{m}, \\ \ddot{Y} &= (-\cos\psi \operatorname{sen}\phi + \operatorname{sen}\psi \operatorname{sen}\theta \cos\phi)\tfrac{U_1}{m}, \\ \ddot{Z} &= -g + (\cos\theta\cos\phi)\tfrac{U_1}{m}, \\ \dot{\phi} &= p + q\operatorname{sen}\phi\tan\theta + r\cos\phi\tan\theta, \\ \dot{\theta} &= q\cos\phi - r\operatorname{sen}\phi, \\ \dot{\psi} &= q\operatorname{sen}\phi\sec\theta + r\cos\phi\sec\theta, \\ \dot{p} &= \tfrac{I_{YY}-I_{ZZ}}{I_{XX}}qr - \tfrac{J_{TP}}{I_{XX}}q\Omega + \tfrac{lU_2}{I_{XX}}, \\ \dot{q} &= \tfrac{I_{ZZ}-I_{XX}}{I_{YY}}pr + \tfrac{J_{TP}}{I_{YY}}p\Omega + \tfrac{lU_3}{I_{YY}}, \\ \dot{r} &= \tfrac{I_{XX}-I_{YY}}{I_{ZZ}}pq + \tfrac{U_4}{I_{ZZ}}. \end{aligned} \tag{39}$$

6.3. Modelado del control (orientación)

Partiendo del modelo simplificado al cual se llegó en la sección anterior, los conceptos más importantes que describen la dinámica del cuadricóptero se resumen en las siguientes 3 ecuaciones (40), (41), (42):

El primer sistema de ecuaciones (40) muestra cómo el cuadricóptero acelera según los cuatro comandos básicos del movimiento.

$$\begin{cases} \ddot{Z} = -g + (\cos\theta\cos\phi)\dfrac{U_1}{m}, \\ \ddot{\phi} = \dfrac{U_2}{I_{XX}}, \\ \ddot{\theta} = \dfrac{U_3}{I_{YY}}, \\ \ddot{\psi} = \dfrac{U_4}{I_{ZZ}}. \end{cases} \tag{40}$$

El segundo sistema explica cómo esos cuatro movimientos básicos están relacionados con el cuadrado de las velocidades de las hélices.

$$\begin{cases} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ b\left(\Omega_4^2 - \Omega_2^2\right) \\ b\left(\Omega_3^2 - \Omega_1^2\right) \\ d\left(\Omega_2^2 + {\Omega_4}^2 - {\Omega_1}^2 - {\Omega_3}^2\right) \end{bmatrix}, \\ \Omega = \Omega_1 + \Omega_2 + \Omega_3 + \Omega_4. \end{cases} \tag{41}$$

La tercera ecuación toma en cuenta la dinámica de los motores. Muestra la relación entre la velocidad del rotor y el voltaje en los motores.

$$\dot{\vec{\Omega}} = -\left(\frac{K_M^2\eta r^2}{RJ_{TP}} + \frac{2d\omega_0}{J_{TP}}\right)\vec{\Omega} + \left(\frac{K_M\eta r}{RJ_{TP}}\right)\vec{v} + \left(\frac{d\omega_0^2}{J_{TP}}\right) \tag{42}$$

El algoritmo de control que se vaya a implementar para el control del dron debe tener como parámetros de entrada las lecturas de los sensores que midan la posición angular (orientación) y la tarea que se quiera realizar (4 maniobras básicas del dron). En cuanto a la salida, el algoritmo de control, debe ser capaz de determinar la señal de PWM de los cuatro motores para variar así su velocidad de acuerdo a la maniobra que se vaya a realizar. Por lo tanto el controlador puede dividirse en cuatro submódulos como se puede observar en la Figura 11, según [13].

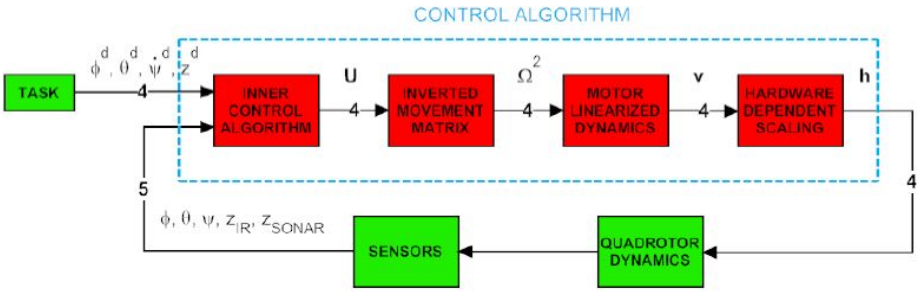


Figura 11: Diagrama de bloques del algoritmo de control [14].

6.3.1. Algoritmo de control interno (Inner Control Algorithm)

Este bloque representa el núcleo de los algoritmos de control del dron en general. Es en este bloque que se procesa la tarea y los datos de los sensores para poder proporciona una señal para cada movimiento básico que equilibra el error de posición. La ec. (40) se utiliza dentro de este bloque para transferir un comando de aceleración a un movimiento básico en el dron.

6.3.2. Matriz de movimientos invertidos (Inverted Movements Matrix)

Este segundo bloque es utilizado para calcular la velocidad al cuadrado de las hélices a partir de los cuatro señales de movimiento básicas. Ya que el determinante de la matriz de la ec. (41) es distinto de cero, esta es invertible. Dicha matriz se puede invertir para poder encontrar la relación de $\boldsymbol{U}$ a Ω^2 dada la expresión 44.

$$\begin{bmatrix} {\Omega_1}^2 \\ {\Omega_2}^2 \\ {\Omega_3}^2 \\ {\Omega_4}^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b} & 0 & \frac{-1}{2b} & \frac{-1}{4d} \\ \frac{1}{4b} & \frac{-1}{2b} & 0 & \frac{1}{4d} \\ \frac{1}{4b} & 0 & \frac{-1}{2b} & \frac{-1}{4d} \\ \frac{1}{4b} & \frac{-1}{2b} & 0 & \frac{1}{4d} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}, \tag{43}$$

$$\begin{cases} \Omega_1^2 = \frac{1}{4b}U_1 - \frac{1}{2b}U_3 - \frac{1}{4d}U_4, \\ \Omega_2^2 = \frac{1}{4b}U_1 - \frac{1}{2b}U_2 + \frac{1}{4d}U_4, \\ \Omega_3^2 = \frac{1}{4b}U_1 + \frac{1}{2b}U_3 - \frac{1}{4d}U_4, \\ \Omega_4^2 = \frac{1}{4b}U_1 + \frac{1}{2b}U_2 + \frac{1}{4d}U_4, \end{cases} \tag{44}$$

6.3.3. Dinámica linealizada del motor (Motor Linearized Dynamics)

La dinámica del motor es una ecuación diferencial altamente no-lineal. Su versión ya linealizada se representada en la ecuación 42. Existen varios métodos para poder resolver esta ecuación. El primero es simplemente resolver numéricamente la ecuación para obtener los voltajes adecuados. Según [13] el mejor método para resolver esta ecuación es caracterizar experimentalmente el comportamiento del motor y utilizar la relación deducida, este método ha sido adoptado por su sencillez y la certeza de la descripción del modelo.

6.3.4. Escalamiento dependiente del hardware (Hardware Dependent Scaling)

Este último bloque se encarga de procesar el vector de las entradas de los voltajes controlados y producir en su salida las señales PWM para los motores del cuadricóptero [14].

El algoritmo de control antes de este bloque no sabe el voltaje de la batería que se está utilizando, por lo tanto, se debe agregar una escala para considerar el voltaje adecuado. Esta variable es muy importante para determinar los comandos correctos para controlar el dron.

6.3.5. Control Clásico: PID

Uno de los algoritmos de control clásicos más utilizados es el controlador ***proporcional-integral-derivativo*** o PID por sus siglas en inglés. Esta técnica de control es bastante popular debido a su simplicidad. Este controlador es muy común en usos industriales ya que se caracteriza por poder ser implementable sin que sea necesario conocer el modelo específico de la planta a controlar, además que posee un buen desempeño aunque este no sea el más óptimo.

Las principales razones de su frecuencia de uso son las siguientes:

- Estructura simple y buen rendimiento.
- Pueden ajustarse de forma heurística, es decir, no óptima ni perfecta, pero suficientemente buena.
- No se necesita un modelo de la planta exacto para poderse implementarse.

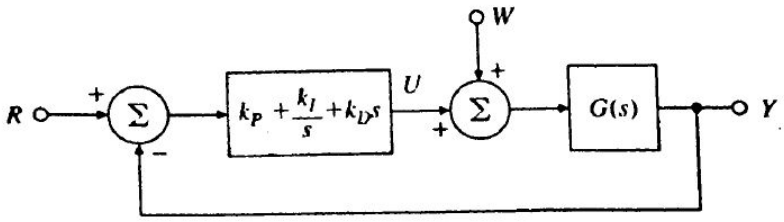


Figura 12: Estructura PID [15].

Parámetro	Tiempo de subida	Sobre-elevación y oscilaciones	Tiempo de asentamiento	Error en estado estable	Estabilidad
k_P	Decrece	Crece	Cambio pequeño	Decrece	Degrada
k_I	Decrece	Crece	Crece	Elimina	Degrada
k_D	Cambio menor	Decrece	Decrece	No tiene efecto (en teoría)	Mejora si K_D pequeño

Figura 13: Tabla para tuneo PID [15].

Dicha técnica de control se describe en el dominio de tiempo mediante la siguiente expresión:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \dot{e}(t) \tag{45}$$

Idea detrás del control PID

Esta técnica de control posee tres partes, la proporcional, integral y la derivativa.

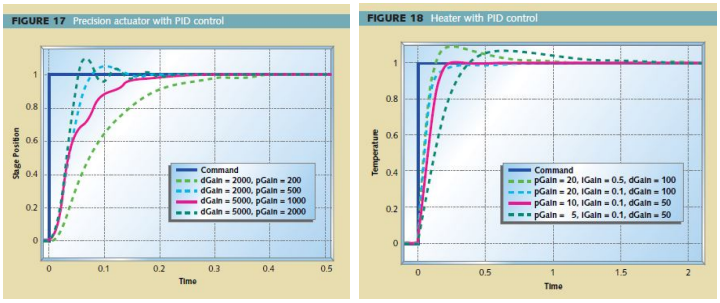


Figura 14: Idea detrás del funcionamiento PID [16]

- Parte proporcional (error presente): trata de alcanzar la referencia y hacer el error cero lo mas rápido posible.
- Parte integral (error pasado): tiene un atributo único, eliminar el error en estado estable y permite el rastreo de referencias.
- Parte derivativa (error futuro): trata de predecir el error y actúa sobre el mismo. Ayuda a reducir la sobre-elevación (overshoot) y el tiempo de establecimiento (settling time).

La acción de la parte integral (K_I) combinada con un actuador en estado de saturación puede provocar un efecto no lineal no deseado que hará disminuir el rendimiento del sistema de control. Cuando el valor de la parte integral es grande y el error cambia de signo, es necesario esperar mucho tiempo antes de que el sistema restaure su comportamiento lineal, a este fenómeno se le conoce como $wind - up$. Para evitar esto, se le agrega un saturador después de la integral para limitar su valores máximos y mínimos [13].

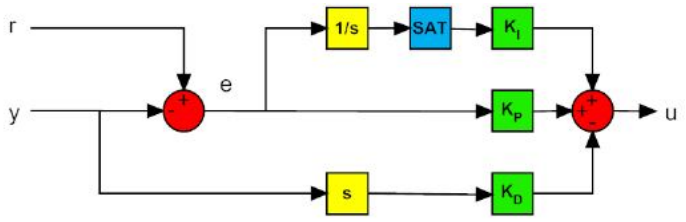


Figura 15: Estructura mejorada del PID.

6.3.6. Control de orientación con PID

Control del alabeo "roll"

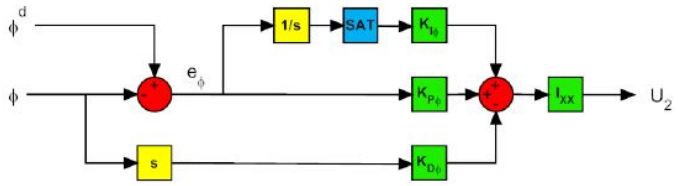


Figura 16: Diagrama de bloques del control PID roll.

En donde $\phi^d[rad]$ representa el ángulo deseado al cual se desea llegar, $\phi[rad]$ es el ángulo actual medido y $e_\phi[rad]$ es el error actual del alabeo. U2 es el componente 2 de la matriz de la ec. (20) y el bloque I_{xx} es el momento de inercia alrededor del eje x.

Control del cabeceo "pitch"

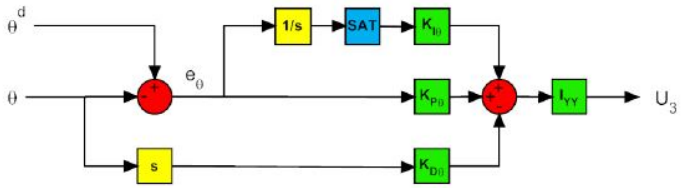


Figura 17: Diagrama de bloques del control PID pitch.

En donde $\theta^d[rad]$ representa el ángulo deseado al cual se desea llegar, $\theta[rad]$ es el ángulo actual medido y $e_\theta[rad]$ es el error actual del alabeo. U3 es el componente 3 de la matriz de la ec. (20) y el bloque I_{yy} es el momento de inercia alrededor del eje y.

Control de guiñada "yaw"

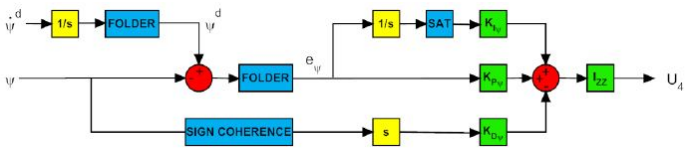


Figura 18: Diagrama de bloques del control PID yaw.

Esta estructura presenta unas diferencias con respecto a las otras 2, estas se discutirán más adelante.

6.4. Generalidades del controlador Pixhawk PX4

Este controlador de vuelo se caracteriza por ser un auto-piloto de bajo costo muy bueno enfocado en el uso académico, industrial o incluso como pasatiempo. Dicho de otra forma, este es un piloto automático de código abierto que se enfoca en el control autónomo de drones específicamente. En cuanto a la historia, el PX4 nace en 2008 por Lorenz Meier que trabajaba en el Instituto Federal Suizo de Tecnología [17].

Este controlador de vuelo ha alcanzado un nivel que permite que la codificación de una tarea avanzada no requiera saber a profundidad cómo diseñar un piloto automático en sí, sino que para implementar soluciones de control automático se necesita tener un conocimiento medio de teoría de control y nivel medio-alto en programación.

Figura 19: Pixhawk 1 - FMUv2.0 [8].

Cada diseño de este auto-piloto se nombra utilizando la designación FMUvX (Flight Management Unit Version X) los números de FMU más altos indican que la placa es más reciente, pero puede no indica mayores capacidades. El FMUv5 es la última versión producida y es la que se utiliza en el Pixhawk 4. El Pixhawk 4 es la última actualización de la familia de controladores de vuelo PX4, esta familia de auto-pilotos se basa en la tecnología de procesador avanzada de STMicroelectronics, tecnología de sensores de Bosch e InvenSense y un sistema operativo en tiempo real NuttX, que ofrece un buen rendimiento, flexibilidad y confiabilidad para controlar vehículos autónomos. La Figura 19 representa la tarjeta del Pixhawk 1 y en dicha imagen se puede observar todos sus puertos frontales que se necesitan conectar para armar el dron [5].

6.4.1. Especificaciones técnicas del PX4 FMUv2

1. Procesador:
    - STM32F427 de 32 bits Cortex-M4F (abre una nueva ventana) núcleo con FPU.
    - 168 MHz.
    - 256 KB de RAM.
    - Flash de 2 MB.
    - Coprocesador a prueba de fallos STM32F103 de 32 bits.
2. Sensores:
    - Giroscopio ST Micro L3GD20H de 16 bits.
    - Acelerómetro / magnetómetro ST Micro LSM303D de 14 bits.
    - Acelerómetro / giroscopio de 3 ejes Invensense MPU 6000.
    - Barómetro MEAS MS5611.
3. Interfaces:
    - 5x UART (puertos serie), uno con capacidad de alta potencia, 2x con control de flujo de HW.
    - 2x CAN (uno con transceptor interno de 3.3V, uno en conector de expansión).
    - Entrada compatible con Spektrum DSM / DSM2 / DSM-X® Satellite.
    - Entrada y salida compatible con Futaba S.BUS®.
    - Entrada de señal de suma PPM.
    - Entrada RSSI (PWM o voltaje).
    - I2C.
    - SPI.
    - Entradas ADC de 3,3 y 6,6 V.
    - Puerto microUSB interno y extensión de puerto microUSB externo. [18]

6.4.2. Arquitectura general del PX4

Para comunicación externa, el PX4 utiliza un protocolo de mensajería ligero llamado MAVLink, diseñado específicamente para el ecosistema de drones. Este protocolo brinda la posibilidad de comunicarse con estaciones terrestres e integrar la placa del controlador de vuelo con otros componentes, como ordenadores complementarios, cámaras habilitadas, sensores de proximidad, etc [5].

La arquitectura de software de alto nivel de PX4 incluye módulos para almacenamiento, conectividad externa, controladores, bus de mensajes de publicación-suscripción uORB (micro Object Request Broker) y componentes de pila de vuelo. Algunos de los módulos e interfaces de la arquitectura general del PX4 se pueden integrar con el paquete de soporte UAV Toolbox para pilotos automáticos PX4 que se mencionará más adelante [19].

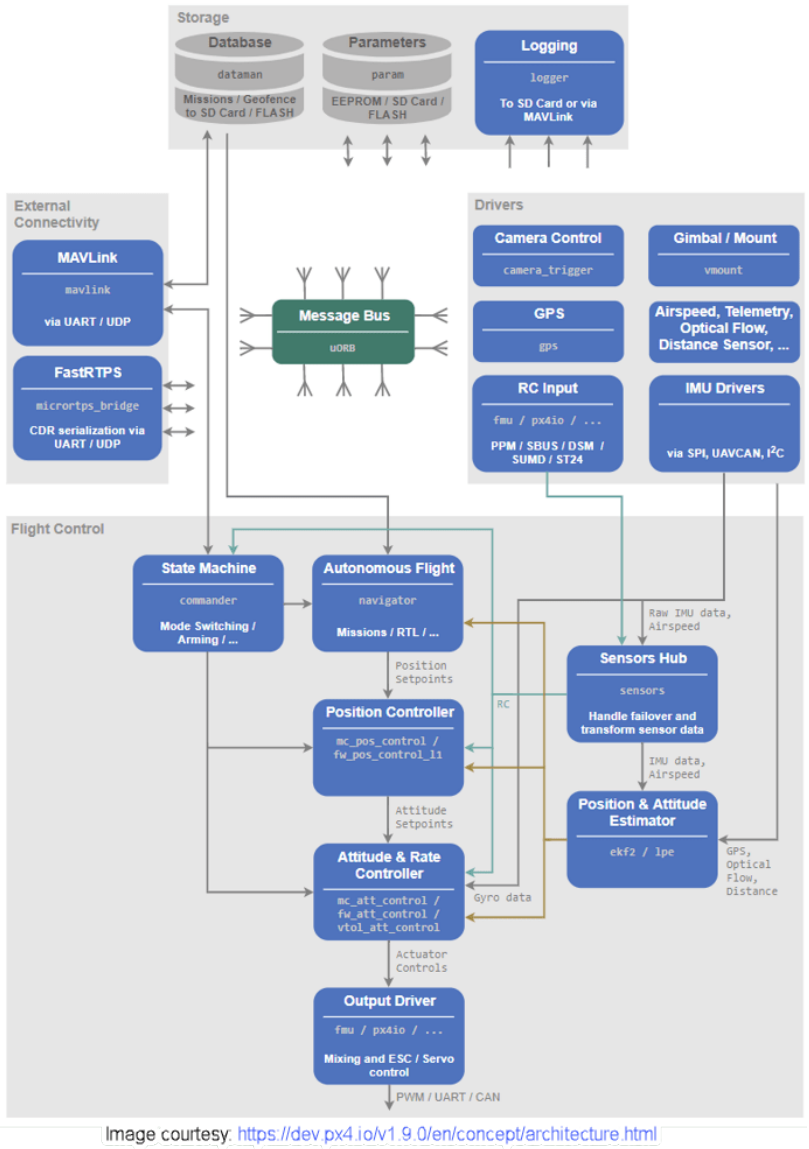


Image courtesy: https://dev.px4.io/v1.9.0/en/concept/architecture.html

Figura 20: Arquitectura general del PX4 [19].

6.4.3. Pixhawk Toolbox Matlab/Simulink UAV Toolbox Support Package for PX4 Autopilots

Gracias a las herramientas actuales que posee el entorno de programación visual de Matlab/Simulink, se encontró que modificar el firmware del controlador de vuelo es posible mediante el paquete de soporte para vehículos no tripulados PX4 que desarrolló MathWorks. Cabe mencionar que esta opción es eficiente ya que se caracteriza por no tener que entrar en mucho detalle sobre los niveles mas bajos de programación que este controlador de vuelo posee.

En la investigación académica, el entorno MATLAB / Simulink se utiliza con bastante frecuencia como herramienta principal para el modelado de sistemas y diseño de control. En particular, MATLAB/Simulink es la herramienta estándar para explotar el enfoque de diseño basado en modelos **Model Based Design (MBD)** que consiste en el desarrollo de software embebido, a partir de modelos de bloques [20].

Cabe mencionar que este paquete de soporte para este tipo de auto-piloto funciona gracias a los avances logrados en la codificación embebida automatizada por los desarrolladores de MathWorks, los diseños basados en modelos de bloques se le pueden cargar con facilidad al auto-piloto gracias al paquete de soporte de Embedded Coder (Embedded Coder Support Package for PX4), el cual se encuentra disponible desde el 2019 [21]. Gracias a este entorno de desarrollo es posible acceder a los periféricos del auto-piloto desde el entorno de MATLAB/Simulink y allí mismo generar código C ++ y cargárselo directamente a la tarjeta del controlador de vuelo [5].

El paquete de soporte “UAV Toolbox Support Package for PX4 Autopilots"proporciona interfaces para algunos de los componentes de la arquitectura PX4 de la Figura 20, mediante el uso de bloques Simulink. Se pueden utilizar estos bloques como entrada y salida para los algoritmos en el modelo de Simulink, en la Figura 21 se presenta una tabla de los módulos compatibles con los de la arquitectura en general del auto-piloto [19].

Component in PX4 Architecture	Simulink Block in the Support Package
Parameters	Read Parameter
uORB Message Bus	PX4 uORB Read PX4 uORB Write PX4 uORB Message
RC Input	Radio Control Transmitter
Sensors Hub	Vehicle Attitude Accelerometer Magnetometer Gyroscope
GPS	GPS
Position & Attitude Estimator	Vehicle Attitude
Output Driver	PX4 PWM Output
External serial communication	Serial Receive Serial Transmit

Figura 21: Módulos disponibles para utilizar del Toolbox de Auto-Pilotos PX4.

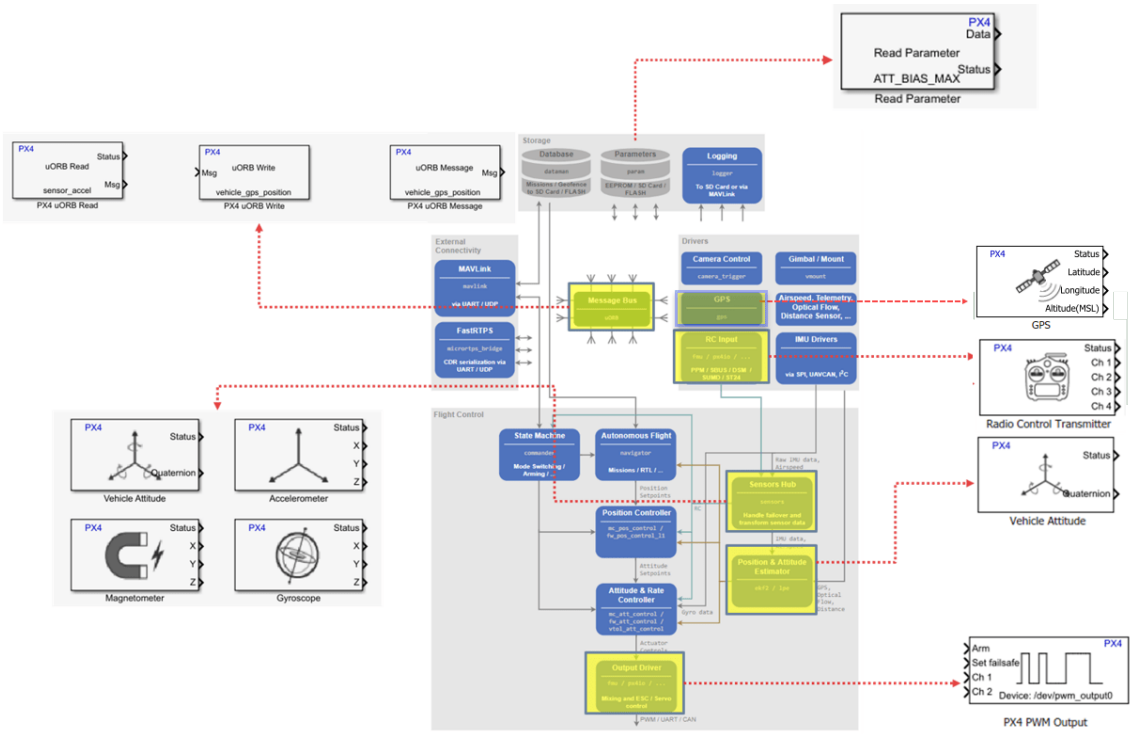


Figura 22: Mapa de los componentes disponibles del Toolbox para Auto-Pilotos de PX4.

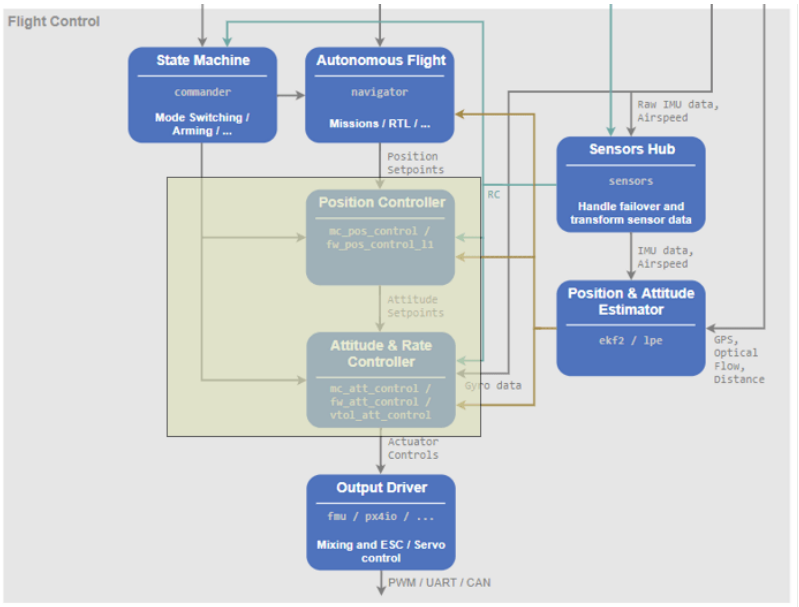


Figura 23: Componentes reemplazables por el usuario del Toolbox para Auto-Pilotos de PX4.

En la Figura 23 se muestran los componentes que se pueden reemplazar (los módulos de controlador de posición y controlador de actitud) en la arquitectura general PX4 con algoritmos definidos por el usuario que se desarrollen utilizando el paquete de soporte [19].

Para tener una visión mas clara sobre el cableado en general de este auto-piloto, con todos los componentes que conforman el dron, a continuación en la Figura 24 se presenta una infografía detallada con una breve descripción de varios componentes que se podrían llegar a utilizar.

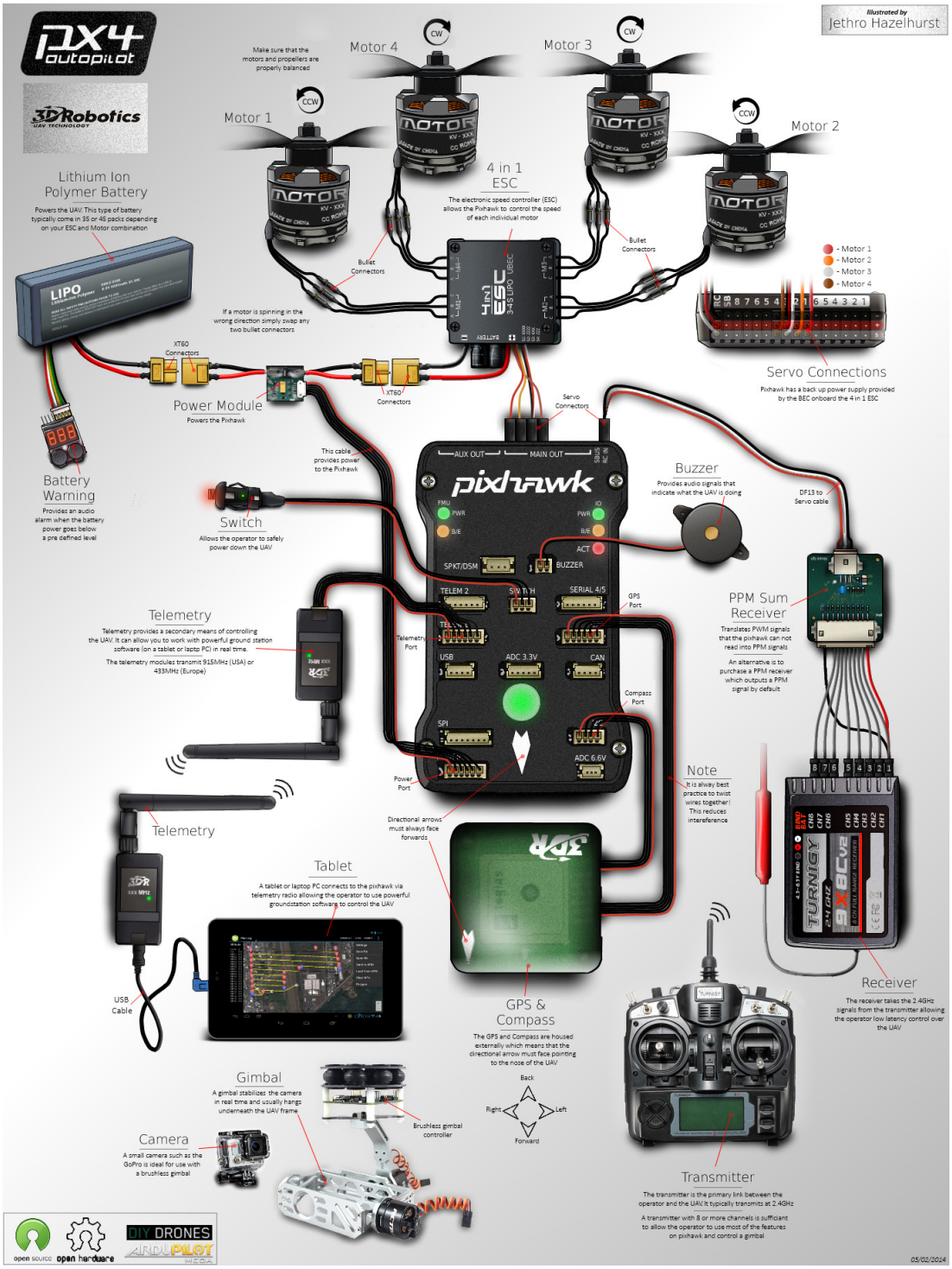


Figura 24: Infografía detallada de cableado del Pixhawk [22].

6.5. Codificador rotacional

Un codificador (encoder en inglés) no es más que un sensor que genera pulsos de señales periódicas en respuesta al movimiento. Existen dos tipos: incremental y absoluto. A su vez, cada uno de estos dos tipos pueden ser rotativos o lineales. En el caso de este proyecto se profundizará con el codificador rotativo de tipo incremental, ya que este es el que se implementó en la plataforma física del dron. La utilización de estos sensores brinda, a los sistemas de control de movimiento, información acerca de la posición, velocidad y dirección de los sistemas analizados.

Para el caso de los codificadores absolutos, la señal de salida posee información acerca de la velocidad angular (si son codificadores rotativos) o lineal, la dirección, y la posición, además de mantener la información aun cuando se le quita la fuente de energía. La característica de estos codificadores de movimiento es que son mucho más precisos y reducen la cantidad de código para interpretar su información, su desventaja radica en un rango de precio más elevado que el incremental.

En cuanto a los codificadores incrementales, la información proporcionada es únicamente la dirección de giro y el desplazamiento recorrido, ya sea lineal o angular. Estos codificadores son muy populares ya que además de su accesible precio, la implementación es más sencilla que el absoluto. El codificador rotacional incremental que se utilizará en esta plataforma en específico es el **LPD3806-600BM-G5-24C**, tal como se observa en la Figura 28.

6.5.1. Codificador rotativo mecánico (incremental)

Para entender el funcionamiento de un codificador incremental se utilizará como ejemplo uno rotativo mecánico de cuadratura simple de dos canales AB como el de la Figura 26. En cuanto a la obtención del desplazamiento angular, representado en grados, se puede utilizar la siguiente fórmula para obtener cada ángulo respectivo a la posición actual del encoder, para esto se debe conocer de cuantos canales es y la resolución de la señal de salida en pulsos/revolución (PPR).

$$Grados = (PosicionActualEncoder)\frac{360}{PPR}. \tag{46}$$

En cuanto a cómo saber si la rotación es en sentido horario (CW - Clock Wise) o antihorario (CCW - Counter Clock Wise), debido a que las ranuras de codificación de estos sensores están desfasadas 90° un canal con respecto del otro, esto hará que siempre se active el canal A antes que el B si el giro es en sentido horario (CW) y si el sentido es anti-horario el canal que se activará primero será el canal B (ver Figura 25). Por lo tanto lo que se realiza para saber la dirección de giro es una comparación de un estado actual con respecto al previo.

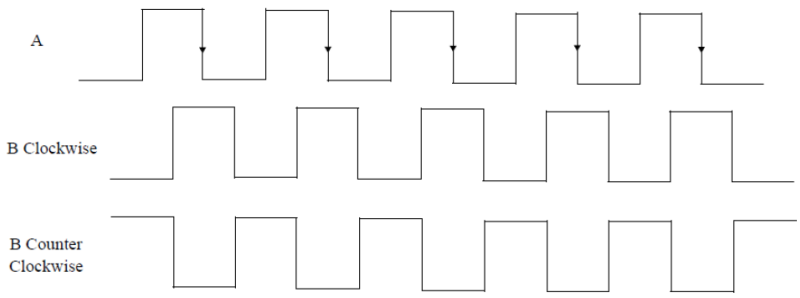


Figura 25: Salidas de onda cuadrada canales (A y B) de un codificador rotatorio incremental [23].

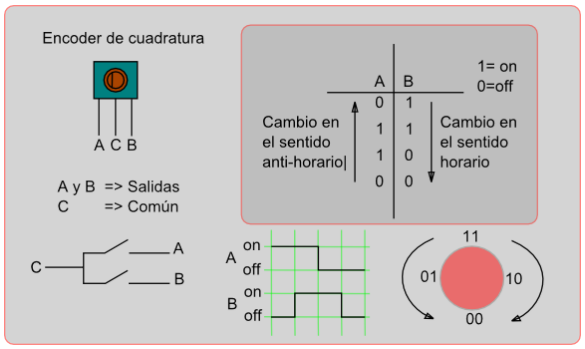


Figura 26: Estructura básica del encoder de cuadratura [24].

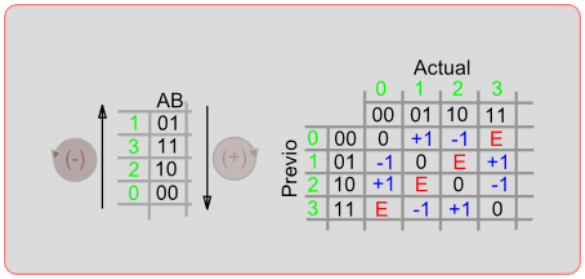


Figura 27: Matriz de incrementos - codificador ejemplo [24].

Como se ve, si se asume la salida del canal A como el bit menos significativo y la B como el bit más significativo, en la tabla de verdad se puede saber la dirección de giro si se compara la posición actual con la anterior.

Entonces, viendo la secuencia del encoder en la tabla de la Figura 26 y de la Figura 27, la cual corresponde a la matriz de incrementos, se puede comparar el estado previo con el actual y saber cómo fue el desplazamiento. Por ejemplo, si el estado actual del encoder es AB = 11 (3 en binario) y el estado anterior fue 01 (1 binario) se sabe que el encoder estaba girando en sentido horario (CW) ya en en la tabla de estado previo vs actual, se obtiene +1 como resultado de la intersección. Pero ahora, si el estado anterior es la secuencia 10 (2 binario) en vez de la otra, entonces el el encoder estaba girando en sentido anti-horario (CCW) ya que la tabla indica -1. El 1 indica la resolución de dicho encoder [24].

6.5.2. Codificador LPD3806-600BM-G5-24C

Figura 28: Codificador rotacional incremental de 2 canales [25].

Especificaciones técnicas [23]:

- Codificador rotatorio incremental, AB (dos fases).
- 600 pulsos por revolución (por canal o fase).
- Fuente de alimentación DC 5-24V.
- Velocidad mecánica máxima de 6000 rev / min.
- Frecuencia de respuesta: 0-20 KHz.

Wire Colour	Representation
Red wire	DC Power input, (5-24V DC)
Black wire	DC Power input, Ground (0V)
Green wire	A-phase output
White wire	B-phase output

Figura 29: Conexiones de cableado, encoder rotatorio incremental [23].

En codificadores de un canal es posible duplicar la resolución si se utiliza tanto el flanco de subida como el de bajada. Ya que este codificador tiene 2 fases AB, la resolución de pulsos por revolución (PRM) aumenta 4 veces ($600 * 2^2 = 2400 pulsos/rev$) lo cual es algo de beneficio, ya que se podrá determinar de una manera mas precisa la posición angular del dron abordo para verificar los métodos de control mas adelante. Dichas especificaciones se pueden encontrar en [23].

$$Resolucin = \frac{360}{2^2 \times 600} = \frac{360}{2400} = 0.15. \tag{47}$$

La resolución de este encoder es de 0.15° por paso en cada ranura, dicho de otra forma, por cada revolución completa se llegan a detectar 2,400 posiciones individuales.

6.5.3. Velocidad medida empleando el encoder

En el caso de la medición de la velocidad, el método más sencillo según [25] es midiendo directamente la frecuencia de pulsos del encoder. Esto se logra midiendo dicha cantidad de pulsos del encoder en una ventana de tiempo lo suficientemente pequeña para que la estimación de la velocidad sea la adecuada. Dicha velocidad angular estimada será una señal tipo discreta.

$$\omega[RPMs] \cong \frac{\Delta pulsos}{T_{sc}[ms]} \times \frac{1000[ms]}{1[seg]} \times \frac{1[rev]}{PPR_{[ENCODER]}} \times \frac{60[seg]}{1[min]}, \tag{48}$$

donde:

- Δ pulsos: Cantidad de pulsos medidos en el intervalo de tiempo definido.
- T_{sc}: tiempo de muestreo de la lectura de los pulsos.
- $PPR_{[ENCODER]}$: Pulsos por revolución del encoder (600 PPR en este caso ya que se utiliza sólo 1 canal).

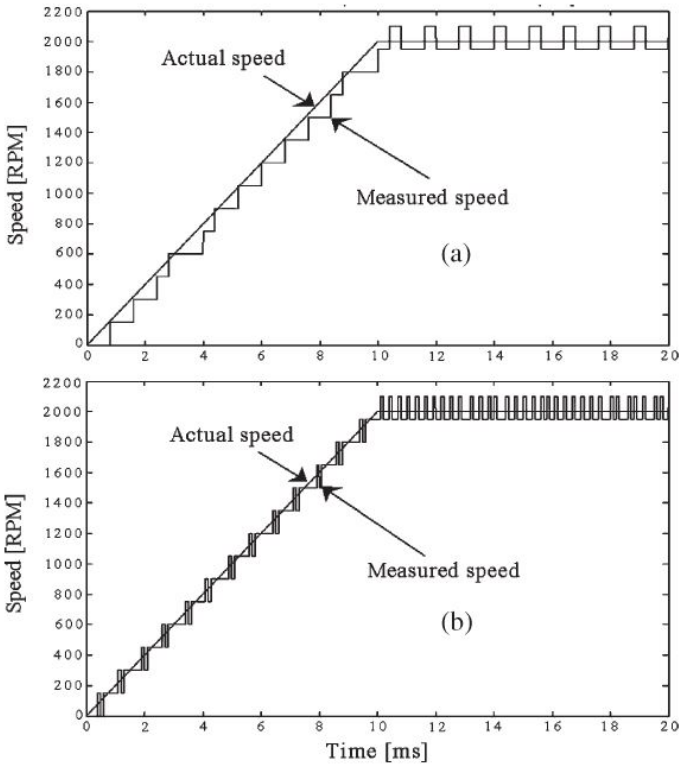


Figura 30: Medición RPMs: (a) Δpulsos = 1000, $T_{sc} = 400\mu s$; (*b*)Δpulsos = 4000, $T_{sc} = 100\mu s$[25].

Como vemos en la Figura 30, mientras menor sea el periodo de muestreo T_{sc} (el cual será la ventana de tiempo para leer los pulsos del encoder), mejor será la aproximación de la velocidad angular del eje que se esté analizando.

6.6. Motor sin escobillas (*brushless*)

Los motores sin escobillas o más conocidos como *brushless (BLDC)* en inglés, son motores eléctricos sincrónicos de tres fases que se componen por dos partes principales, el estator y el rotor. El estator es la parte inmóvil del mecanismo y el rotor es la parte giratoria del motor.

El estator está compuesto por una serie de ranuras en donde se enrollan grandes cantidades de cobre, a las que se les conoce como bobinas, cada uno de estos embobinados sirve como polos magnéticos para hacer funcionar al motor. El rotor está formado por una serie de imanes permanentes, los cuales son la característica principal del nombre que se le dio a este tipo de motor ya que dichos imanes hacen que este motor pueda girar sin contacto y desgaste físico/mecánico. Al tener un campo magnético constante creado por los imanes, no hay necesidad de un bobinado con escobillas para conducir la corriente al rotor [26].

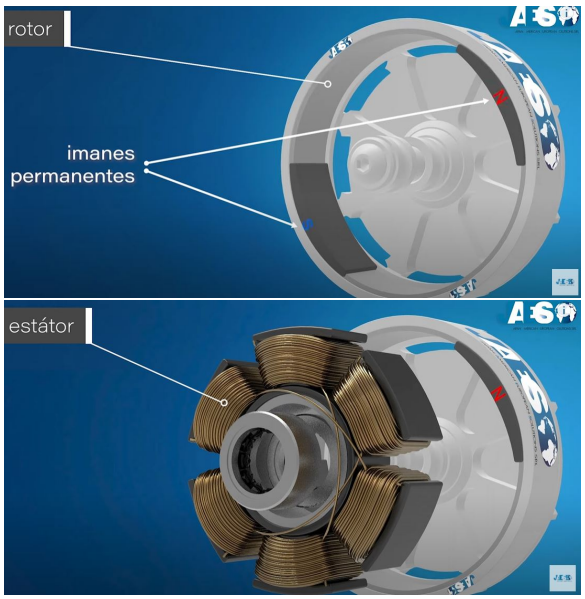


Figura 31: Partes de un motor sin escobillas - animado [26].

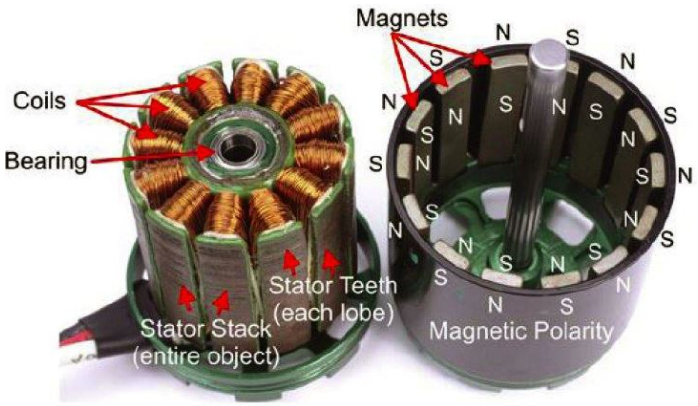


Figura 32: Partes de un motor sin escobillas - físico [26]

Algunas de las ventajas de estos motores sin escobillas (BLDC) con respecto a los motores con escobillas son las siguientes [23]:

- Mayor eficiencia (velocidad y par).
- Silencioso y fiable.
- Menor mantenimiento.
- Precios bajos.

El funcionamiento de estos motores es similar a otros motores eléctricos de imanes permanentes. Al energizar una bobina en el estator, esta crea un campo magnético variable. El rotor al poseer un campo magnético constante, detecta la variación entre ambos y tiende a alinear el campo creado por el estator y el propio haciendo girar el rotor ya que es la parte móvil del motor, el movimiento se da gracias a esa variación de polos entre ambos campos magnéticos es por ello que se dice que el rotor sigue siempre al estator.

Para lograr que el rotor siga girando, antes de que se alinee por completo la bobina energizada con el rotor, se energiza la bobina que le sigue y se apaga o se deja de alimentar a la anterior. Esto provoca que el campo magnético del rotor siga al campo magnético del estator, que va variando en el tiempo, haciendo que el rotor gire.

Una de las pocas desventajas que posee este tipo de motores, es que necesitan de un control electrónico para su correcto funcionamiento. Es bastante complicado controlar cuándo encender y apagar las bobinas del estator considerando que se quiere girar al rotor de manera uniforme, es por ello que se utiliza un ESC (Controlador Electrónico de Velocidad, por sus siglas en inglés) para poder variar las velocidades de giro por medio de PWM [26]. Dicho sistema de control utiliza un sensor tipo hall para cerrar el lazo y así poder saber cuando encender y pagar los transistores que controlan los campos eléctricos en las bobinas del estator.

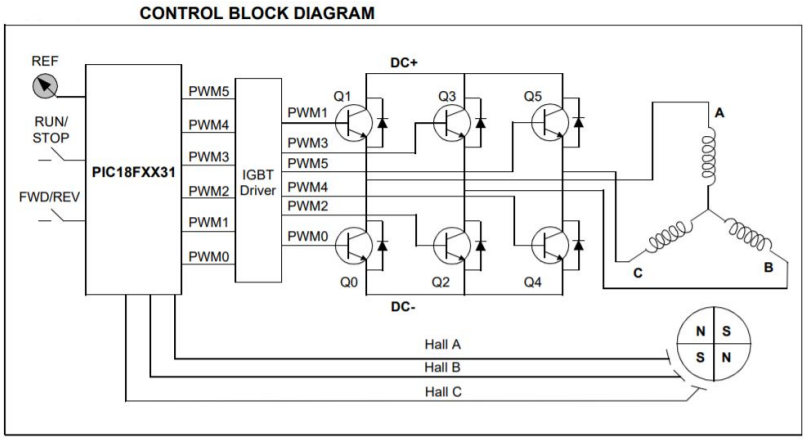


Figura 33: ESC - diagrama de bloques de control [27].

Usualmente las señales trifásicas en los motores son de forma sinusoidales, sin embargo, los motores DC sin escobillas pueden funcionar tanto con ese tipo de señal como también

con una conmutada. Si se utiliza una señal conmutada (cuadrada) entonces se trata de un motor DC sin escobillas (BLDC). La idea del funcionamiento de estos motores se ilustra en la Figura 36.

Sequence #	Hall Sensor Input			Active PWMs		Phase Current		
	A	B	C			A	B	C
1	0	0	1	PWM1(Q1)	PWM4(Q4)	DC+	Off	DC-
2	0	0	0	PWM1(Q1)	PWM2(Q2)	DC+	DC-	Off
3	1	0	0	PWM5(Q5)	PWM2(Q2)	Off	DC-	DC+
4	1	1	0	PWM5(Q5)	PWM0(Q0)	DC-	Off	DC+
5	1	1	1	PWM3(Q3)	PWM0(Q0)	DC-	DC+	Off
6	0	1	1	PWM3(Q3)	PWM4(Q4)	Off	DC+	DC-

Figura 34: Secuencia del control de rotación en sentido CW [27].

Sequence #	Hall Sensor Input			Active PWMs		Phase Current		
	A	B	C			A	B	C
1	0	1	1	PWM5(Q5)	PWM2(Q2)	Off	DC-	DC+
2	1	1	1	PWM1(Q1)	PWM2(Q2)	DC+	DC-	Off
3	1	1	0	PWM1(Q1)	PWM4(Q4)	DC+	Off	DC-
4	1	0	0	PWM3(Q3)	PWM4(Q4)	Off	DC+	DC-
5	0	0	0	PWM3(Q3)	PWM0(Q0)	DC-	DC+	Off
6	0	0	1	PWM5(Q5)	PWM0(Q0)	DC-	Off	DC+

Figura 35: Secuencia del control de rotación en sentido CCW [27].

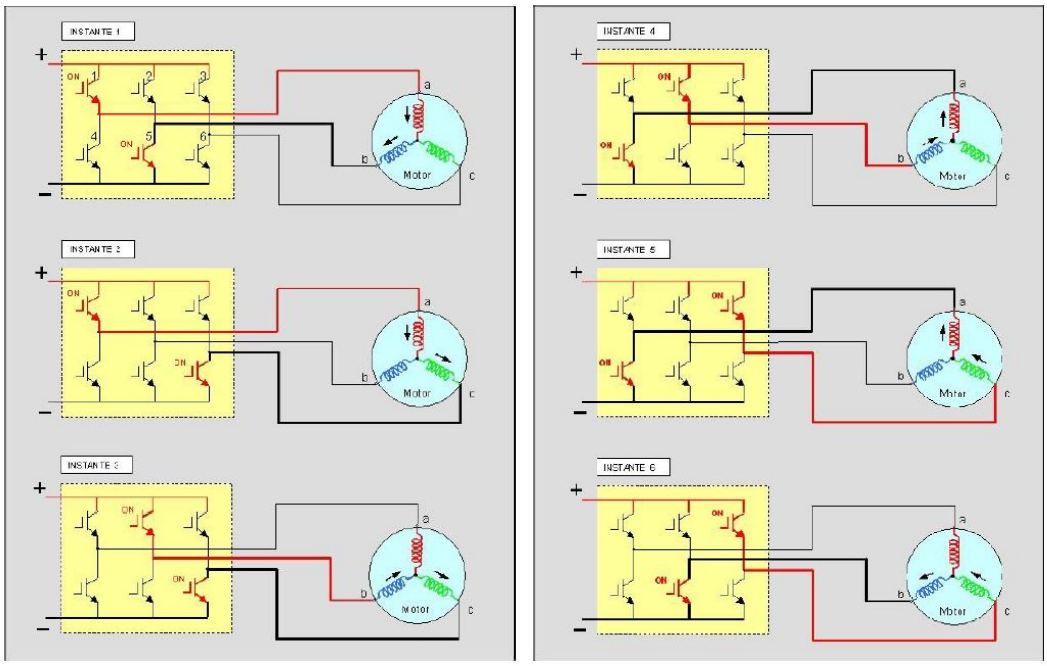


Figura 36: ESC - Diagrama de bloques de control [26].

6.6.1. Clasificaciones *Kv* para motores brushless

Cuando el motor se encuentra en movimiento, los imanes permanentes del rotor generan una corriente en los bobinados que no se encuentran energizados debido a la variación de flujo magnético, según la Ley de Faraday-Lenz. Esa corriente en el bobinado generada crea una tensión denominada fuerza contra electromotriz inducida, en inglés Back Electromotive Force (back EMF), que es directamente proporcional a la velocidad del motor mediante un coeficiente Kv [26].

Por medio de la ecuación (49), se puede relacionar en régimen permanente la velocidad del motor con la fuerza contra electromotriz.

$$Velocidad(RPM) = Kv \times Tension(Volts). \tag{49}$$

El coeficiente Kv es característico de cada motor (no debe confundirse con kV, la abreviatura de kilovoltio) sin escobillas y en este tipo de motores se proporciona con las unidades de revoluciones por minuto por voltio (RPM/Voltio) sin carga en ese motor, de tal forma que conociendo la tensión a la que se va a alimentar el motor, se conocerá la velocidad del mismo indirectamente [28].

Entender la clasificación ***Kv*** para este tipo de motor es sumamente importante ya que con un motor de bajos ***Kv*** (motor mas grande físicamente) puede producir un par de torsión más alto utilizando una hélice más grande, pero se caracteriza por trabajar a bajas revoluciones. En cambio un motor con altos ***Kv*** (motor mas pequeño físicamente) puede llegar a producir una mayor velocidad (RPMs), pero un torque no tan alto como el otro caso.

Un motor de altos Kv con una hélice pequeña (de 8 pulgadas por ejemplo $\rightarrow$ 8045) puede llegar a generar una fuerza de empuje mayor a un motor de bajos Kv con el mismo tipo y tamaño de hélice pequeña. Es por esto que es importante conocer esta clasificación, porque depende mucho el caso que se esté analizando ya que si en caso contrario tenemos un motor con de alto Kv con hélice mas grande (13 pulgadas por ejemplo) esa hélice no se puede utilizar en un motor de altos Kv ya que no aguantará la carga o torque que produce dicha hélice al empezar a girar. Es por esto que los fabricantes de cada motor recomiendan siempre qué tipo de alimentación con su tamaño respectivo de hélice.

Cabe mencionar que en la literatura para aplicaciones de drones utilizan la designación "S" refiriéndose con esa letra al número de celdas LiPo con cuál se alimentarán los motores sin escobillas. Considerando que cada celda de batería LiPo tiene un voltaje nominal de 3.7v.

CAPÍTULO 7

Estructura y hardware implementado

En este capítulo se presenta el procedimiento de las fases e interacciones realizadas en la implementación del diseño y manufactura de todo el hardware relacionado a este trabajo de graduación, con el objetivo de poder cumplir con los requerimientos y limitaciones de la plataforma a utilizar.

Cabe mencionar que dada la naturaleza de este trabajo para este capítulo en particular se decidió utilizar el procedimiento de diseño de Shigley [29], ya que a lo largo del mismo se realizaron varias iteraciones y mejoras con los elementos diseñados y manufacturados. Dicho esquema de diseño se ilustra en la Figura 37.

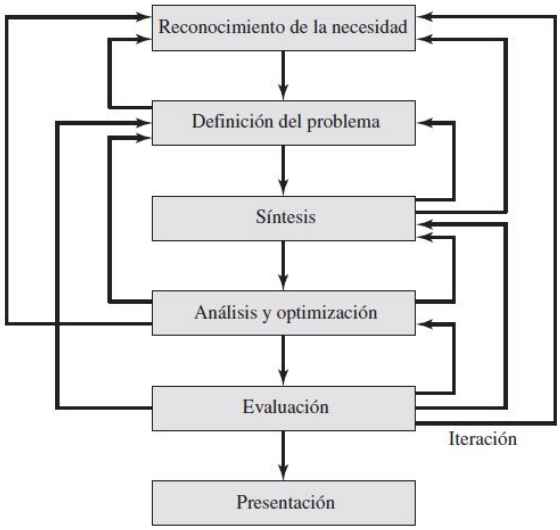


Figura 37: Fases del proceso de diseño de Shigley [29].

Los elementos más importantes a considerar dentro hardware implementado, diseñado y manufacturado son los que se presentan a continuación:

- Selección en los componentes básicos que conforman al multirrotor (motores, hélices, ESC).
- Diseño electrónico del multirrotor (selección de la fuente de poder y fabricación del PCB).
- Diseño mecánico del marco del multirrotor.

En las siguientes secciones, se describirá brevemente de la manera más detallada posible las necesidades y limitaciones de cada elemento manufacturado y diseñado, tomando siempre en consideración las fases de diseño de la Figura 37. Se describirá generalmente las necesidades en particular de cada uno de los tres elementos descritos anteriormente, así como también se mencionarán los problemas y retos que se encontraron a lo largo de esta fabricación e implementación física y como estos se llegaron a resolver mediante nuevas iteraciones.

7.1. Selección de componentes

En esta sección se justificó la selección de dos componentes fundamentales que conformaron al marco del dron. Estos componentes son los motores sin escobillas que conforman al multirrotor junto con sus controladores de velocidad (ESC) y el tamaño de las hélices mas adecuado a utilizar acorde a ese motor seleccionado.

7.1.1. Motores sin escobillas

7.1.2. Reconociendo la necesidad

La necesidad de escoger otro tipo de motor mas robusto sin escobillas surge a raíz de los obstáculos encontrados en la primer fase de este trabajo de graduación, ya que inicialmente la estructura de esta plataforma tipo gimbal fue pensada y diseñada para utilizarse con un dron Parrot AR 2.0, pero el primer problema que se tuvo es que este dron fue descontinuado del mercado desde hace ya mas de 8 años.

Figura 38: Dron Parrot AR 2.0 (descontinuado) [30].

7.1.3. Definición del problema

Hablando más a profundidad sobre los problemas que surgieron al utilizar el dron de la Figura 38 en la primer fase, se puede mencionar que los motores que tenía dicho dron eran obsoletos para lo que se quería lograr con la plataforma, esto se debe a que eran demasiado pequeños y no podían generar el torque y la velocidad necesaria para hacer girar al dron dentro de la estructura.

Figura 39: Motor del dron Parrot AR 2.0 (descontinuado) [30].

7.1.4. Síntesis y análisis del problema

Opción 1: Selección de motores y hélices

El primer motor sin escobillas que se consideró para el diseño del marco del dron, fue un motor Hitec con clasificación Kv de 390 [RPMs/Volt] sin carga. Los factores más importantes para seleccionar este motor como la primera opción al diseño corresponden a las siguientes ventajas:

- Disponibilidad de 6 motores en la UVG.
- La combinación de motor y ESC que ahorra espacio y peso de componentes por separado.
- Motor eficiente y de alto rendimiento.
- Selector de giro (CW y CCW - REV Switch).
- Firmware actualizable mediante puerto Flash.
- Control de velocidad programable.
- Respaldado por la garantía de un año de Hitec.

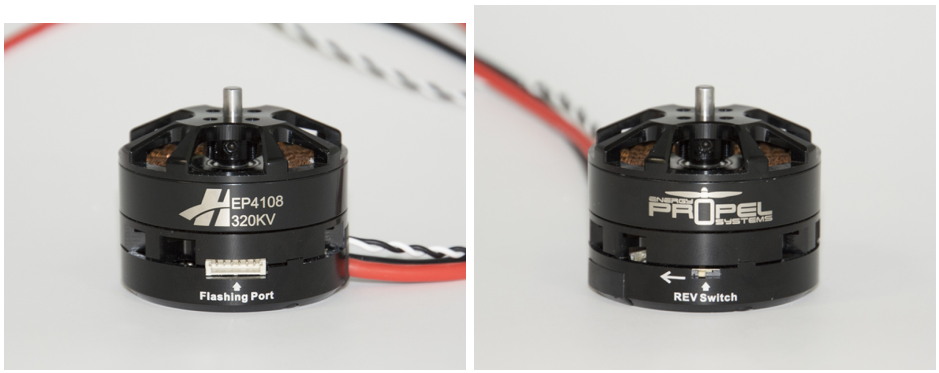


Figura 40: Motor Hitec 4108 de 390kv físico [31].

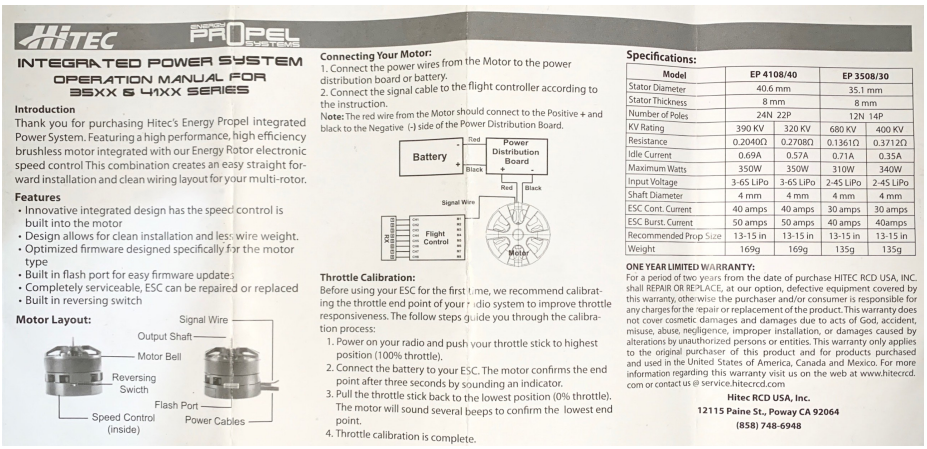


Figura 41: Ficha técnica motor Hitec 4108 de 390kv [31].

Selección hélice

La selección del tamaño de hélice es sumamente importante ya que si se escoge una hélice muy pequeña puede que no se aproveche al 100 % las capacidades físicas del motor y si se selecciona una muy larga, puede que esta no quepa debido a las limitaciones físicas de la plataforma, tal como se ve en la Figura 72 de la sección correspondiente al diseño del marco del multirrotor.

Figura 42: Hélices disponibles de 10, 9, 8 y 6 pulgadas respectivamente.

En la Figura 42 se muestra el tamaño de las hélices que se pudieron conseguir en el mercado de la región. Dichos tamaños son de:

- 1045: 10x4.5 pulgadas de largo y ancho.
- 9047: 9x4.7 pulgadas de largo y ancho.
- **8045: 8x4.5 pulgadas de largo y ancho**.
- 6045: 6x4.5 pulgadas de largo y ancho.

Al ver la hoja técnica de las especificaciones del motor de 390kv (Figura 41) se muestra que el fabricante para este motor en específico recomienda utilizar una hélice de 13-15 pulgadas (recommended prop size) de tamaño, considerando que el dron fuera a volar. Pero, para este caso esto no es necesario ya que solo se desea que funcione dentro de la estructura, es por ello que se puede seleccionar perfectamente una hélice más pequeña.

Las hélices de tamaño de 10 y 9 pulgadas se descartan ya que con la iteración final del marco del dron estas aún siguen topando con el eje del cabeceo y ya no es conveniente reducir el largo de los brazos del marco, por lo tanto las únicas dos opciones restantes fueron las de 8 y 6 pulgadas. La hélice de 6 pulgadas es demasiado pequeña en comparación con lo que recomienda el fabricante (era casi la mitad del mínimo que recomienda) entonces se descarta también ya que si se usa esta no se aprovechará al máximo las características del motor.

La hélice que se seleccionó finalmente fue la 8045 de 8 pulgadas de largo ya que es un valor intermedio entre los 4 largos disponibles con las cuales se contaban además de ser un tamaño de hélice que de acuerdo a las limitaciones de largos máximos de la plataforma se asegura que va a caber tal como podemos ver en la última iteración (ver Figura 81) del marco del dron.

Figura 43: Hélice final seleccionada (8045).

Obteniendo fuerza de empuje motor 390*Kv*

La fuerza de empuje dada la hélice de 8 pulgadas seleccionada fue otro factor importante a considerar. Dicha fuerza de empuje sirvió para tener una idea del consumo de potencia por cada motor para validar la fuente de poder. Además, también sirvió para tener una idea si puede o no haber otro motor como segunda opción que funcione mejor. Por otro lado, dicha fuerza de empuje obtenida fue utilizada para validar esfuerzos en nuestro diseño del marco del dron (posteriormente en la sección respectiva).

El siguiente experimento de la Figura 44, muestra como se estimó la fuerza de empuje de manera experimental.

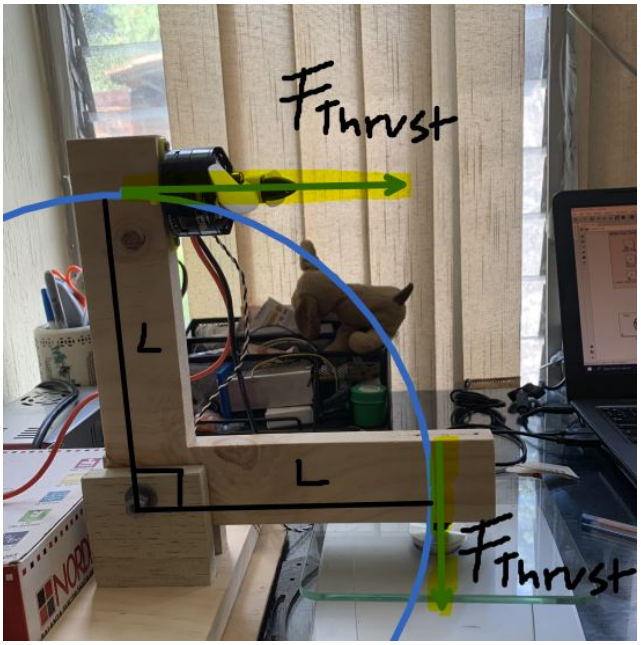


Figura 44: Diagrama de fuerza de empuje experimental (motor 390Kv).

La idea detrás de este experimento fue poder estimar de la manera mas económica y eficaz posible la fuerza de empuje de este motor. Para ello se armó la estructura de madera solida en forma de escuadra tal como se muestra en la Figura 44, en dicha figura se puede notar que los largos de ambos extremos de la escuadra de madera son de la misma longitud L (9 pulgadas) y la clave es poner el eje central de la hélice a una misma distancia que el apoyo de la pata de goma donde se colocará la pesa electrónica debajo tal como se muestra en la Figura 44.

Entonces analizando la Figura 44, si se aplica una fuerza tangencial al arco del círculo azul en el extremo superior de la escuadra de madera, esa misma fuerza será ejercida de manera perpendicular al plano de la balanza electrónica dado que la escuadra tienen la misma longitud L y esta gira bajo un mismo pivote. Por lo tanto, la fuerza que ejerza el motor con la hélice arriba (extremo a 90 grados de la horizontal) será la medición que tome la balanza electrónica en el otro lado (extremo a 0 grados de la horizontal) de la escuadra.

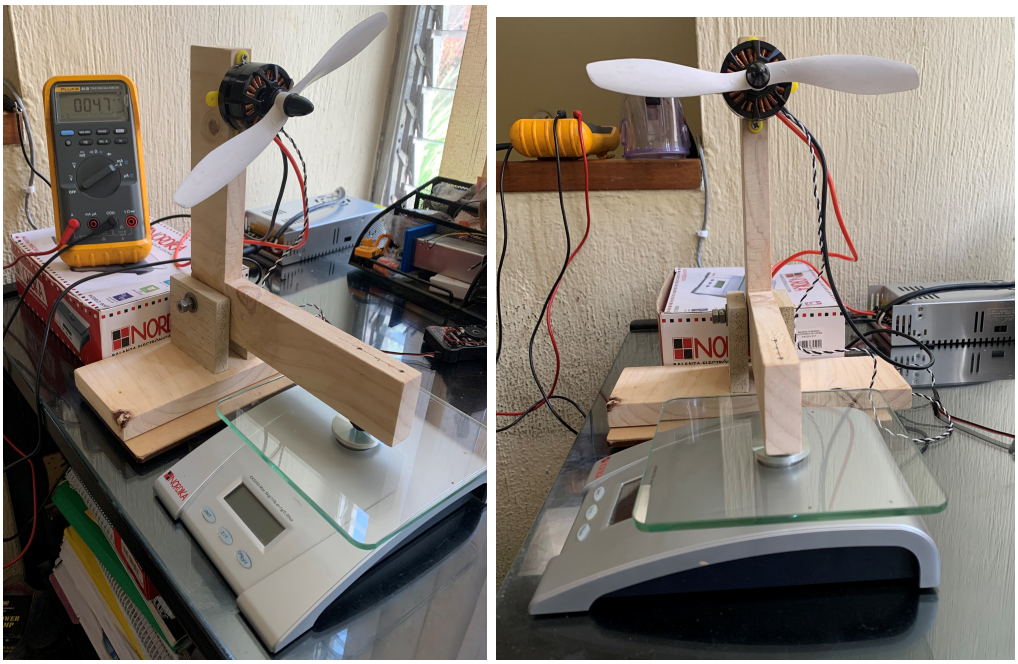


Figura 45: Vistas de la maqueta para el experimento.

Figura 46: Puntos clave de la maqueta.

Con la maqueta ensamblada y lista ya se pudo continuar a realizar las mediciones respectivas de la fuerza de empuje. Entonces, teniendo el controlador de vuelo y Matlab se fue variando el PWM para controlar la velocidad del motor de 1200 (mín.) a 2000 (máx.) en intervalos de 50 y así obtener la data medida.

Resultados opción 1: selección de motor y hélice

HiTec 4108- 390KV (Hélice 8045)			
Voltaje de Alimentación		12.25	Voltios
PWM	***Fuerza [gramos]***	***Corriente [mA]***	***Potencia [W]***
1000	95	47	0.576
1100	104	98	1.201
1200	127	150	1.838
1250	138	298	3.651
1300	150	372	4.557
1350	159	450	5.513
1400	168	516	6.321
1450	173	485	5.941
1500	184	646	7.914
1550	190	704	8.624
1600	195	764	9.359
1650	204	816	9.996
1700	210	857	10.498
1750	216	901	11.037
1800	220	924	11.319
1850	228	961	11.772
1900	240	1022	12.520
1950	256	1112	13.622
2000	261	1236	15.141

Cuadro 1: Resultados motor brushless Hitec 4108 de 390kKv.

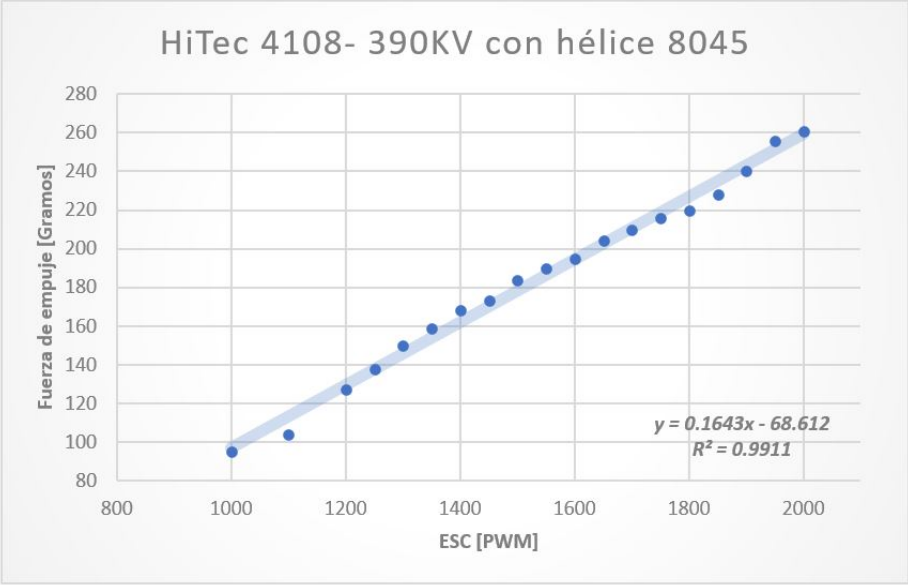


Figura 47: Motor 1 - gráfica fuerza de empuje vs PWM.

Opción 2: selección de motores y hélices

El segundo motor que se consideró como opción secundaria, fue un motor brushless de mayor coeficiente Kv(RPM/voltio sin carga). Este es el motor del fabricante Emax MT2213 de 935kV [32].

Los factores importantes o ventajas a tomar en consideración de este motor como segunda opción son los siguientes:

- Posee un menor costo con respecto a la opción 1 ($15.50).
- Entre las hélices recomendadas se encuentra la 8045.
- Mas liviano - 53gramos (posee un peso mucho menor al Hitec).
- Posee una tabla característica de la corriente, potencia, eficiencia y velocidad de acuerdo a sus 2 hélices recomendadas.
- Incluye un par de hélices de 1045 y 3 conectores bala.
- Se cuenta con 4 motores actualmente con sus respectivos ESC.

Los puntos en contra de este motor fueron los siguientes:

- Poca disponibilidad en el mercado nacional.
- No incluye el ESC incorporado.
- Alimentación recomendada para Hélice de 8045 de 4S (14.8v).

Figura 48: Combo del motor Emax de 935Kv [32].

Figura 49: Motor físico y conectores bala.

Multi copter motor MT2213-935KV(1045Combo)

DIAMETER.....27.9MM
LENGHT(MM)....39.7MM
WEIGHT.....53G
KV.....935
3S(LIPO).....1045(PROP)
ESC recommended.....18A
MAX THRUST.....860G

Motor for Multi Copter			
Model Type	Lipo	RPM/V	Propeller
2213	3S	935	1045
2213	4S	935	8045

Figura 50: Especificaciones técnicas del motor $935Kv$.

The voltage (V)	Paddle size	current (A)	thrust (G)	power (W)	efficiency (G/W)	speed (RPM)	Working temperature (° C)
11	EMAX8045	1	110	11	10.0	3650	
		2	200	22	9.1	4740	
		3	270	33	8.2	5540	
		4	330	44	7.5	6200	
		5	390	55	7.1	6700	
		6	440	66	6.7	7150	
		7.1	490	78.1	6.3	7400	36
	EMAX1045	1	130	11	11.8	2940	
		2	220	22	10.0	3860	
		3	290	33	8.8	4400	
		4	370	44	8.4	4940	
		5	430	55	7.8	5340	
		6	480	66	7.3	5720	
		7	540	77	7.0	5980	
		8	590	88	6.7	6170	
		9	640	99	6.5	6410	
		9.6	670	106	6.3	6530	43

Figura 51: Tabla característica del motor Emax $935Kv$ [32].

Este motor tenía la gran ventaja de poseer la tabla característica de la fuerza de empuje dadas las hélices recomendadas por el fabricante, esta se muestra en la Figura 51. Con esta tabla se podría comprobar si la maqueta realmente medía un valor aproximado al que el fabricante dice en dicha tabla y así comparar los resultados experimentales. Para este motor se realizó el mismo proceso que para el caso del motor anterior.

Figura 52: Emax con ESC físico.

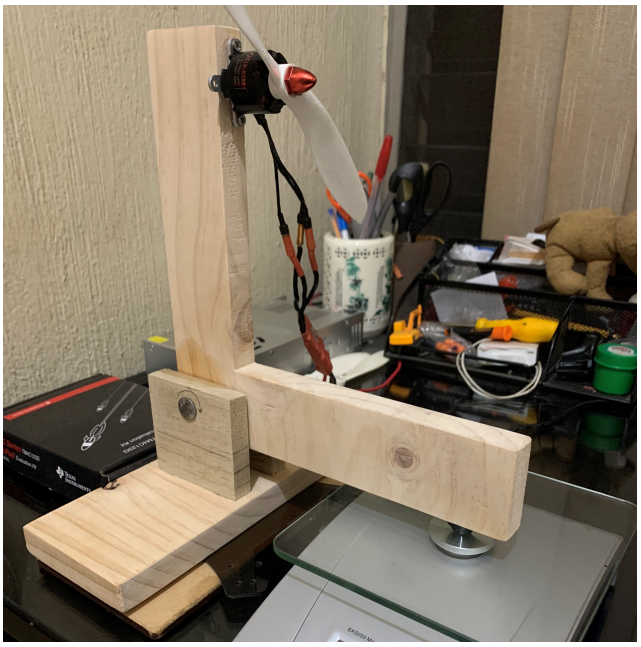


Figura 53: Maqueta de prueba con motor Emax $935Kv$.

Resultados opción 2: selección de motor y hélice.

Para esta segunda opción, dada las limitaciones de espacio, se utilizó el mismo tamaño de hélice de la opción uno, la de 8 pulgadas (8045).

EMAX MT2213 - 935KV (Hélice 8045)			
Voltaje de Alimentación		12.25	Voltios
PWM	***Fuerza [gramos]***	***Corriente [mA]***	***Potencia [W]***
1000	47	32	0.392
1200	69	192	2.352
1250	96	330	4.043
1300	117	502	6.150
1350	147	705	8.636
1400	168	923	11.307
1450	190	1158	14.186
1500	217	1402	17.175
1550	238	1648	20.188
1600	257	1978	24.231
1650	287	2388	29.253
1700	303	2853	34.949
1750	362	3878	47.506
1800	382	4020	49.245
1850	402	5010	61.373
1900	413	5150	63.088
2000	430	5190	63.578

Cuadro 2: Resultados motor brushless Emax MT2213 de 935Kv.

Figura 54: Motor 2 - gráfica fuerza de empuje vs PWM.

7.2. Diseño electrónico

En esta sección se presentan las mediciones de los parámetros de corriente máximos para considerar en el diseño de la alimentación del dron con los motores Hitec 4108 de 390kv con las hélices de 8 pulgadas escogidas.

Cabe mencionar que tanto la selección de la fuente de alimentación, como los requerimientos necesarios para el diseño del PCB (para alimentar a los 4 motores del multirrotor) estuvieron relacionados directamente.

7.2.1. Fuente de alimentación

Reconociendo la necesidad

Seleccionar y validar qué fuente de alimentación se puede utilizar para alimentar los motores sin escobillas escogidos (Hitec 390kv) con la carga de las hélices establecidas (hélices 8045).

Definición del problema

Para poder tener una idea de la fuente de poder que se iba a necesitar para alimentar los 4 motores del dron con las hélices escogidas, se realizaron varias pruebas variando la velocidad de dichos motores con el mínimo y máximo valor de PWM para así establecer los rangos máximos del consumo de corriente por cada motor con la carga de la hélice de 8 pulgadas.

Para tener una idea de qué voltaje utilizar como punto de partida se utilizó el mínimo recomendado por el fabricante de los motores. Como se muestra en la Figura 41, se observa que lo recomendado es de 3S-6S y como se detalló en el apartado del marco teórico respectivo, por cada celda ”S” corresponde un voltaje de 3.7 voltios. Por lo tanto se tiene un rango de alimentación de 11.10 - 22.2 voltios para alimentar los 4 motores.

Síntesis y análisis del problema

Iteración 1: Selección fuente de poder

Para esta primer iteración se utilizó una de las nuevas fuentes de alimentación del laboratorio del CIT para ver si esta podía llegar a entregar la corriente y potencia necesaria para alimentar a los 4 motores con el tamaño de hélices seleccionadas.

El modelo y marca de esta fuente es **KEITHLEY modelo 2231A-30-3**. Su forma física y sus especificaciones se muestran en la Figura 55.

Specifications

DC OUTPUT RATING

	Channel 1	Channel 2	Channel 3
Voltage	0–30 V	0–30 V	0–5 V
Current	0–3 A	0–3 A	0–3 A

MAXIMUM POWER: 195W

LOAD REGULATION:
Voltage: ≤0.02% + 4mV
Current: ≤0.2% + 3mA

LINE REGULATION:
Voltage: ≤0.02% + 4mV
Current: ≤0.2% + 3mA

RIPPLE AND NOISE (20Hz–20MHz):
Voltage: ≤1mVrms/≤5mVp-p
Current: ≤6mArms

SETTING RESOLUTION:
Voltage: 10mV
Current: 1mA

SETTING ACCURACY:
Voltage: ≤0.06% + 20mV
Current: ≤0.2% + 10mA

READBACK RESOLUTION:
Voltage: 10mV
Current: 1mA

READBACK ACCURACY:
Voltage: ≤0.06% + 20mV
Current: ≤0.2% + 10mA

ISOLATION VOLTAGE, OUTPUT TO CHASSIS: Any output can be isolated up to 240V (DC + peak AC with AC limited to a maximum of 3Vpk-pk and a maximum of 60Hz) relative to the earth ground terminal.

ISOLATION VOLTAGE, OUTPUT TO OUTPUT: Any output can be isolated up to 240V (DC + peak AC with AC limited to a maximum of 3Vpk-pk and a maximum of 60Hz) relative to any other output terminal.

TRACKING AND COMBINATION MODES:
Tracking Mode: Maintains the ratio on the two 30V output channels that is present when the control is activated.
Combination $V_1 + V_2$ Series Mode: Deliver up to 60V when CH1 and CH2 are wired in series. Meter reads back combined voltage.
Combination $I_1 + I_2$ Parallel Mode: Deliver up to 6A when CH1 and CH2 are wired in parallel. Meter reads back combined current.

Figura 55: Iteración inicial, fuente de alimentación para el dron.

Resultados físicos iteración 1: selección fuente de poder

Para los resultados de esta primera iteración se alimentó a los 4 motores con un voltaje nominal de 11.10 voltios (3S) y se varió el valor de 1 PWM (1 motor girando) de 1000 a 2000. Los resultados de las mediciones se detallan en el Cuadro 3.

Fuente UVG			
Voltaje de Alimentación	**Vcc (4 Motores Conectados 1 solo funcionando)**	**11.10**	**Voltios**
VALOR PWM	Medición: Display Fuente	Medición: Multimetro Fluke 85	
(1250-2000)	Corriente Consumida (Amperios)	Corriente Consumida (Amperios)	mA
1000 (CERO MOTOR SIN GIRAR)	0.167	0.168	168
1250	0.24	0.247	247
1300	0.31	0.322	322
1350	0.39	0.390	390
1400	0.47	0.474	474
1450	0.54	0.543	543
1500	0.61	0.610	610
1550	0.67	0.667	667
1600	0.73	0.725	725
1650	0.79	0.776	776
1700	0.83	0.818	818
1750	0.87	0.860	860
1800	0.92	0.893	893
1850	0.96	0.932	932
1900	1.03	1.020	1020
1950	1.14	1.100	1100
2000	1.2	1.1275	1275

Cuadro 3: Resultados corriente consumida (1 motor), fuente iteración 1.

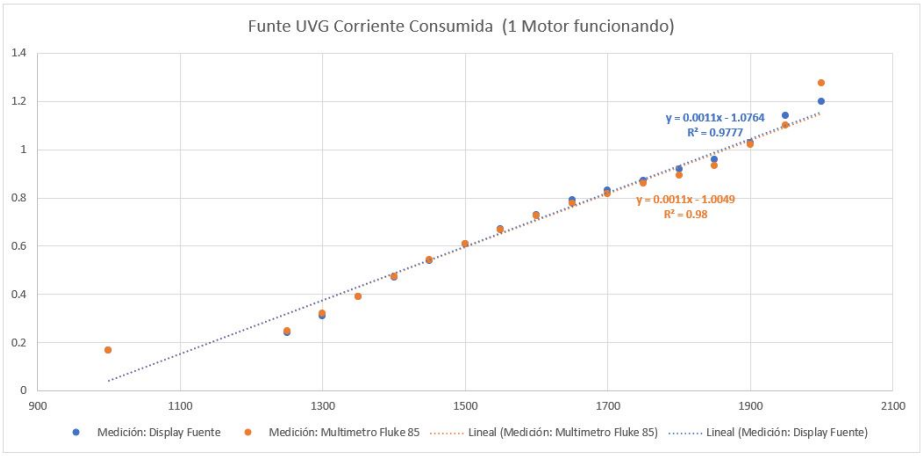


Figura 56: Gráfica consumo corriente, fuente iteración 1.

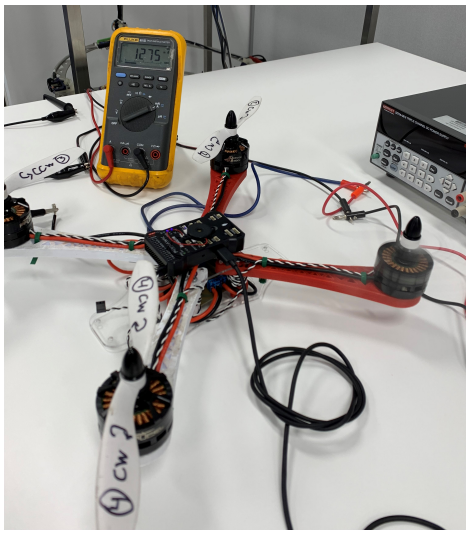


Figura 57: Consumo máximo (4 motores a Vcc), 1 motor funcionando al máximo.

Iteración 1: Análisis y optimización de los resultados

De acuerdo con los resultados de las mediciones anteriores de corriente, para un motor funcionando a su máxima velocidad y alimentando los cuatro a Vcc, se nota que esta fuente no logra satisfacer los requerimientos de corriente mínima para que los cuatro puedan funcionar al mismo tiempo ya que como se observa en la Figura 55, esta fuente puede suplir únicamente 3 amperios como máximo y, si se multiplica el consumo de un motor 1275A por cuatro, se obtiene un consumo mínimo de 5.10 amperios. Por lo tanto esta opción quedó totalmente descartada por lo que hubo que buscar otra fuente de alimentación que pudiese suplir al menos esa cantidad de corriente.

Iteración 2: selección fuente de poder

Los nuevos requerimientos para esta segunda iteración se basaron en seleccionar y validar una fuente de alimentación de por lo menos 11.10 voltios y que fuese capaz de entregar al menos 5.10 amperios de corriente.

El modelo y marca de esta segunda fuente fue una **ZYLTECH modelo S-360-12 (8PS120360DCU)**. Su forma física y especificaciones se muestran en las Figura 58.

		8PS120360DCU
Output	DC voltage	12V
	Voltage tolerance	±1%
	Rated current	30A
	Current range	0~30A
	Rated power	360W
	Ripple&noise	120mvp-p
	DCvoltage ADJ.range	±10%
Input	Voltage range	90~132VAC/170~264VAC(selected by switch),235~373VDC
	frequency	47~63HZ
	AC current (max.)	6.5A/115V 4A/230VAC
	Efficiency	83%
	Inrush current	40A
	Leakage current	<3mA/240VAC
Protection	Overload, Short-circuit	Rated output power 115%~135% start over load protection protection type: hiccup mode, auto-recovery after fault condition is removed
Environment	Working temp & humidity	-30°C~+60°C,20%~90%RH
	Storage temp & humidity	-60°C~+85°C,10%~95%RH non-condensing
	Withstand vibration	10~500HZ,2G 10min/1cycle,period for 60 minutes, each axes
Safety	Withstand voltage	I/P-O/P:1.5KVAC I/P-FG:1.5KVAC O/P-FG:0.5KVAC
	Isolated resistance	I/P-O/P,I/P-FG,O/P-FG:100M ohms/500VDC
Safety & EMC	Safety standard	EN61347-2-13
	EMC standard	refer to EN55022, CLASS B
Others	Dimension	215*115*50mm (L*W*H)
	Net Weight	970GR

Figura 58: Iteración 2, fuente de alimentación para el dron.

Fuente 30A						
Voltaje de Alimentación	**Vcc (4 Motores alimentadaos)**		**12.23**		**Voltios**	
	1 Motor funcionando (4 a VCC)		2 Motores funcionando (4 a VCC)		4 Motores funcionando (4 a VCC)	
PWM (1250-2000)	Corriente (mA) - Fluke 85	Corriente (A) - Fluke 85	Corriente (mA) - Fluke 85	Corriente (A) - Fluke 85	Corriente (mA) - Fluke 85	Corriente (A) - Fluke 85
1000	174	0.174	174	0.174	174	0.174
1250	261	0.261	343	0.343	509	0.509
1300	342	0.342	496	0.496	-	-
1350	422	0.422	643	0.643	-	-
1400	513	0.513	808	0.808	-	-
1450	588	0.588	950	0.950	-	-
1500	663	0.663	1090	1.090	1925	1.925
1550	734	0.734	1212	1.212	-	-
1600	801	0.801	1319	1.319	-	-
1650	859	0.859	1419	1.419	-	-
1700	911	0.911	1500	1.500	-	-
1750	950	0.950	1588	1.588	2782	2.782
1800	986	0.986	1673	1.673	-	-
1850	1032	1.032	1721	1.721	-	-
1900	1106	1.106	1871	1.871	-	-
1950	1221	1.221	2085	2.085	-	-
2000	1275	1.275	2152	2.152	3790	3.790

Cuadro 4: Resultados corriente consumida , fuente iteración 2.

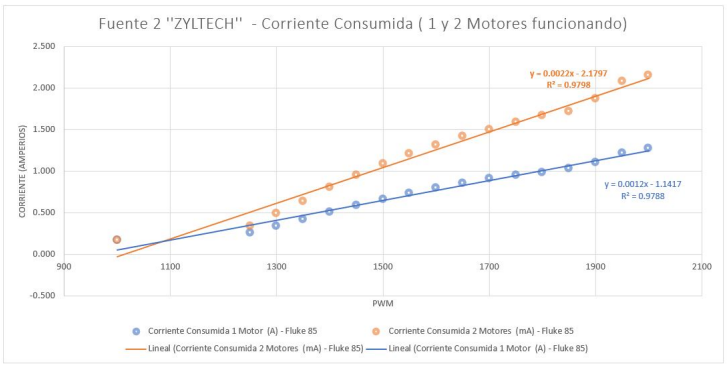


Figura 59: Fuente iteración 2, gráfica (a) consumo de corriente (1 y 2 motores en funcionamiento).

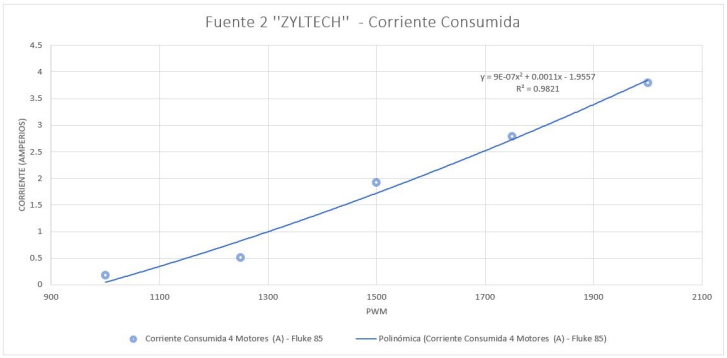


Figura 60: Fuente iteración 2, gráfica (b) consumo de corriente (4 motores en funcionamiento).

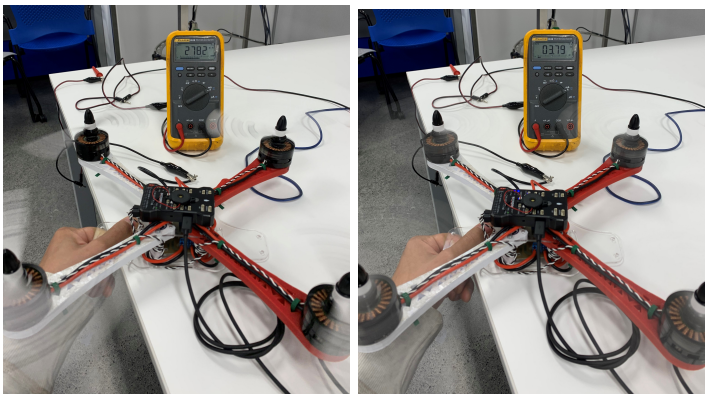


Figura 61: Corriente consumida en PWM de 1750 y 2000 respectivamente.

Iteración 2: Análisis y optimización de los resultados

Como se puede observar tanto en el Cuadro 4 como en la Figura 60, para esta segunda iteración se logró asegurar que la fuente de poder "ZYLTECH" fue capaz de entregar el mínimo esperado de **5.10 amperios** (teóricos). Sin embargo, al probar los cuatro motores en conjunto a su máximo valor de PWM (2000) el consumo experimental fue de **3.79 amperios**, en cualquiera de los dos casos se cumple lo mínimo requerido. Utilizando la ecuación (49) ese consumo máximo correspondería a una velocidad angular máxima de aproximadamente 4769.70 RPM. La fuente puede llegar entregar hasta 30 amperios por lo que se tiene un margen bastante amplio para exigirle más si se quisiera en futuras iteraciones.

7.2.2. Diseño y fabricación del PCB

Reconociendo la necesidad

Diseñar un circuito impreso que cumpla con los requerimientos de consumo de corriente (5.1 Amperios teóricos / 3.79 Amperios experimentales) para alimentar a los cuatro motores del multirrotor al mismo tiempo.

Definición del problema

Con base en los resultados de la corriente máxima que pueden llegar a consumir los cuatro motores del marco del dron con la hélice de 8 pulgadas, se calculó el ancho de pistas para el PCB. Se utilizará una corriente superior a la experimental (3.79A) ya que ese consumo se obtuvo con el marco del dron fuera de la plataforma. Para tener un rango amplio y seguro se tomará una corriente de consumo de 6 amperios para el cálculo del ancho de pistas y de espaciado entre cada una de estas. Además de cumplir con esos requerimientos la placa debe poder ser montada directamente en el marco del dron o ser parte del mismo.

Síntesis y análisis del problema

Para el cálculo de pistas y espaciado entre ellas se utilizó la calculadora **ANSI PCB Trace Width Calculator**. Dicha calculadora con los parámetros de entrada y salida para el diseño, se muestra en la Figura 62.

ANSI PCB TRACE WIDTH CALCULATOR

Input Data			Results Data				
				Internal Traces		External Traces	
Field	Value	Units	Trace Data	Value	Units	Value	Units
Current (max. 35A)	6	Amps	Required Trace Width	9.49	mm	3.65	mm
Temperature Rise (max. 100°C)	10	°C	Cross-section Area	501.89	mil²	0.12	mm²
Cu thickness	1	oz/ft²	Resistance	0	Ω Ohms	0	Ω Ohms
Ambient Temperature	27	°C	Voltage Drop	0.01	Volts	0.02	Volts
Conductor Length	1	inches	Loss	0.05	Watts	0.14	Watts
Peak Voltage	12.25	Volts	Required Track Clearance	0.65	mm		

Figura 62: Parámetros calculadora de pistas ANSI [33].

Iteración 1: Diseño PCB

La primera versión del PCB se fabricó directamente debido a que se tenía que realizar pruebas de la plataforma con el dron lo antes posible. Esta versión inicial de la placa se realizó con el programa Circuit Wizard [34], este fue mas accesible utilizar debido al tiempo y recursos en ese momento. Los resultados de dicha placa se muestran a continuación.

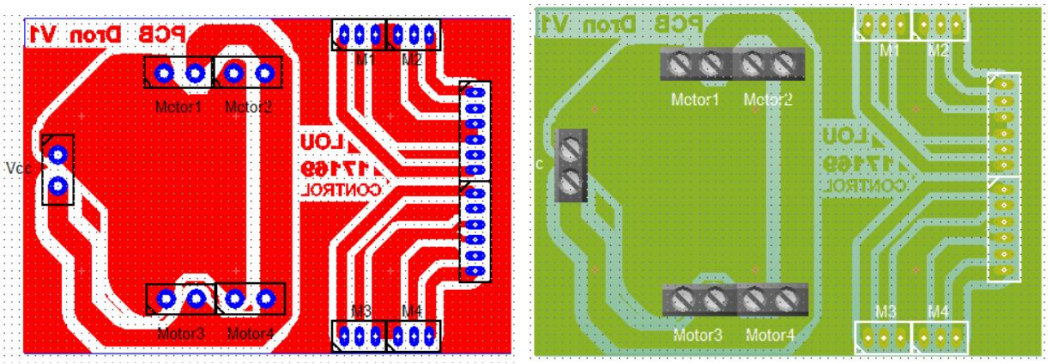


Figura 63: Placa versión 1, Circuit Wizard.

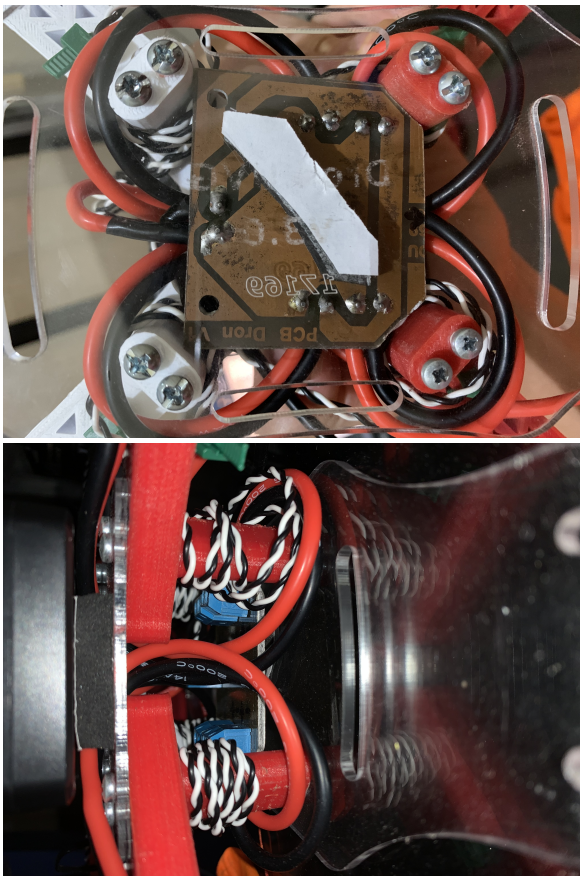


Figura 64: Placa versión 1 (resultados físicos).

Iteración 1: Análisis y optimización de los resultados

Debido a que esta primera versión de la placa se diseñó con las dimensiones de la segunda versión del marco del dron, el tamaño de la placa no cabía debido a que la versión final del marco con la que se trabajó fue el de la iteración tres (ver Figura 81) que corresponde al marco F366.

Por ende, a la placa se le tuvo que cortar algunos bordes para que cupiera. La recomendación para la siguiente iteración fue realizar la placa con el borde del CAD de Inventor para que esta forme parte del marco del dron con las medidas exactas, de esa forma se ahorraría espacio y peso al tener la placa y la base inferior del marco (acrílico) combinados en una sola pieza.

Iteración 2: Diseño PCB

Para esta segunda iteración se mejoró el diseño de la placa con el software de diseño para placas de Altium Designer. La ventaja de dicho programa es que puede importar archivos CAD que es justo lo que se necesita para realizar un PCB con la geometría de la base inferior del marco, que inicialmente fue manufacturada de acrílico [35].

Cabe mencionar que Altium únicamente permite importar archivos DXF/DWG creados exclusivamente con el software de AutoCAD. Ya que AutoCAD e Inventor son de Autodesk se pueden exportar e importar archivos CAD entre ambos programas. Los pasos para importar el CAD con la geometría del contorno de la placa que se necesita en Altium son los siguientes [35]:

1. Desde Inventor realizar un boceto final en la pieza con la geometría deseada, proyectar dicha geometría, exportar y guardar el archivo como archivo DXF/DWG.
2. Abrir AutoCAD e importar dicho archivo DXF/DWG creado en Inventor.
3. Verificar que las dimensiones de AutoCAD e Inventor sean las mismas.
4. Si todo esta en orden, guardar el archivo como "AutoCAD 2013/LT2013 DXF" (esta versión funcionó, no todas las versiones sirven).
5. Luego en Altium importar dicho DXF/DWG en la pestaña de File»Import»DXF/DWG.
6. En la ventana despeglada seleccionar en Blocks»Import as primitives, en Drawing space»Model y en scale seleccionar las unidades con las cuales se diseño el CAD, que en este caso fue en milímetros.
7. Seleccionar la geometría importada y seleccionar Desing»Board Shape»Define from selected objects.

Dicho proceso de importación se muestran en las Figuras 13.2 - 13.2 del Anexo correspondiente, en esa misma sección también puede verse en la Figura 13.2 la asignación de las reglas de diseño del ancho del las pistas y su espaciado mínimo entre ellas asignando los valores obtenidos con la calculadora ANSI, dichos parámetros de diseño se observan en la Figura 62.

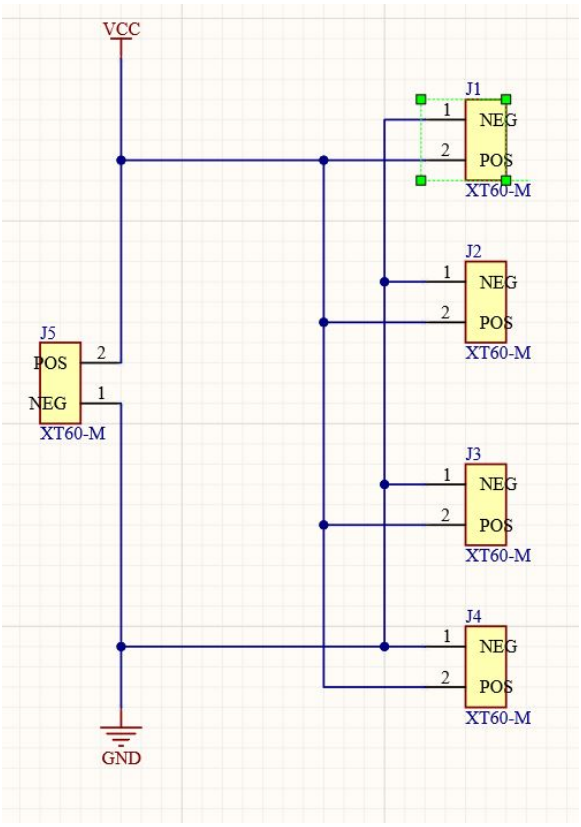


Figura 65: Esquemático PCB versión 2, Altium Designer

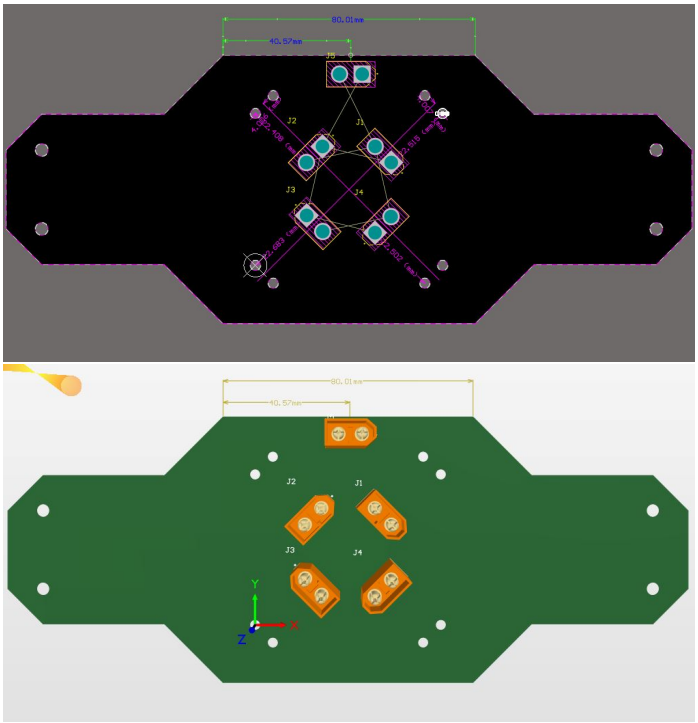


Figura 66: Ubicación conectores XT60.

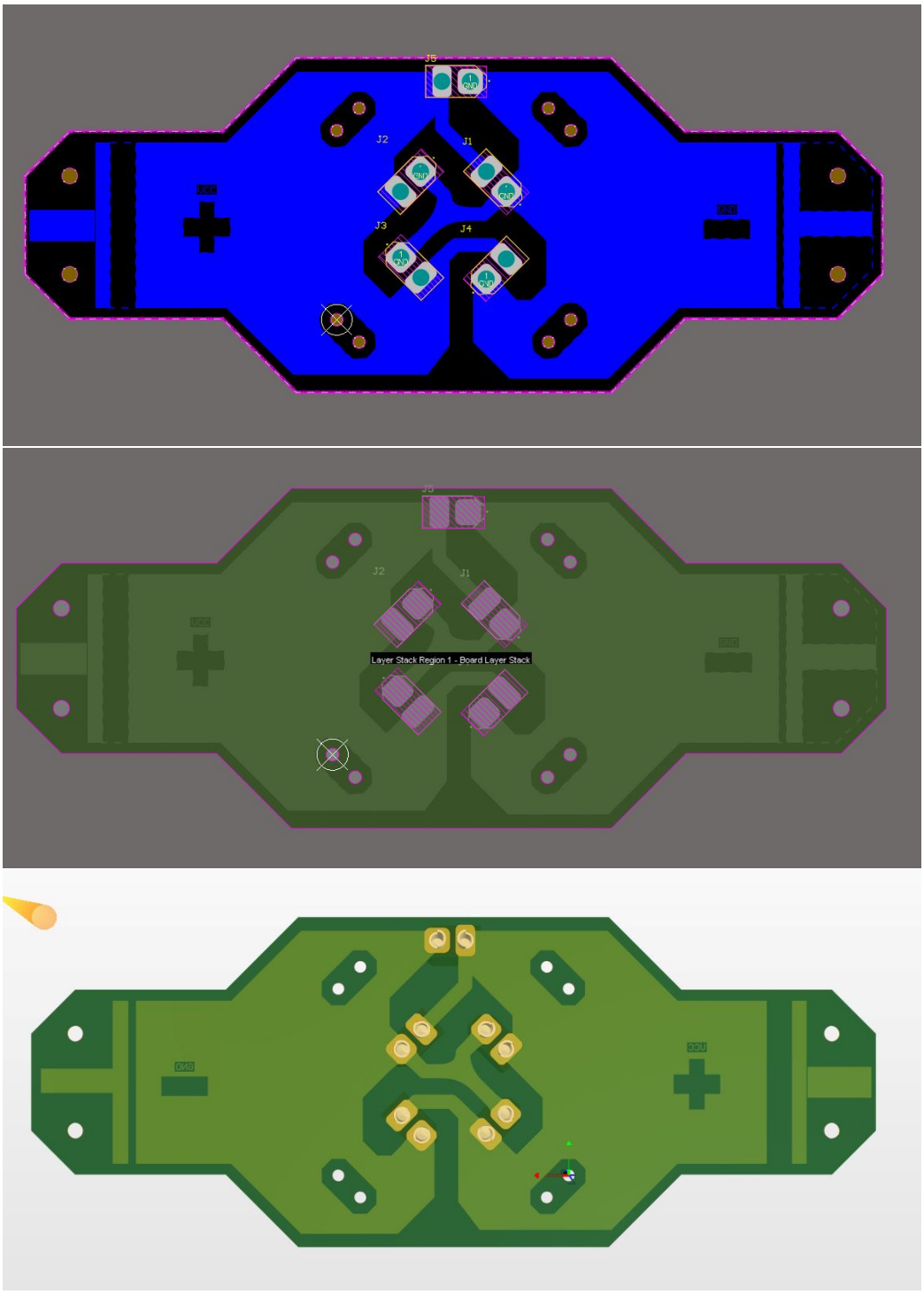


Figura 67: Resultados diseño PCB versión 2, Altium Designer (vista del ruteo y 3D)

7.3. Diseño mecánico del marco

Lo primero que se necesita para poder empezar a realizar pruebas dentro de esta plataforma de 3 grados de libertad, es contar con un marco para montar el dron y que este sea acorde a las limitaciones de espacio de la plataforma física.

7.3.1. Reconociendo la necesidad

Diseñar un marco acorde a las limitaciones de espacio, peso, tiempo de fabricación, generación de torque considerando componentes electrónicos del dron.

7.3.2. Definición del problema

Para poder definir este problema, lo primero que se realizó fue medir los límites físicos de la plataforma, luego con base en esas restricciones de tamaño (largos mínimos y máximos) se diseñaron los brazos del marco del dron, considerando que estos deben ser capaces de generar el mayor torque posible para poder girar al dron dentro de la estructura así como también teniendo en cuenta que el tamaño de la hélice a seleccionar no tope entre ellas mismas o con el anillo exterior de la estructura.

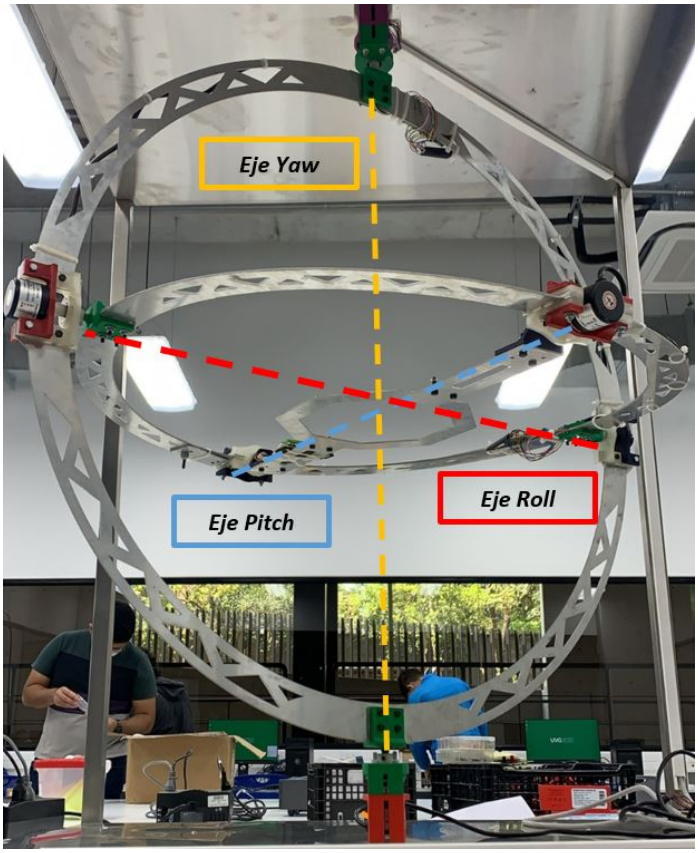


Figura 68: Definición de ejes de la plataforma.

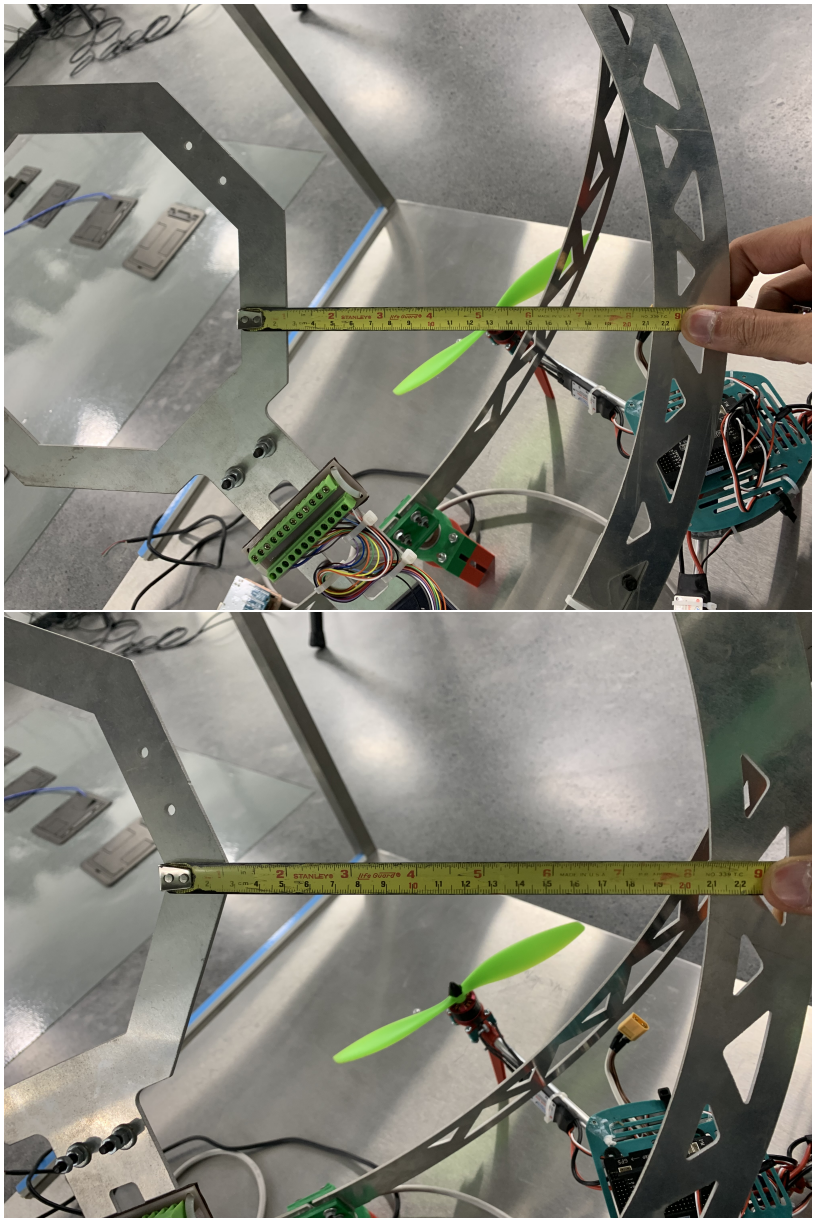


Figura 69: Restricciones plataforma, largos marco en configuración X.

Debido a la geometría de la estructura y a cómo podría llegarse a cablear (de la manera mas simétrica posible para mantener el equilibrio) las conexiones de alimentación y comunicación del dron, cabe mencionar que la modalidad para el marco del dron que se eligió fue la configuración en equis X.

Teniendo en cuenta lo anterior se puede ver en la Figura 69, que dada la configuración del dron escogida tenemos un rango de distancia límite entre el eje del cabeceo (pitch) con el aro interno del alabeo (roll) de 21cm a 21.5cm aproximadamente. Esta es una de las principales limitaciones de la plataforma.

7.3.3. Síntesis y análisis del problema

Iteración 1: Marco del dron

Se utilizó el software de Autodesk Inventor [36] para realizar el diseño tridimensional de los 4 brazos así como también la base superior e inferior que sujetarían a los mismos. Es sumamente conveniente trabajar un archivo de ensamblaje para colocar el marco en donde se tiene previsto montarlo (eje del cabezeo) físicamente. De esa manera se establece un enlace entre el proceso de diseño y manufactura considerando las restricciones de tamaño de los límites máximos-mínimo de longitud del eje del roll que se midieron en las Figura 69 para dicho marco, así como también considerando el tamaño del controlador de vuelo.

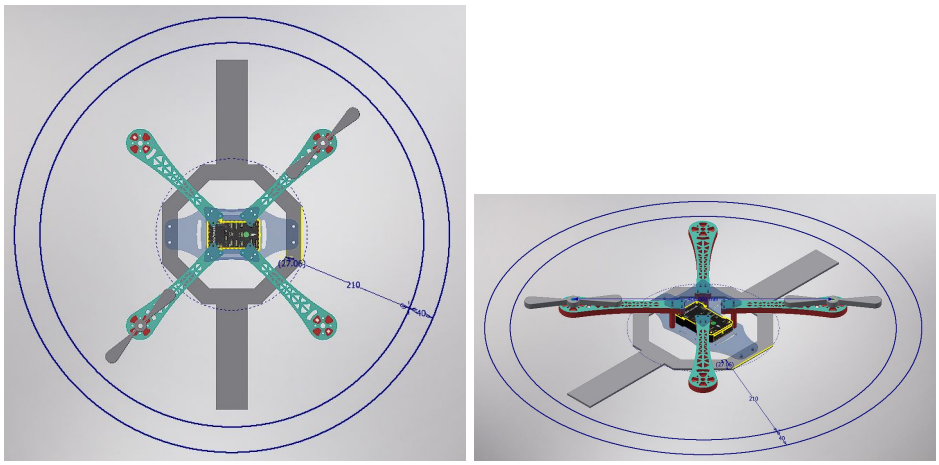


Figura 70: Síntesis iteración 1, diseño marco F406 con hélice 6045.

Para esta primera iteración, ya que no se contaba con ninguna hélice aún, se decidió utilizar la hélice de tamaño de 6 pulgadas (hélices grises en el assembly) ya que era la más barata, accesible de conseguir y acorde al espacio que quedaba con las limitaciones de espacio de la plataforma.

Resultados físicos iteración 1: Marco del dron

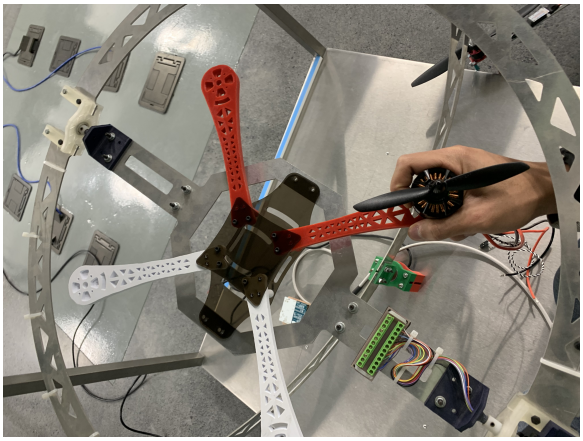


Figura 71: Resultado físico iteración 1, diseño marco hélice 6045.

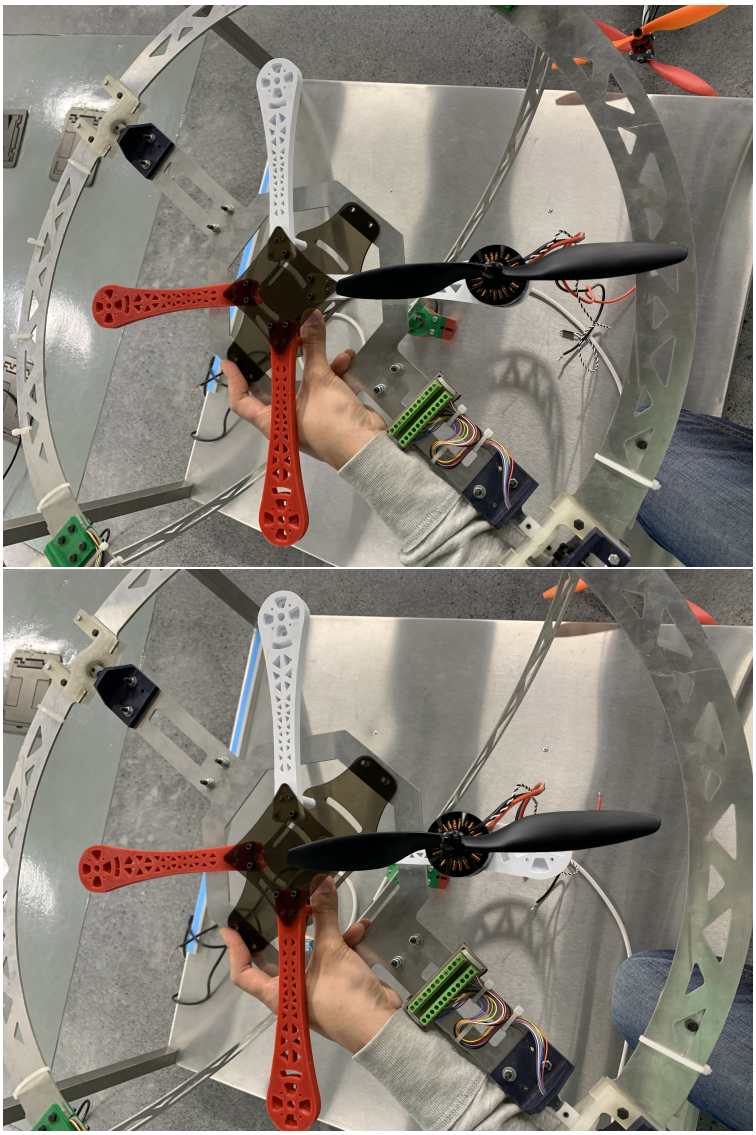


Figura 72: Resultado físico iteración 1, diseño marco hélice 1045.

Iteración 1: Análisis y optimización de los resultados

Como se observa en la Figura 72 (a) la hélice de 10 pulgadas (10x45) no cabe con las dimensiones de ese marco y en la Figura 72 (b) se observa que para poder utilizar esta hélice se debe reducir demasiado el largo del brazo, tanto que si se hace las hélices de la vecindad van a topar con la analizada. Por lo tanto, esto no es factible y hay que buscar otra alternativa que minimice la longitud del dron en la siguiente iteración. Se podría optimizar tanto el largo del brazo como la base superior e inferior de acrílico. Cabe mencionar que dichas bases que sostienen a los 4 brazos se pensaron fabricarlas en acrílico ya que es un material que puede cortarse rápidamente en la cortadora laser de la Universidad.

Iteración 2: Marco del dron

En esta nueva iteración se consideró un tamaño de hélice más grande, ya que el fabricante de los motores utilizados recomienda un tamaño de hélice de 13 a 15 pulgadas esto considerando que el dron va a volar. Para el caso de este trabajo dicho dron no tiene como objetivo volar fuera de la plataforma por ende se puede tomar una hélice mas pequeña que las recomendadas por el fabricante.

En la iteración pasada se tomó una hélice de aproximadamente la mitad de la mínima recomendada, para esta nueva se seleccionó una hélice un poco mas larga que la anterior, esta es de 8 pulgadas (8045). Esto con el objetivo de aprovechar más la fuerza de empuje que pueda realizar este motor con este tamaño de hélice.

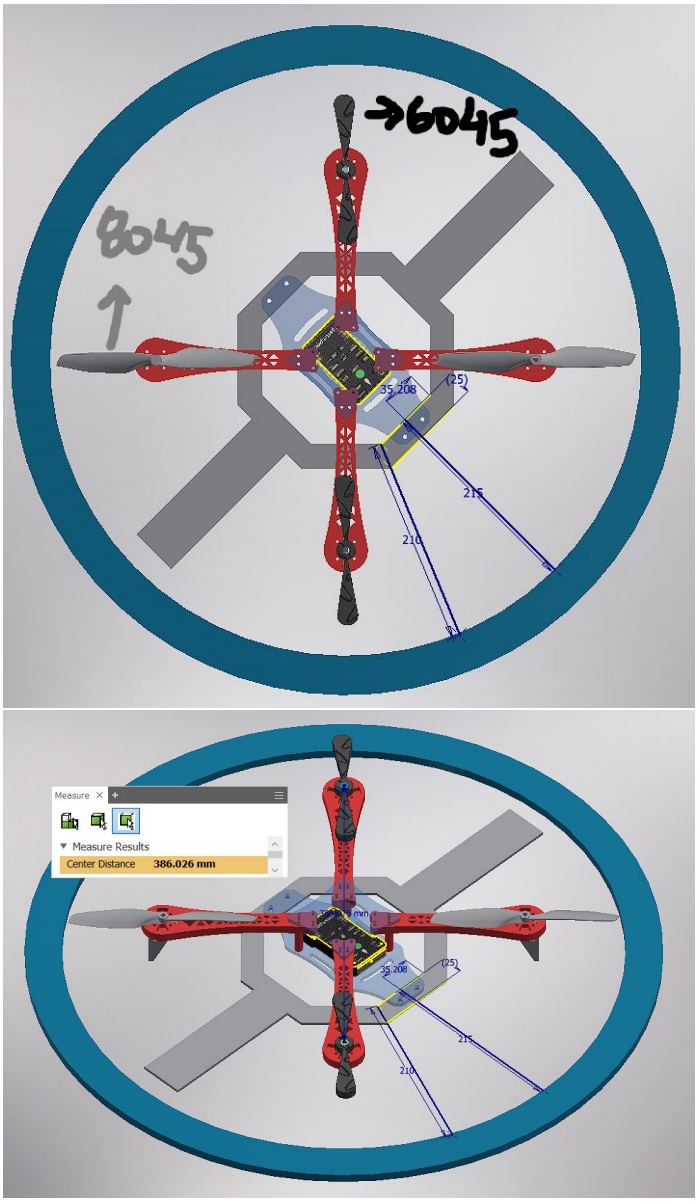


Figura 73: Síntesis iteración 2, diseño marco F386 con hélice 8045(gris) y 6045(negra).

Resultados físicos iteración 2: Marco del dron

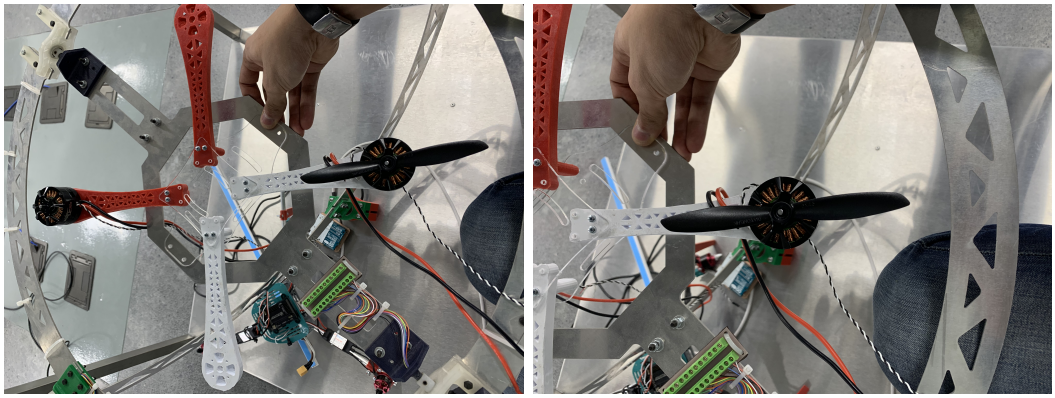


Figura 74: Resultado físico iteración 2 - diseño marco hélice 6045.

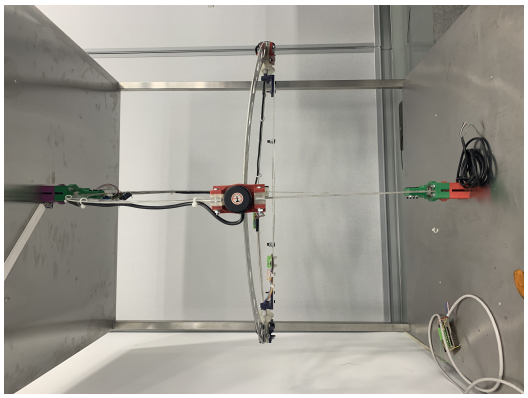


Figura 75: Resultado físico iteración 2 - pandeo diseño marco hélice 8045.

Iteración 2: Análisis y optimización de los resultados

Como se observa en los resultados físicos, la hélice de 6 pulgadas sigue quedando muy bien. Sin embargo, la hélice para la cual se diseñó esta iteración (la de 8 pulgadas) como vemos en la Figura 75, notamos que a pesar de que en el assembly de inventor haya cazado de manera óptima, esto no fue así en la vida real. Al ver la figura anterior observamos que un parámetro que no se tomo en consideración fue el pandeo de la estructura, este pandeo provocó que las hélices de 8 pulgadas toparan en el aro interno del eje del roll cuando la estructura giraba levemente. Por lo tanto hubo que reducir aún más los brazos y la base de acrílico del dron.

Iteración 3: marco del dron

Para esta tercera y última iteración se ajustaron varios parámetros para cumplir con las restricciones que el diseño necesitaba. Para esta iteración lo más importante que se pudo solucionar fue lograr el equilibrio del dron montado en la estructura, además reduciendo levemente la base superior e inferior donde están montados los 4 brazos, se logró solucionar que las hélices de 8 pulgadas ya no toparan con el eje del roll al girar.

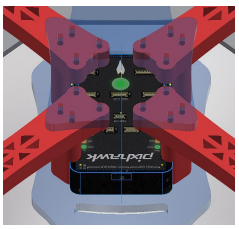


Figura 76: Problema con la ubicación del Pixhawk.

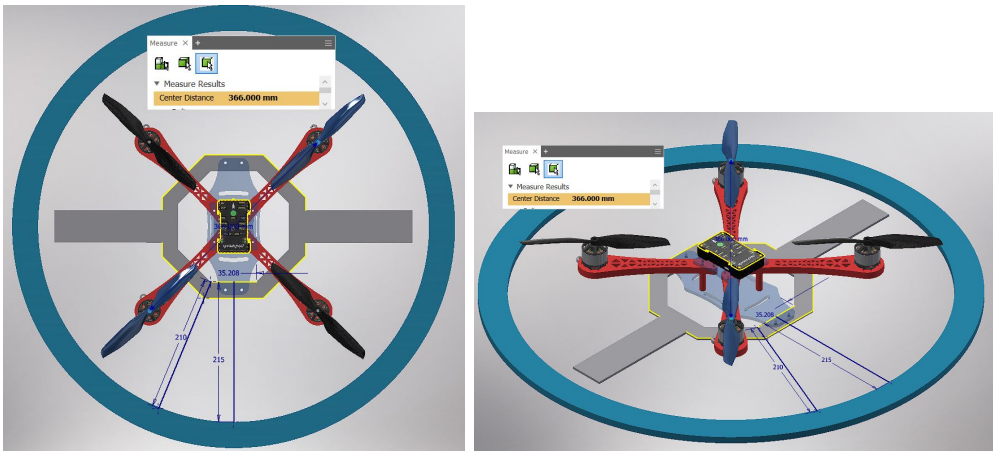


Figura 77: Síntesis iteración 3, diseño marco F366 con hélice 8045.

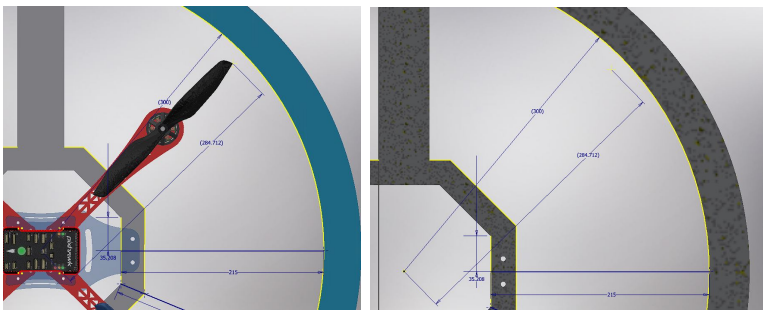


Figura 78: Reducción de 16mm de c/lado, longitud dron con hélice.

Como se observa en la Figura (76), el reducir 1.6cm de cada lado en la longitud de la base, Figura (78), hace que ya no sea posible colocar el Pixhawk en la base inferior, esta colocación del controlador de vuelo era importante ya que así se reducía lo mas posible el centro de gravedad de todo el marco en conjunto. Es sumamente importante tener un centro de gravedad lo más bajo posible para lograr un buen equilibrio.

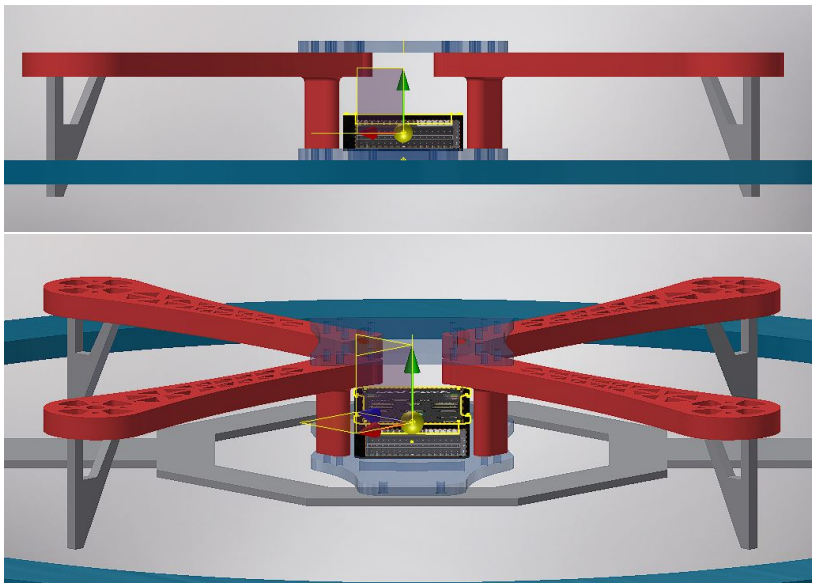


Figura 79: Centro gravedad, marco iteración 2.

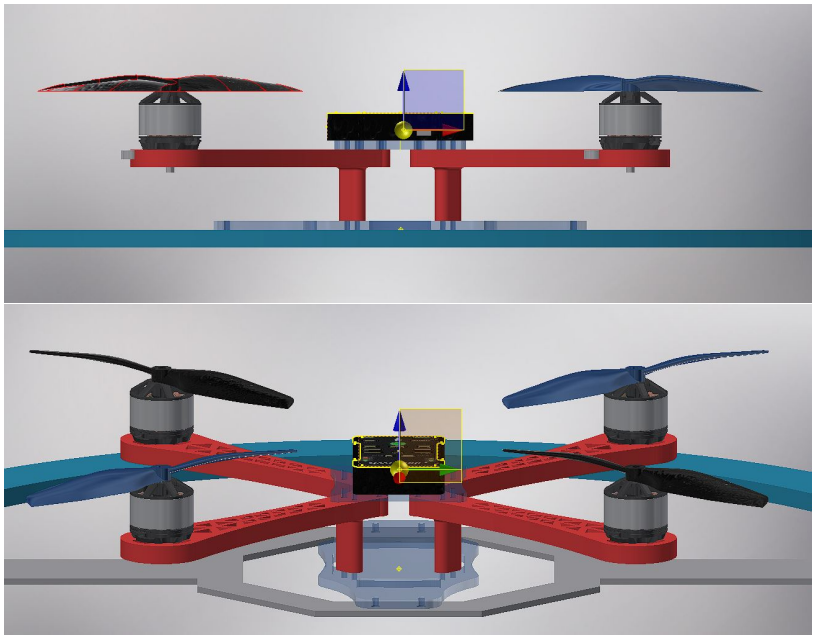


Figura 80: Centro gravedad, marco iteración 3.

Resultados físicos iteración 3: marco del dron

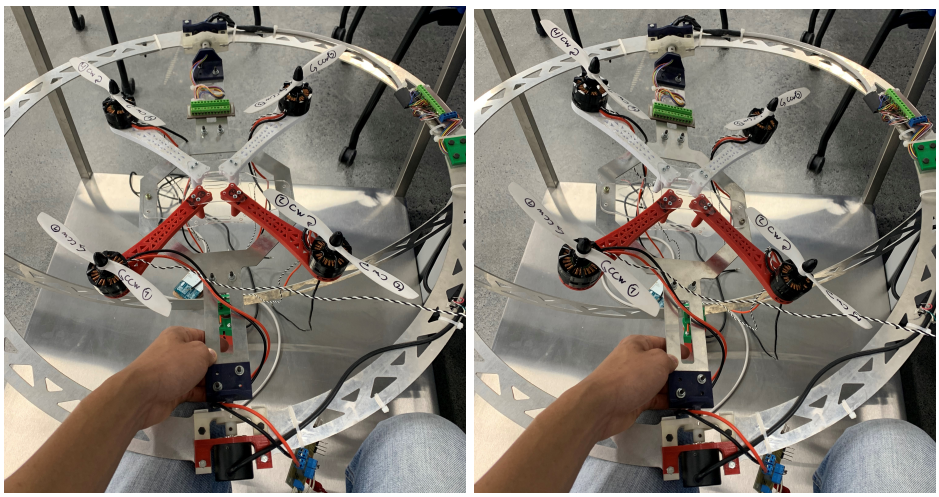


Figura 81: Resultado físico iteración 3, diseño marco hélice 8045.

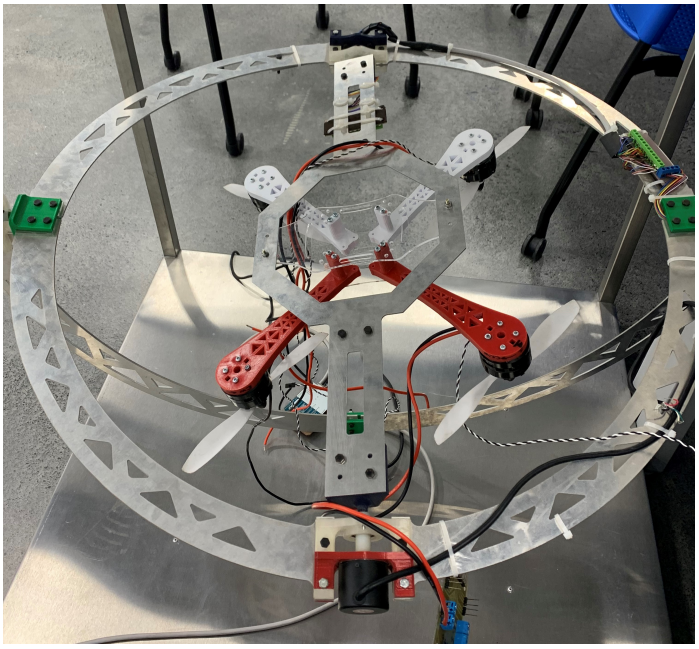


Figura 82: Resultado físico iteración 3, CG inadecuado.

Iteración 3: Análisis y optimización de los resultados

Como se observa se logró solucionar el problema que se tenía con el roce de las hélices de 8 pulgadas debido al pandeo. Sin embargo, debido a que el centro de gravedad del marco en conjunto esta muy elevado de la base el dron junto con la estructura se comportó de manera muy inestable (Figura 83). Hay que ver cómo se soluciona para que vuelva a estar lo mas bajo posible como lo fue en el caso de la iteración 2 (Figura 79). La solución a este problema de equilibrio se arregló fácilmente al ajustar la base del ensamblado 3cm. debajo del eje del cabeceo.

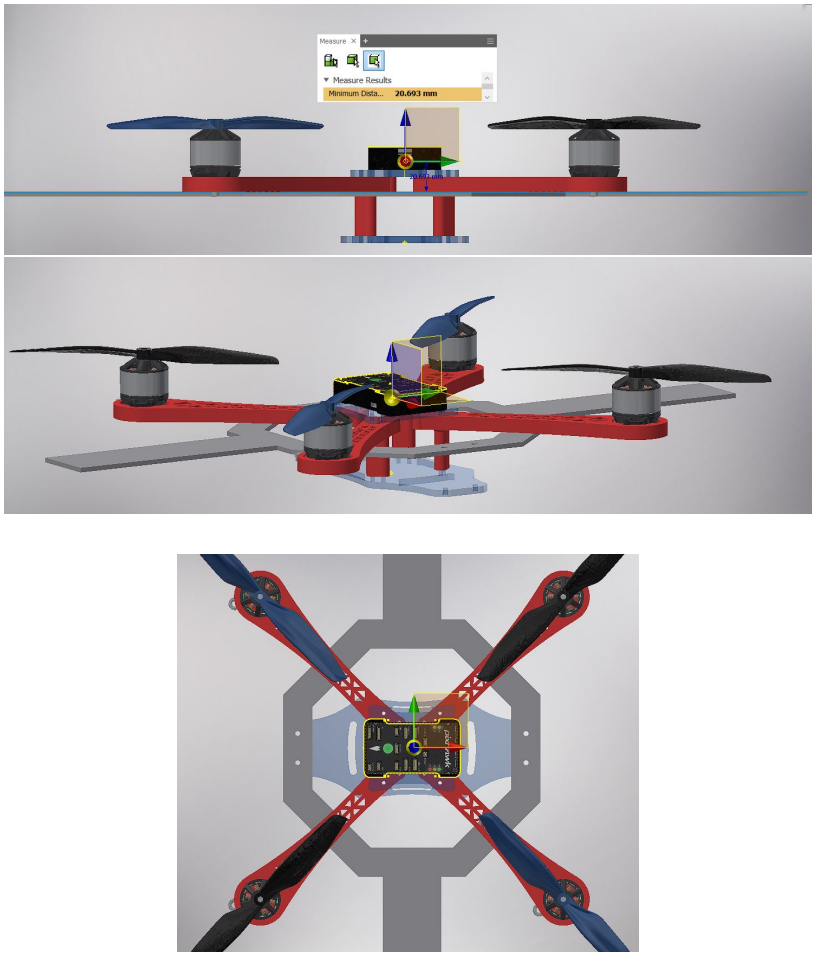


Figura 83: Centro gravedad reducido.

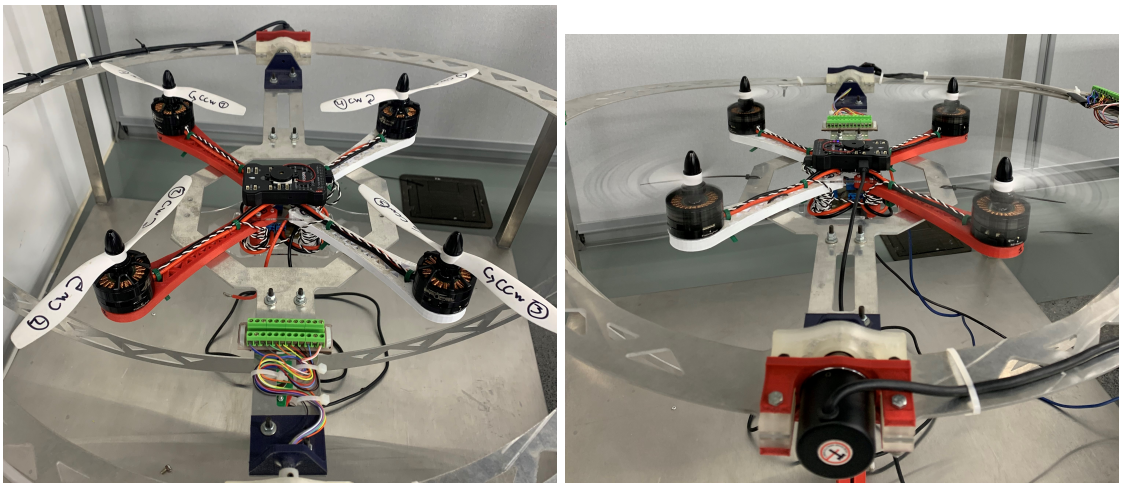


Figura 84: Resultado físico final (a), centro de gravedad reducido.

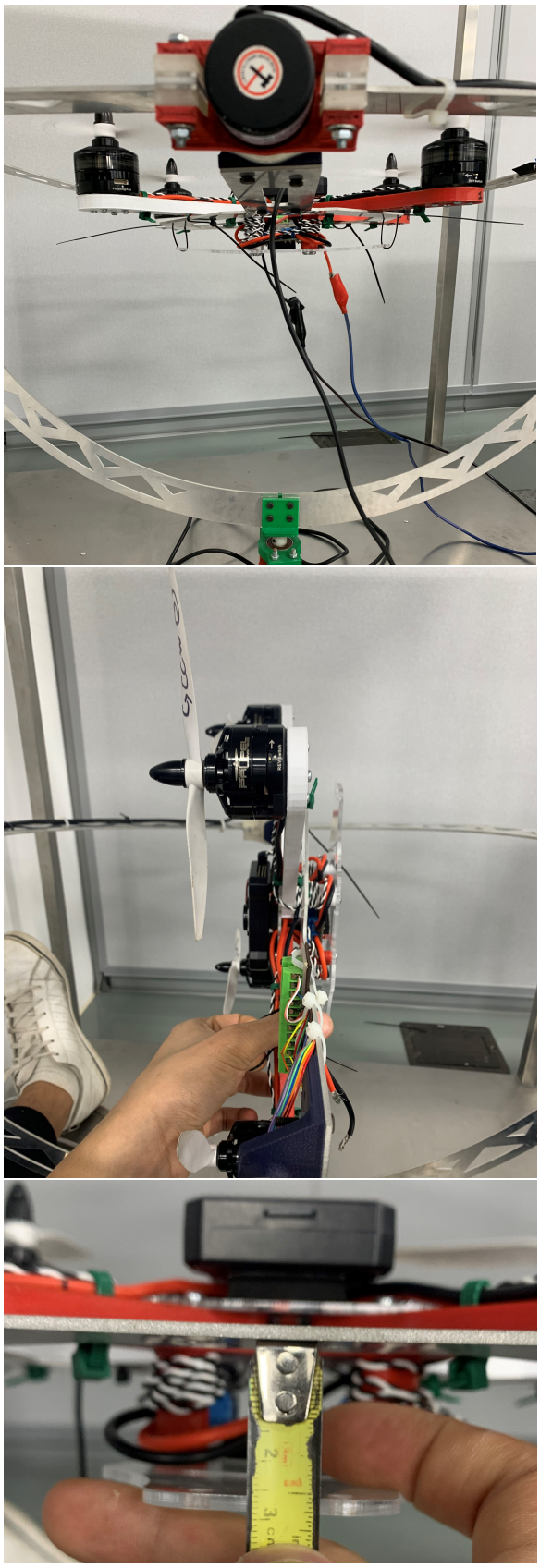


Figura 85: Resultado físico final (b), centro de gravedad reducido.

7.3.4. Validación del marco (Esfuerzos)

Análisis de esfuerzos: Criterio de falla por "Energía de Distorsión (ED)" [29]

Antes de entrar a detalle sobre los cálculos realizados para obtener el factor de seguridad de la pieza se deben establecer algunas suposiciones y simplificaciones del modelo CAD. Los supuestos bajo los cuales se trabajó el análisis del brazo que conforma al dron son los siguientes:

- La pieza se analizará como una viga empotrada en voladizo.
- La sección transversal del brazo es constante y de forma rectangular.
- El porcentaje de relleno de la impresión 3D se tomará como 100 % (pieza sólida).
- Sólo existen fuerzas que generan esfuerzos a flexión (fuerza de empuje de cada motor), no hay ninguna fuerza de torsión.
- Ya que el dron se debe estabilizar en alguna referencia angular, dichas piezas se analizarán bajo carga estática.

Antes de seleccionar que método a utilizar para el análisis de la pieza impresa en 3D sometida bajo carga estática (fuerza de empuje de cada motor) se debe saber si el material para el cual se imprimió dicha pieza es dúctil o frágil. El material que se utilizó fue el plástico *acrilonitrilo butadieno estireno* mejor conocido como ABS ya que de los disponibles en el campus, éste posee mejores propiedades mecánicas que el más utilizado (PLA). Según [37] tanto el ABS como el PLA son materiales impresos que se comportan de manera dúctil.

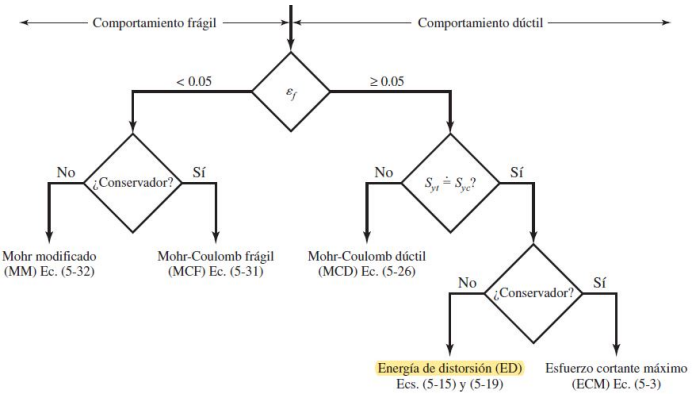


Figura 86: Selección criterio de falla [29]

Debido al comportamiento de la estructura y como el dron se va a comportar dentro de ella se puede modelar que cada brazo se puede simplificarse al análisis de una viga empotrada en voladizo. Se eligió el camino no conservador para obtener resultados mas adecuados, por lo tanto, como se ve en la Figura 86 se seleccionó el criterio de energía de distorsión para nuestro análisis. En la pág. 225 de [29] se presenta un ejemplo (5-4) utilizando dicho criterio, cabe mencionar que la pieza está sometida únicamente a flexión por la carga de la fuerza de empuje de cada motor.

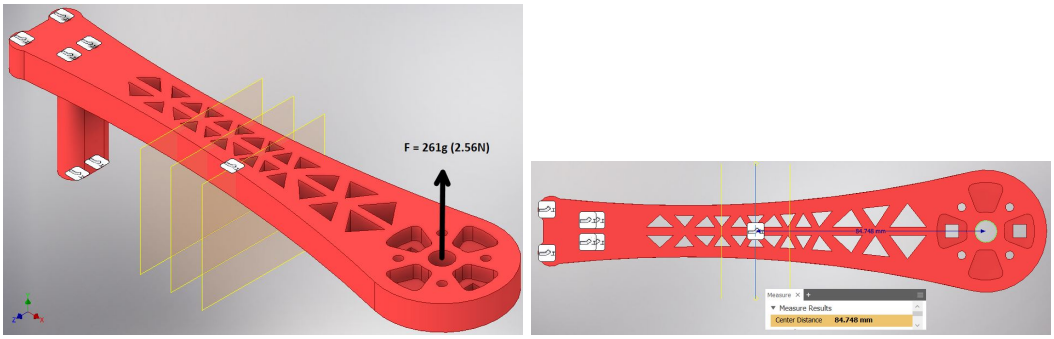


Figura 87: DCL brazo, fuerza motor Hitec 390kv.

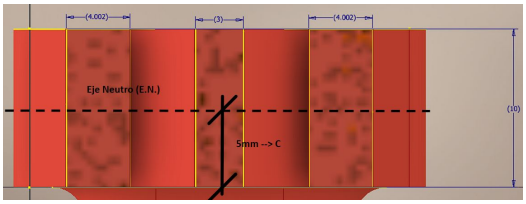


Figura 88: Área de la sección transversal plano 2, Figura 87.

Las ecuaciones a utilizar para el caso en particular son las siguientes:

$$\sigma_x = \frac{P}{A} + \frac{MC}{I} = 0 + \frac{(d_{brazo}F)(C)}{I}, \tag{50}$$

$$\sigma' = \left(\sigma_x^2 + 3\tau_{zx}^2\right)^{1/2} = \left(\sigma_x^2 + 0\right)^{1/2} = \sigma_x, \tag{51}$$

$$n = \frac{S_Y}{\sigma'}, \tag{52}$$

donde:

- P: fuerza axial eje X (perpendicular ala sección transversal analizada no existe).
- A: área de la sección transversal analizada.
- τ_{xx} : Esfuerzo cortante debido a torque al rededor del eje Y (no existe).
- σ': esfuerzo de Von Mises.
- M: momento flector generado por la fuerza axial eje Y del motor.
- C: distancia del eje neutro a la fibra más alejada de la sección transversal.
- I: momento de inercia de la sección transversal (rectángulo).
- n: factor de seguridad
- S_Y: Esfuerzo de fluencia mínimo del material.

Cálculo factor de seguridad - Marco con motor 1

Teniendo en cuenta que la fuerza máxima que puede generar el motor de 390kv es de 261 gramos se utilizará dicha fuerza pero en Newtons, al multiplicar el valor en gramos por 0.009807 se obtiene una fuerza da 2.56 Newtons.

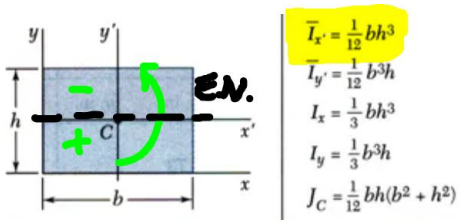


Figura 89: Momento de inercia rectángulo [38].

Sumando la base de los 3 rectángulos de la Figura 88 se obtiene una base del rectángulo total de 11.004mm con una altura de 10mm. El momento de inercia de la sección transversal $Ix = \frac{1}{12} * bh^3$ queda entonces como:

$$Ix = \frac{1}{12}(11.004mm\frac{1m}{1000mm})(10mm\frac{1m}{1000mm})^3 = 9.17e-10m^4. \tag{53}$$

Utilizando la ecuación (50) y la distancia del brazo de palanca de 84.748mm (0.085m) de la Figura 87(b) se obtiene el momento de flexión al cual está sometido el brazo con la fuerza máxima de los motores de 2.51Newtons. Ya que no hay ningún esfuerzo normal axial sobre el eje X, sólo tendrá esfuerzos de flexión.

$$\sigma_x = \frac{(d_{brazo}F)(C)}{I} = \frac{(0.085m2.51N)(0.005m)}{9.17e-10m^4} = 1.18e6N/m^2 = 1.18MPa. \tag{54}$$

Teniendo en cuenta que el ABS tiene un rango de esfuerzo de fluencia de 13-65Mpa [39] se tomará el mínimo ya que es el caso crítico. Las propiedades mecánicas de este material se pueden observar en la Figura 90. Finalmente, con todos estos datos se puede obtener el F.S. con la ecuación (52) sabiendo que al no tener esfuerzo cortante el esferuzo de Von Mises es igual al esfuerzo de flexión de 1.18MPa.

- **Factor de seguridad, brazo ABS**

$$n = \frac{13Mpa}{1.18MPa} = 10.99 \approx 11. \tag{55}$$

Como era de esperarse debido a que la fuerza de empuje del motor no es muy grande, se obtuvo un factor de seguridad bastante alto, por lo que se aseguró que al tener un F.S superior a 3 se puede garantizar que la pieza no va a fallar debido a la carga ejercida por el motor. El Cuadro 5 muestra un resumen de los datos obtenidos para el factor de seguridad.

Overview of materials for Acrylonitrile Butadiene Styrene (ABS), Extruded

Categories: Polymer; Thermoplastic; ABS Polymer; Acrylonitrile Butadiene Styrene (ABS), Extruded

Material Notes: This property data is a summary of similar materials in the MatWeb database for the category "Acrylonitrile Butadiene Styrene (ABS), Extruded". Each prop report the average value, and number of data points used to calculate the average. The values are not necessarily typical of any specific grade, especially

Vendors: **Click here to view all available suppliers for this material.**

Please click here if you are a supplier and would like information on how to add your listing to this material.

Printer friendly version Download as PDF Download to Excel (requires Excel and Windows)
Export data to your CAD/FEA program

Physical Properties	Metric	English
Density	1.01 - 1.20 g/cc	0.0365 - 0.0434 lb/in³
Water Absorption	0.050 - 1.0 %	0.050 - 1.0 %
Moisture Absorption at Equilibrium	0.00 - 0.30 %	0.00 - 0.30 %
Water Absorption at Saturation	0.30 - 1.03 %	0.30 - 1.03 %
Maximum Moisture Content	0.010 - 0.15	0.010 - 0.15
Linear Mold Shrinkage	0.0020 - 0.0080 cm/cm	0.0020 - 0.0080 in/in
Linear Mold Shrinkage, Transverse	0.0030 - 0.0080 cm/cm	0.0030 - 0.0080 in/in
Melt Flow	0.10 - 35 g/10 min	0.10 - 35 g/10 min

Mechanical Properties	Metric	English
Hardness, Rockwell R	68 - 118	68 - 118
Ball Indentation Hardness	65.0 - 110 MPa	9430 - 16000 psi
Tensile Strength, Ultimate	22.1 - 74.0 MPa	3210 - 10700 psi
Tensile Strength, Yield	13.0 - 65.0 MPa	1890 - 9430 psi
	22.1 - 59.3 MPa @Temperature -18.0 - 71.0 °C	3210 - 8600 psi @Temperature -0.400 - 160 °F
Elongation at Break	3.0 - 150 %	3.0 - 150 %
Elongation at Yield	0.62 - 30 %	0.62 - 30 %
Modulus of Elasticity	1.00 - 2.65 GPa	145 - 384 ksi
	1.50 - 2.60 GPa @Temperature -18.0 - 71.0 °C	218 - 377 ksi @Temperature -0.400 - 160 °F

Figura 90: Propiedades mecánicas del ABS extruido [39].

Factor de Seguridad Brazo Motor 1		
C (fibra mas alejada del E.N)	0.005	metros
Momento flector	0.217	N-m
Distancia de flexión	0.085	Metros
Fuerza (gramos)	261	Gramos
Fuerza (gramos a Newtons)	2.56	Newtons
Inercia Secc. Trans Simplificada	9.17E-10	metros^4
***Esfuerzo Normal (flexión)**	**1.18E+06** **1.18E+00**	**Pa** **Mpa**
Esfuerzo Von Misses	1.18E+00	Mpa
Esfuerzo de fluencia mín [Sy]	13	Mpa
Esfuerzo de fluencia máx [Sy]	65	Mpa
Factor de Seguruidad Crítico n_min	**10.99**	**11**
n_max	54.95	55

Cuadro 5: Factor de seguridad del brazo con motor de $390Kv$.

Análisis de esfuerzos: Simulación Inventor y ANSYS

Para la simulación del análisis de estrés de la pieza, se utilizarán dos programas: Autodesk Inventor y ANSYS. El objetivo realizar esto en dos programas diferentes fue poder comparar de mejor manera ambos resultados con los obtenidos teóricamente. Además, cabe mencionar que Inventor es un programa dedicado al CAD y ANSYS es un programa dedicado al análisis de elementos finitos.

Para ambas simulaciones es importante mencionar que debido al problema con el centro de gravedad se añadió un soporte fijo en la estructura física para cada uno de los brazos, tal como se ve en la iteración final del marco del dron. De la misma manera fue considerado tanto en la simulación de inventor como en la de ANSYS tal como se muestra en la siguiente imagen.

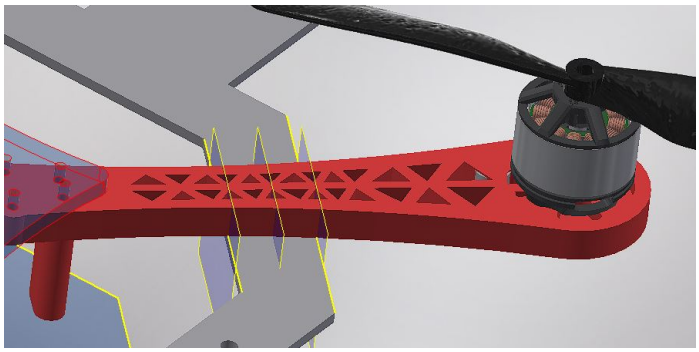


Figura 91: Sujeción fija nueva.

Con la proyección del eje del cabeceo se proyecto el espesor de la placa de aluminio y se crearon dos planos de trabajo para luego con la herramienta de *split* fuera posible crear la restricción del apoyo fijo mostrado en la Figura 87 (a).

También se cambiaron las propiedades mecánicas del material a las utilizadas para los cálculos teóricos dados la página de MatWeb para el ABS 90. La tabla modificada en Inventor se muestra a continuación.

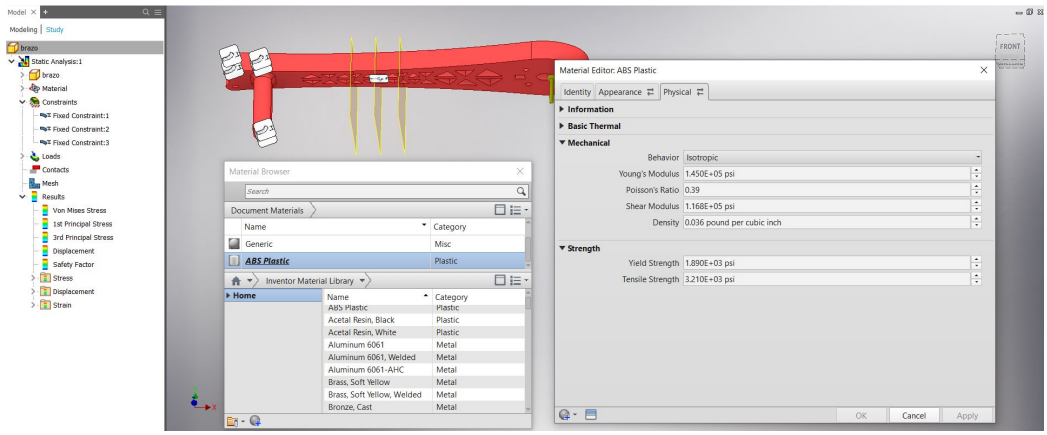


Figura 92: Edición propiedades mecánica del material.

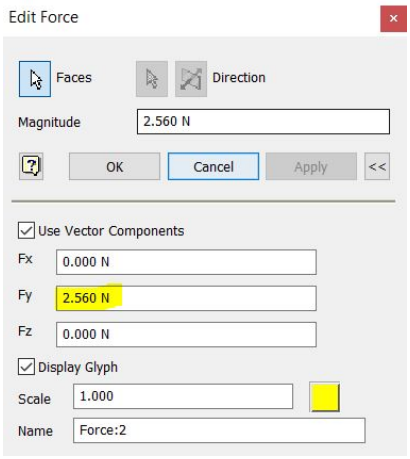


Figura 93: Simulación Inventor, fuerza de empuje.

Resultados simulación - Inventor.

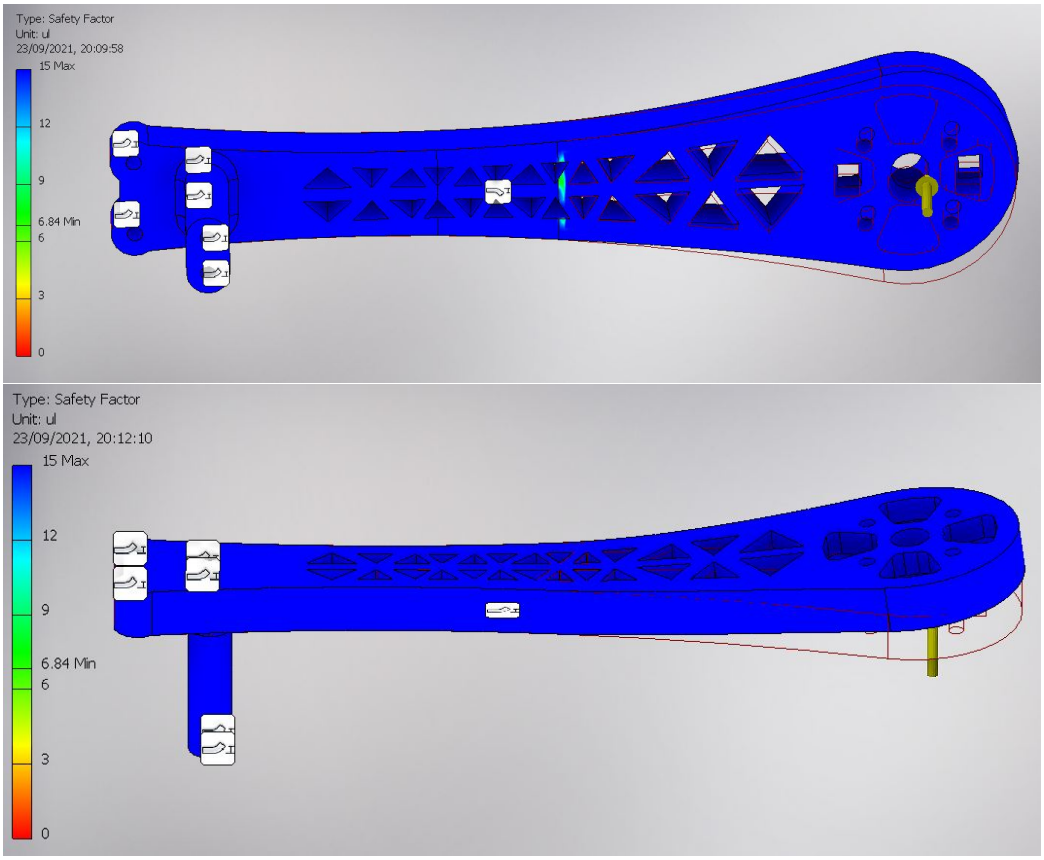


Figura 94: Simulación Inventor, factor de seguridad.

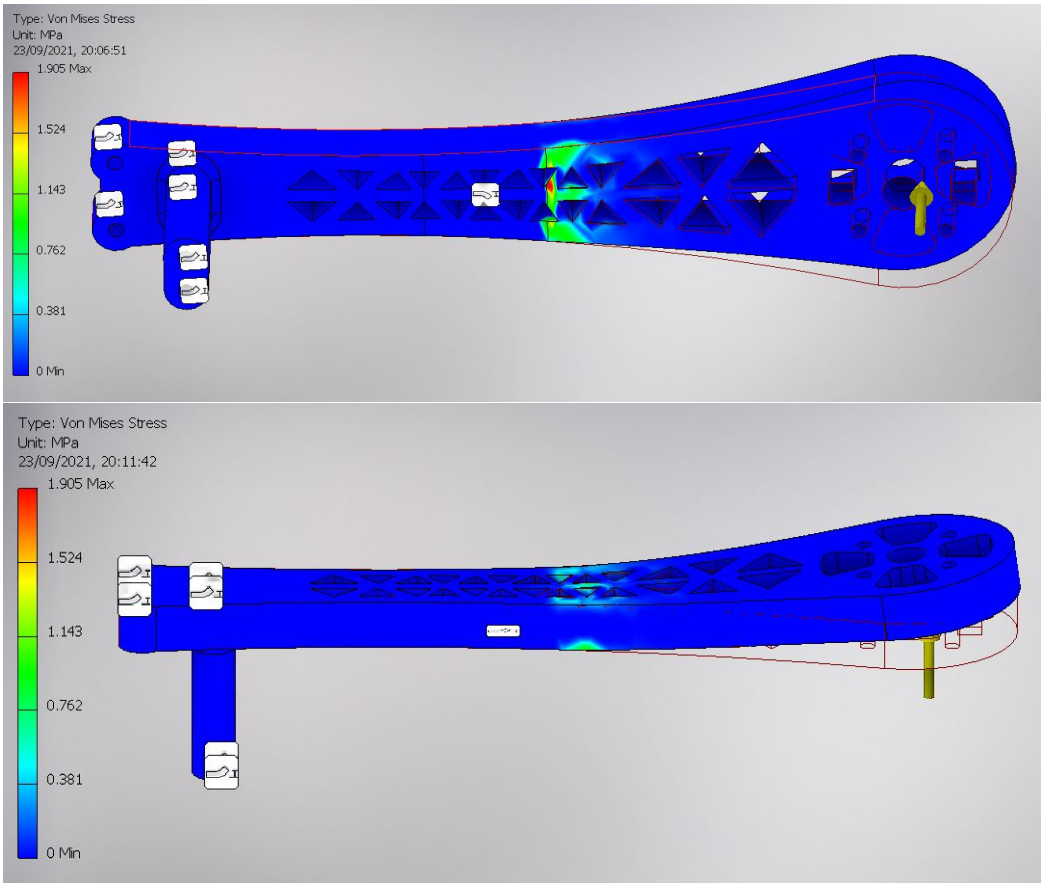


Figura 95: Simulación Inventor, esfuerzo de Von Mises (igual al de flexión).

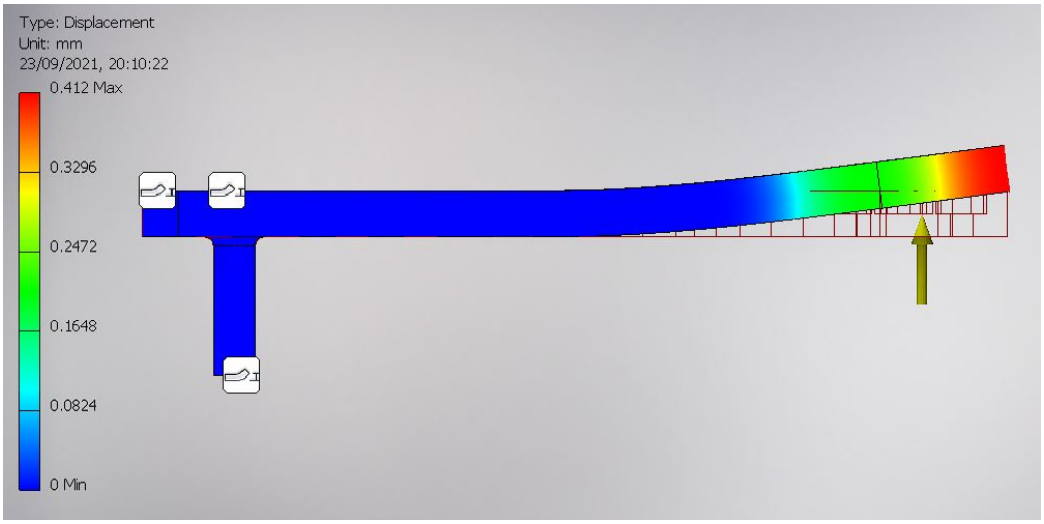


Figura 96: Simulación Inventor, deflexión máxima.

Resultados simulación - ANSYS

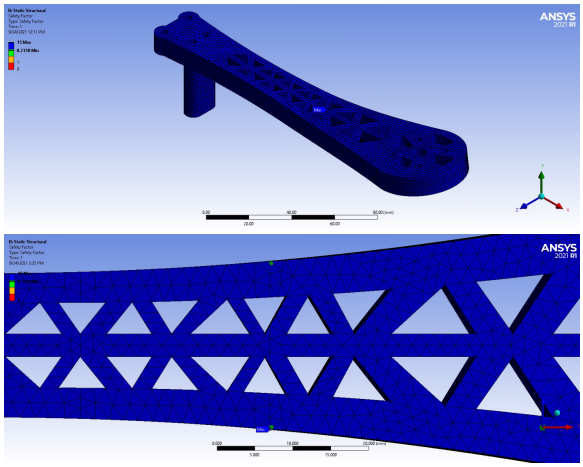


Figura 97: Simulación ANSYS - factor de seguridad.

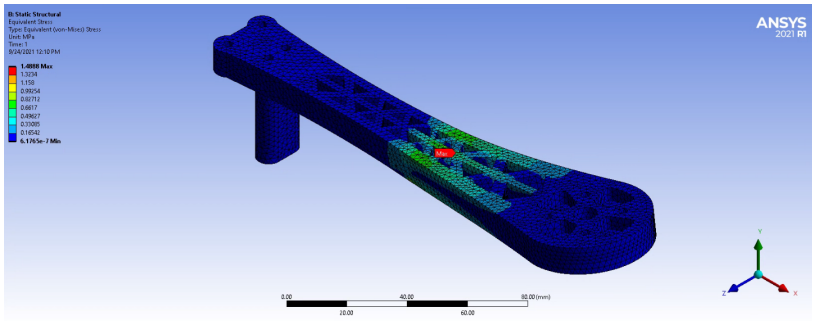


Figura 98: Simulación ANSYS - esfuerzo de Von Mises (igual al de flexión).

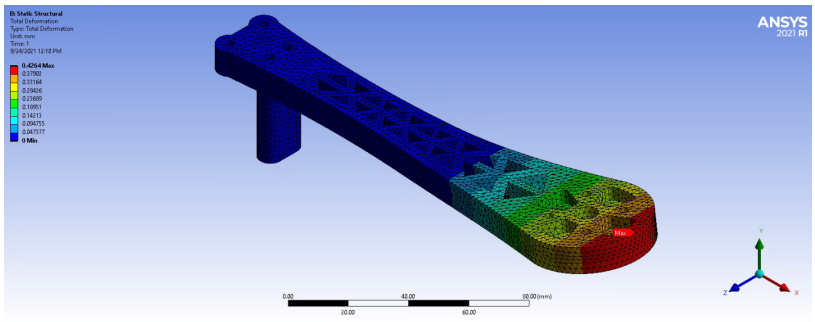


Figura 99: Simulación ANSYS - deflexión máxima.

Como se observa en los resultados de la simulación de Inventor y de ANSYS, para ambas coincide el F.S calculado teóricamente. Al ver la gráfica de coloración se observa que ambos factores son aproximadamente 11 dentro del rango de coloración. En cuando al esfuerzo de Von Mises se nota que ANSYS esta más cerca del valor teórico de 1.18MPa, sin embargo, las discrepancias entre los valores teóricos y los de las simulaciones se deben principalmente a las simplificaciones realizadas.

CAPÍTULO 8

Software implementado

Como se discutió en la sección 6.4.3 del capítulo 6 de este trabajo, el controlador de vuelo con el que se está trabajando puede ser editado de manera muy eficiente mediante el diseño basado en modelos (MBD) en bloques de Matlab/Simulink. En este capítulo se verá con mayor detalle la implementación de este paquete de soporte para el controlador de vuelo de PX4 (Pixhawk 1).

En particular, para los propósitos del trabajo de esta plataforma según [5], utilizar este paquete de soporte de PX4 Matlab/Simulink para editar y modificar el firmware del controlador de vuelo es un buen punto de partida. Además de esto, este paquete utiliza el mismo enfoque de implementación de un *Sliding Mode Control (SMC)* otorgando la seguridad suficiente de que la este auto-piloto (Pixhawk) podría manejar complejidad de los algoritmos de control avanzados [5]. Los requerimientos para poder utilizar este paquete de soporte son los siguientes:

- Matlab 2019a en adelante (2021a recomendada).
- Simulink (entorno de programación visual).
- Mathworks Embedded Coder Support Package.
- Mathworks Simulink Coder Support Package.
- UAV Toolbox.

Cabe mencionar que, además de los requerimientos anteriormente listados para utilizar correctamente el paquete de soporte de PX4 con Simulink, hay que instalar todo lo que nos indica la guía del Anexo. Dichos pasos detallados se muestran desde la Figura 153 a la 173.

Una vez instalado el paquete de soporte *"UAV Toolbox Support Package for PX4 Autopilots"* de manera adecuada en la versión de Matlab. Los ejemplos disponibles para la versión 2019a son los mostrados en la Figuras 100 y 101.

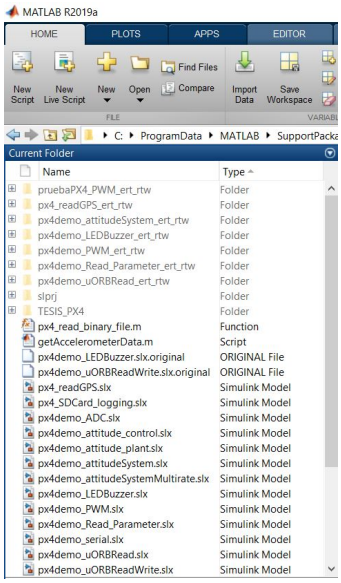


Figura 100: Ejemplos disponibles instalados del paquete de PX4, Matlab 2019a.

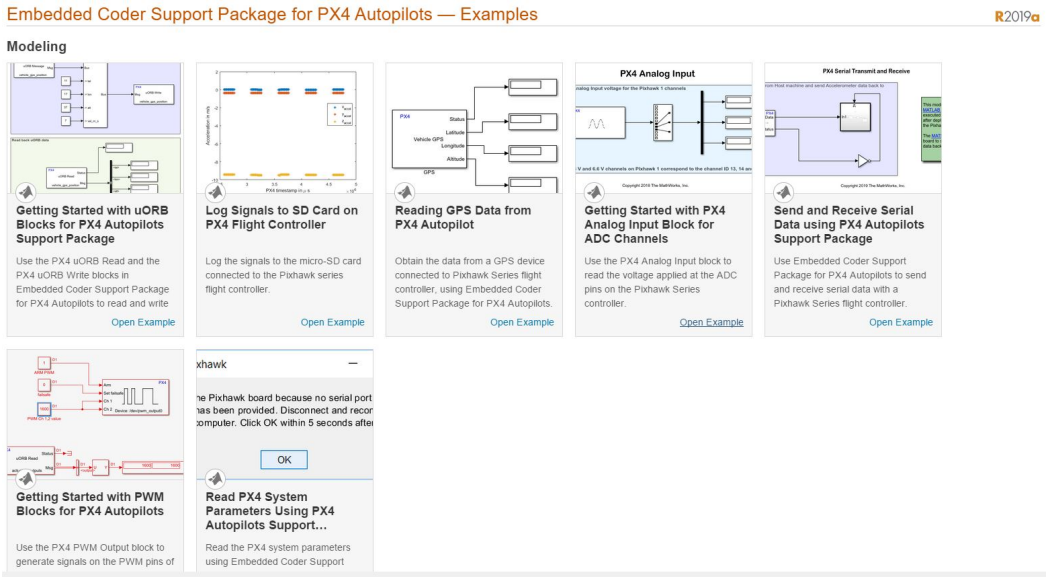


Figura 101: Ejemplos de Mathworks disponibles instalados del paquete de PX4, Matlab 2019a.

8.1. Interactuando con el paquete de soporte del auto-piloto PX4

Para verificar que sí se puede cambiar el firmware del controlador de vuelo desde Simulink se probaron tres ejemplos sencillos.

El primero es el más sencillo de todos, básicamente con este ejemplo se puede cambiar de color el LED incorporado en el auto-piloto, además de cambiar el tono del buzzer conectado a él. El segundo ejemplo que se vio, fue el de modificar una señal de PWM, que son los principales para poder controlar los actuadores del dron que en este caso son los motores sin escobillas de 390kv. En el último ejemplo se utilizó el bloque del giroscopio incorporado en la IMU del controlador de vuelo, esto con el objetivo de poder verificar que al girar el dron en los 3 ángulos (roll, pitch, yaw) se pueda corroborar con respecto a que eje se está moviendo el vehículo.

8.1.1. Ejemplo 1: PX4 LED and Buzzer Control on Pixhawk 1

Para poder verificar este ejemplo se hicieron dos cambios de color del LED uno azul (opción 3) y el otro verde (opción 2) respectivamente.

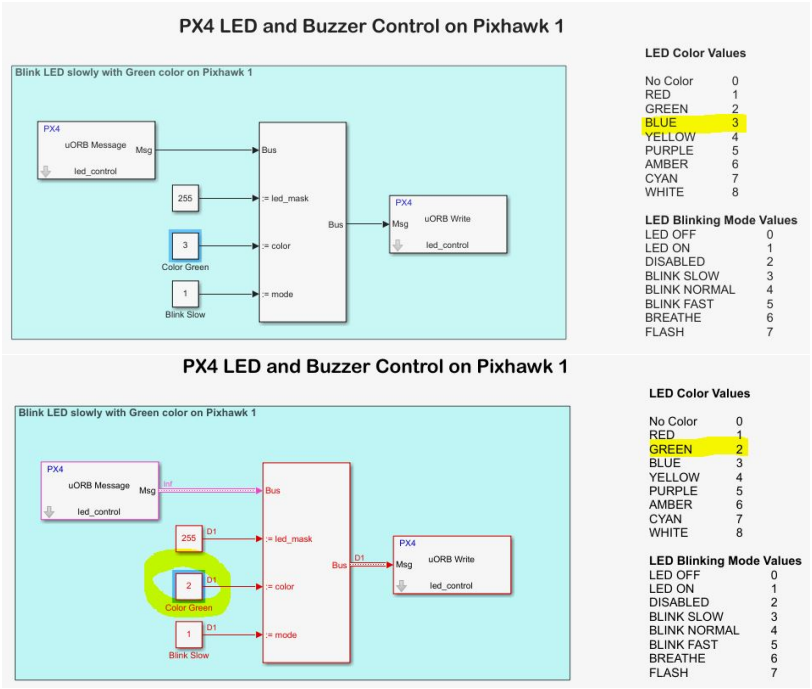


Figura 102: Bloques Simulink ejemplo 1: Control LED (azul y verde).

Antes de poder cargar el bloque de código de Simulink al hardware objetivo (Pixhawk 1) hay que seleccionar el puerto COM correcto asignado para el controlador. En este caso es el COM6, esto se muestra en la Figura 104.

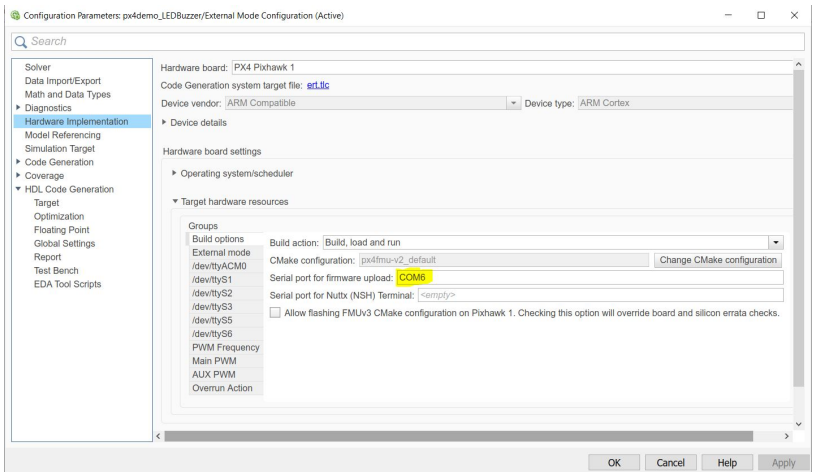


Figura 103: Configuración del hardware objetivo en Simulink.

Luego de un momento hay que darle clic en aceptar en la ventana emergente de reseteo del controlador y reconectar el mismo. Luego de esto Matlab abrirá el reporte del código generado por el paquete de Embedded Coder que se encargó de convertir el código de bloques de Simulink a código en C++. Esto se visualiza en las Figuras 104 y 105.

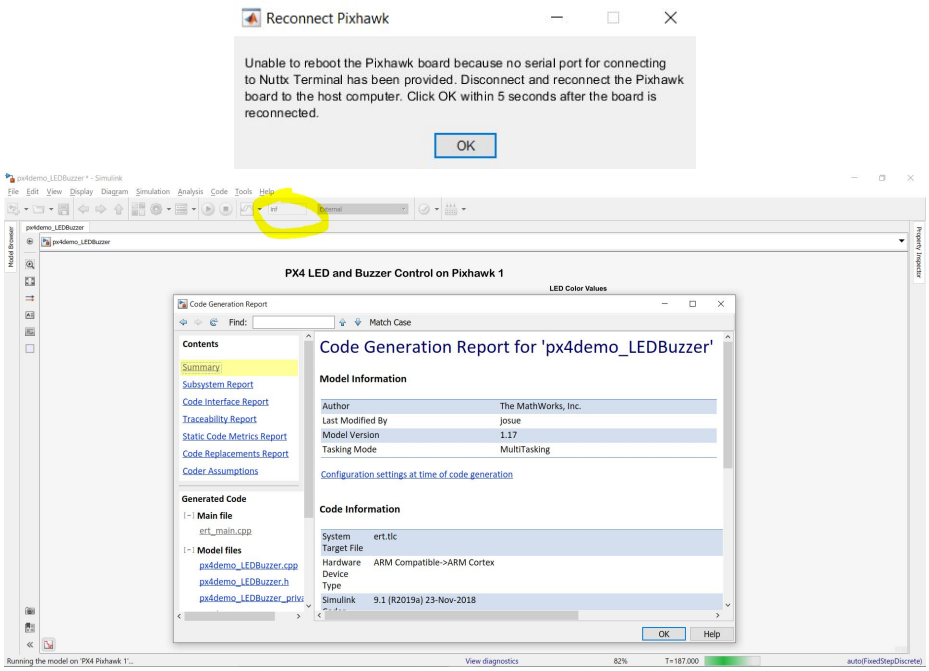


Figura 104: Ventana de reseteo y reporte del código generado en C++.

Se presiona aceptar y se ve cómo el led abordo del controlador de vuelo cambia respectivamente a las opciones seleccionadas.

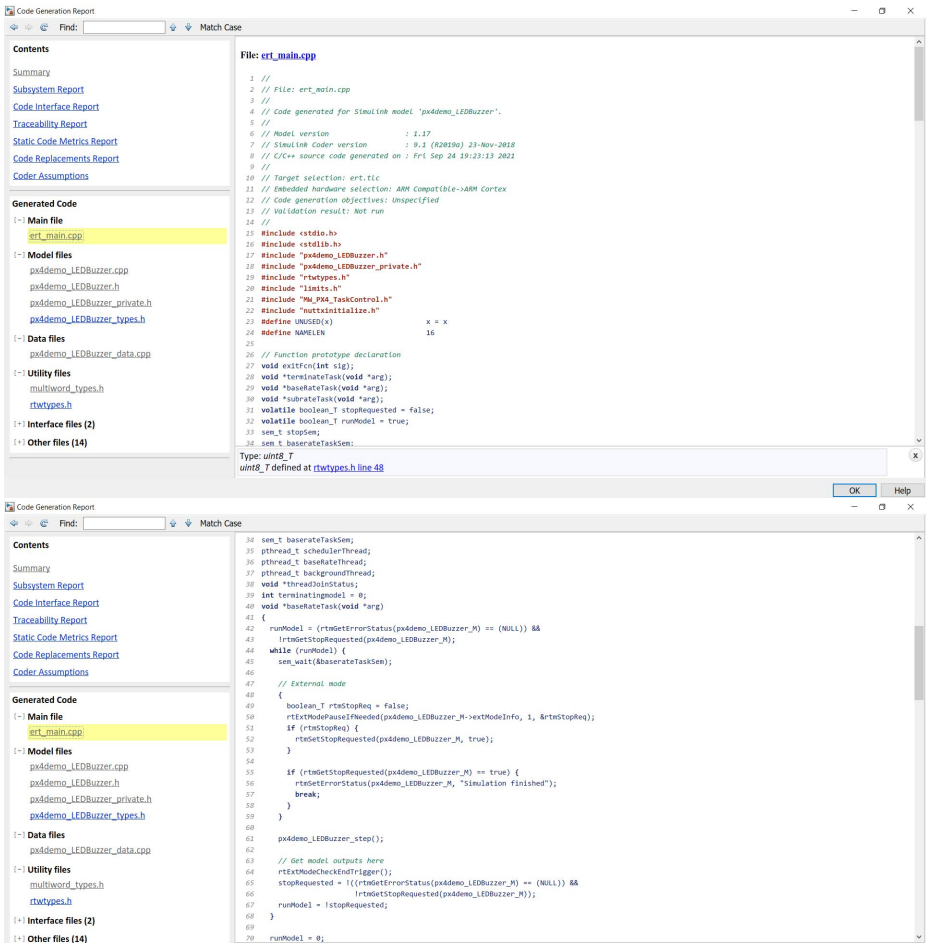


Figura 105: Código generado en C++ por el paquete de Embedded Coder

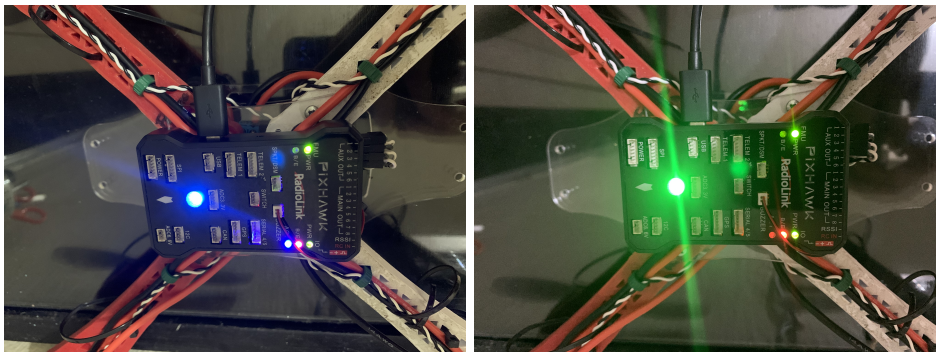


Figura 106: Resultado Ej. 1 LED azul y verde respectivamente.

8.1.2. Ejemplo 2: PX4 PWM - Motor Prueba

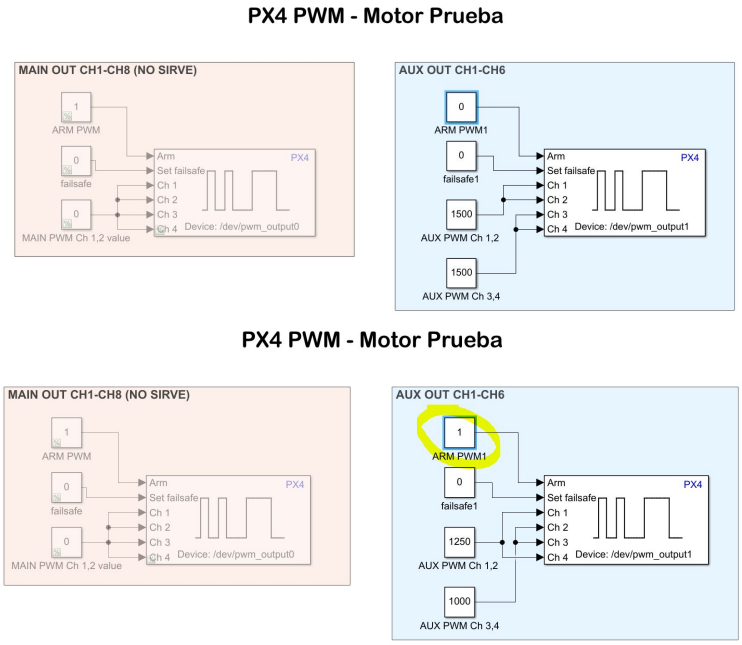


Figura 107: Bloques Simulink ejemplo 2: PWM para motores brushless.

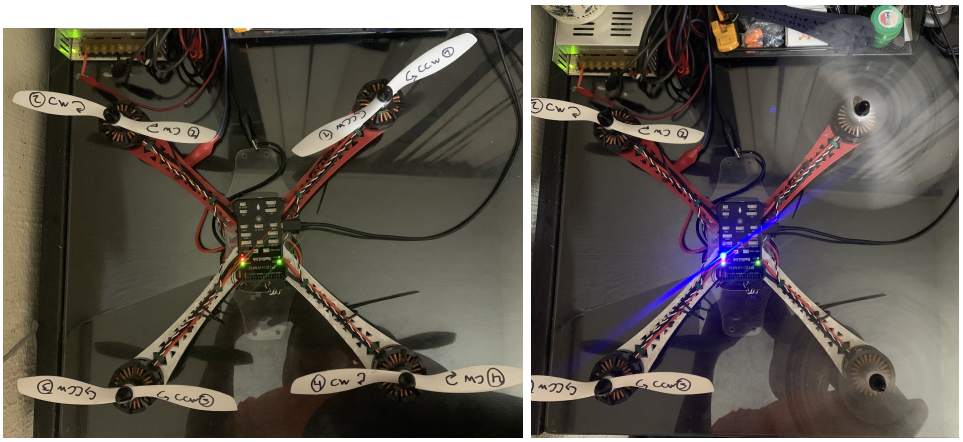


Figura 108: Ejemplo 2: Motores 1 y 4 funcionando (RPM min - 1250PWM).

Cabe mencionar que para poder activar los motores a la señal de PWM establecida hay que cambiar la señal booleana de 0 a 1 para la entrada de ARM como se ve en la entrada encerrada en amarillo de la Figura 107 (b). Como se observa en la Figura 108 los únicos dos motores que están funcionando son los de la entrada AUX. CH1 y CH4 que corresponden a los motores 1 y 4 del marco del dron. Los otros dos motores tienen un valor de 1000 en la señal del PWM y ya que los motores empiezan a girar con la hélice en un valor de aproximado de para el PWM, es por esto que se ve a los otros dos motores sin girar. Cabe mencionar que por algún daño al controlador los canales principales (bloques en rojo) para la señal del "PX4 PWM Output" no sirven, debido a esto se utilizó únicamente las salidas auxiliares (bloques en celeste).

8.1.3. Ejemplo 3: Bloque del sensor giroscopio

El bloque del sensor Giroscopio lee el tema sensor_gyro uORB y genera la velocidad de rotación del controlador de PX4 a lo largo de los ejes x, y y z en rad/s.

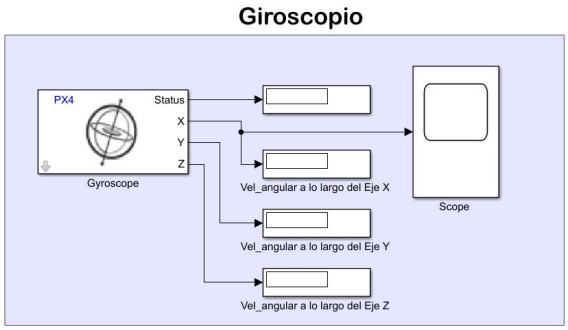


Figura 109: Bloques Simulink ejemplo 3: Bloque sensor giroscopio del PX4.

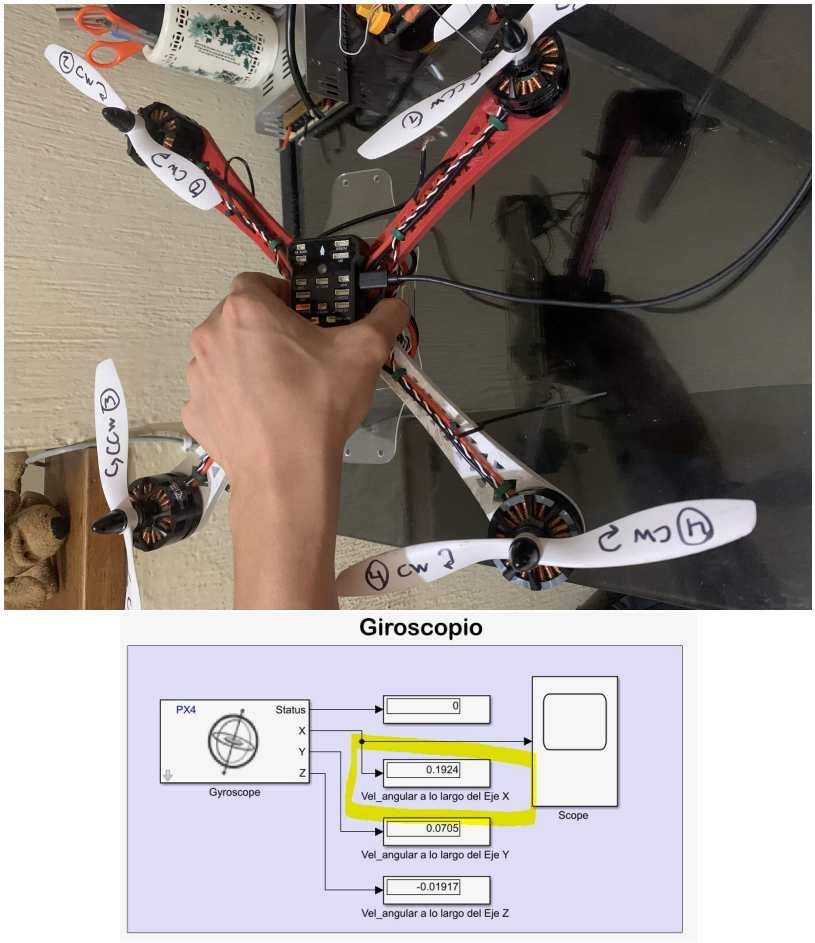


Figura 110: Ejemplo 3: Giro alrededor de Eje x *roll* hacia la derecha.

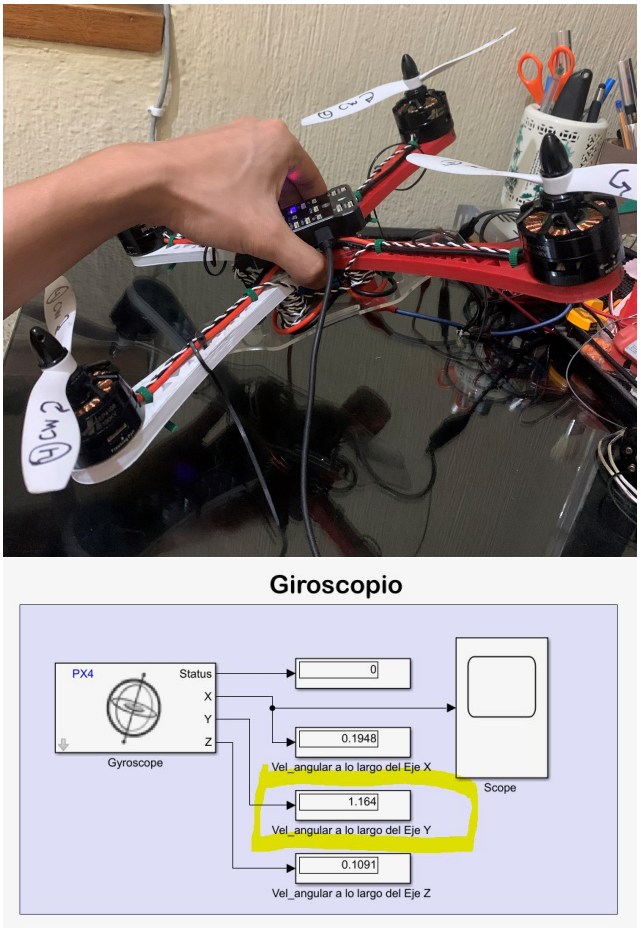


Figura 111: Ejemplo 3: Giro alrededor de eje *y pitch* hacia arriba (brazos 1 y 2 rojos).

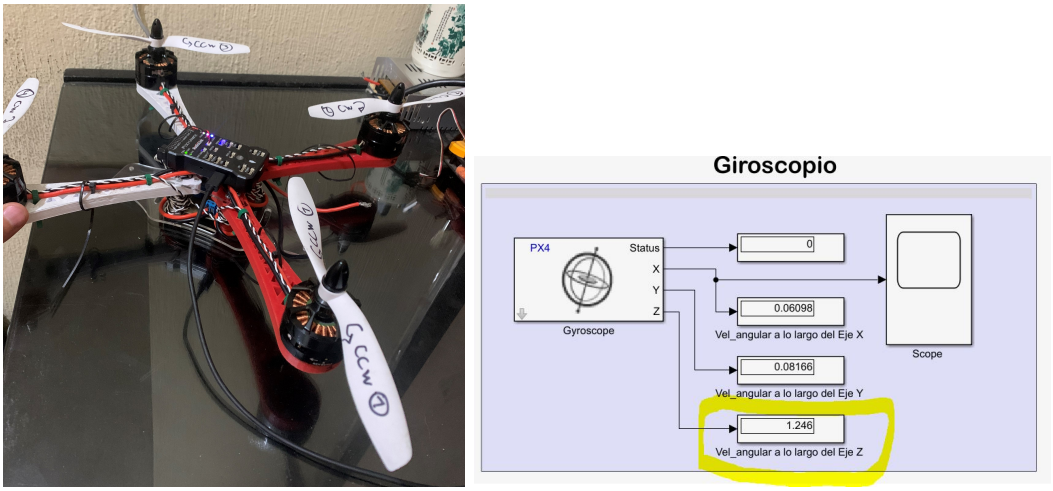


Figura 112: Ejemplo 3: Giro alrededor de eje *z "yaw"* giro CW (visto en planta).

Con los resultados obtenidos de este último ejemplo, se pudo deducir los ejes de referencia locales del controlador de vuelo.

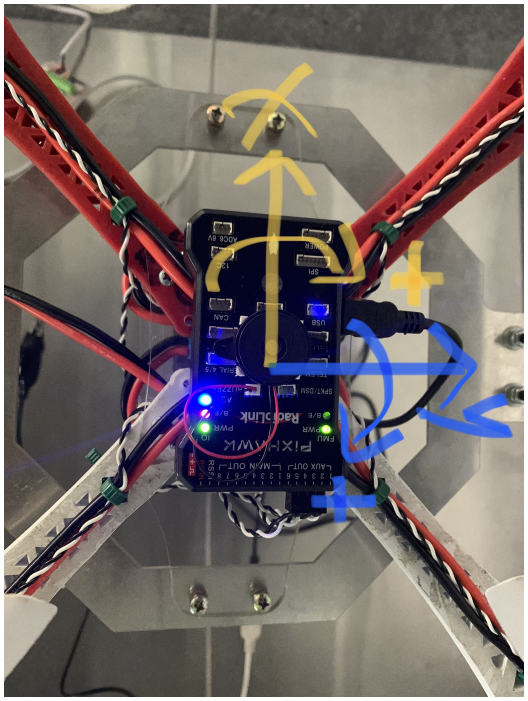


Figura 113: Ejes de referencia locales (roll y pitch), Pixhawk 1.

El último eje local que corresponde al eje y del controlador de vuelo es el eje del *yaw*. Dicho eje positivo apunta hacia el piso como se puede ver en la Figura 114 (a) y en la (b) se muestran los ejes según Simulink, como vemos el eje N, E y D corresponden a los ejes x, y y z respectivamente.

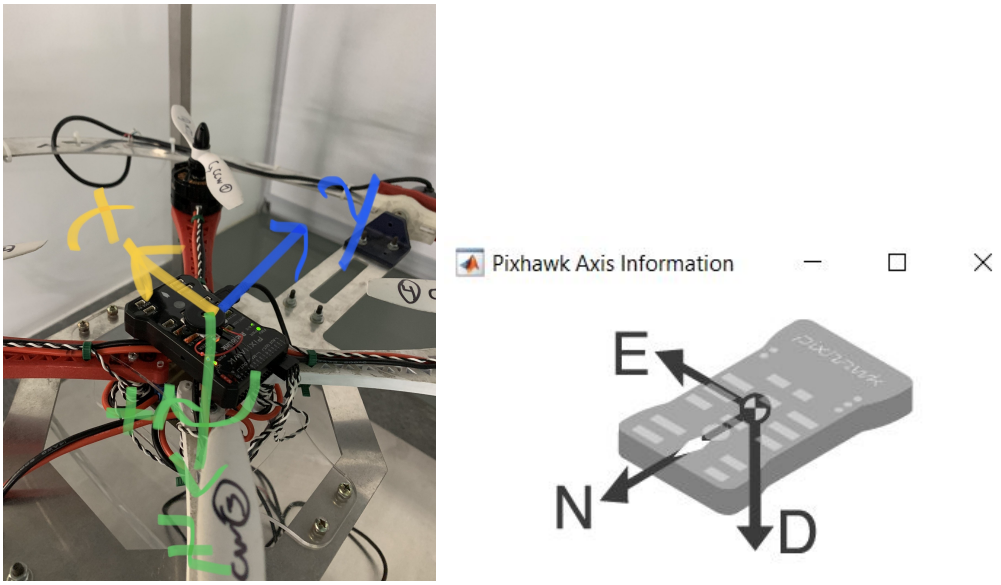


Figura 114: Ejes de referencia locales, Pixhawk 1.

8.2. Validando dron dentro de la plataforma

En esta sección del capítulo se validará por medio de los encoders de la plataforma que el dron, con todos sus elementos físicos (actuadores, marco y pcb), puede girar sin ningún problema sin la ayuda de fuerzas externas para que este se equilibre en lazo abierto.

Para verificar esto utilizó el ejemplo de prueba para probar los motores con la señal que genera el bloque de salida PWM. Dicho diagrama de bloques se muestra en la Figura 108. Cabe mencionar que por motivos prácticos se utilizará la lectura de un enconder, este corresponde al del movimiento de **cabeceo**, giro al rededor del eje y. Por lo tanto los motores que se deben controlar a la misma vez son los dos pares de enfrente motores 1 y 2 (brazos rojos) y los otros dos pares de atrás motores 3 y 4 (pares blancos).

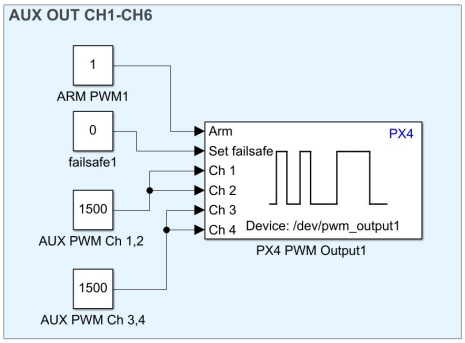


Figura 115: Diagrama de bloque de PWM, movimiento A cabeceo.

Cabe mencionar que las pruebas se realizaron variando el valor de PWM primero para los dos pares de motores de enfrente (1 y 2) los cuales corresponden a los brazos rojos y dejando fija la señal de PWM de 1500 de los dos pares de motores de atrás (3 y 4) los cuales son los montados en los brazos blancos. Dicho de otra manera se variaron los canales CH1 y CH2 de 1500, 1400 y 1300 dejando siempre fijo el CH3 y CH4 con 1500 para la señal de PWM. Y luego se realizó lo mismo para hacer girar al dron en la otra dirección de cabeceo variando los de atrás y dejando fijo los dos de adelante.

En cuanto al diagrama de bloques de Simulink para la lectura de los encoders se realizó de la siguiente manera, como se observa en la Figura 116.

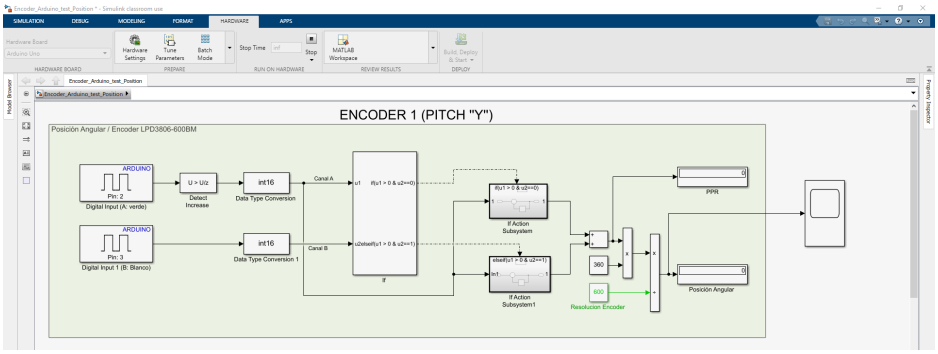


Figura 116: Diagrama de bloques, lectura encoder Arduino Simulink.

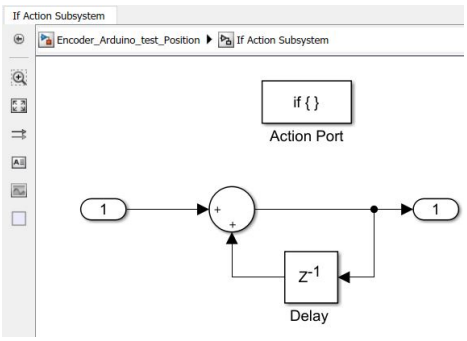


Figura 117: Subsistema 1 para delay, lectura encoder Arduino Simulink.

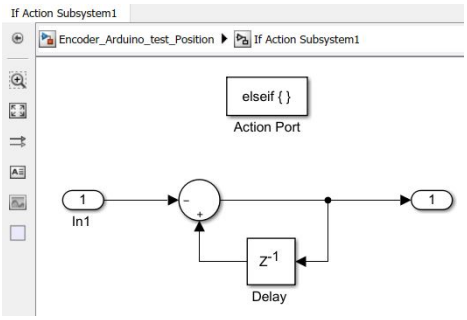


Figura 118: Subsistema 2 para delay, lectura encoder Arduino Simulink.

Explicación bloques lectura encoder

Para la lectura en grados del encoder se analizará el diagrama de bloques de la Figura 116. El primer bloque que se visualiza en dicha figura, es el bloque de la entrada digital (*Digital Input*) para la lectura de los canales A y B respectivamente. Ya que la señal del canal A precede a la del B, la del A será la que manda, como se observa en la Figura 25, por lo tanto únicamente este canal posee en la siguiente línea el bloque de detectar flanco (*Detect Increase*). Luego para ambos canales se hace una conversión de dato `int16` para poder leerlos correctamente en el bloque del `If`, en dicho bloque se comparan el estado del canal B cuando el canal A está en alto. Entonces si A está en HIGH y B está en LOW el giro es *CW*, pero si A y B están en HIGH el giro será *CCW*. El subsistema 1 de la Figura 117 corresponde al *delay* necesario para contabilizar los pulsos del giro a favor de las agujas del reloj (CW) de igual manera para el subsistema 2 (Figura 118) pero para giro (CCW). Luego las dos salidas de estos dos subsistemas se combinan con el bloque de suma y por último se les multiplica el factor de la resolución de nuestro codificador por grados (600 pulsos ya que solo se puede utilizar un canal a la vez) $\frac{1Rev}{600Pulsos} \times 360^o$ para obtener finalmente la salida de posición en grados.

Comportamiento dron - cabeceo A

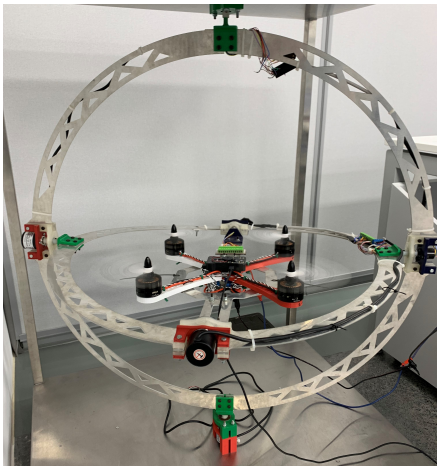


Figura 119: Movimiento cabeceo A - CH1, CH2: 1500 / CH3, CH4: 1500.

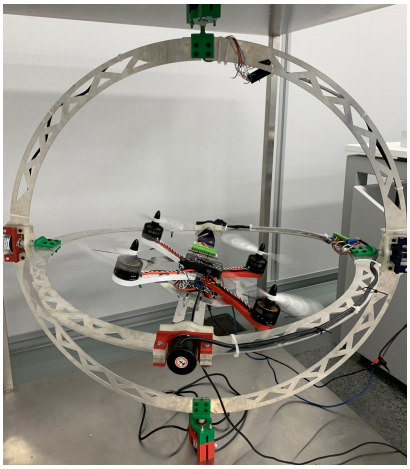


Figura 120: Movimiento cabeceo A - CH1, CH2: 1400 / CH3, CH4: 1500.

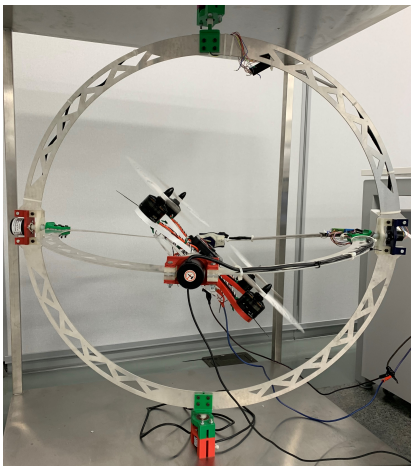


Figura 121: Movimiento cabeceo A - CH1, CH2: 1300 / CH3, CH4: 1500.

Resultados cabeceo A: Lectura Encoder

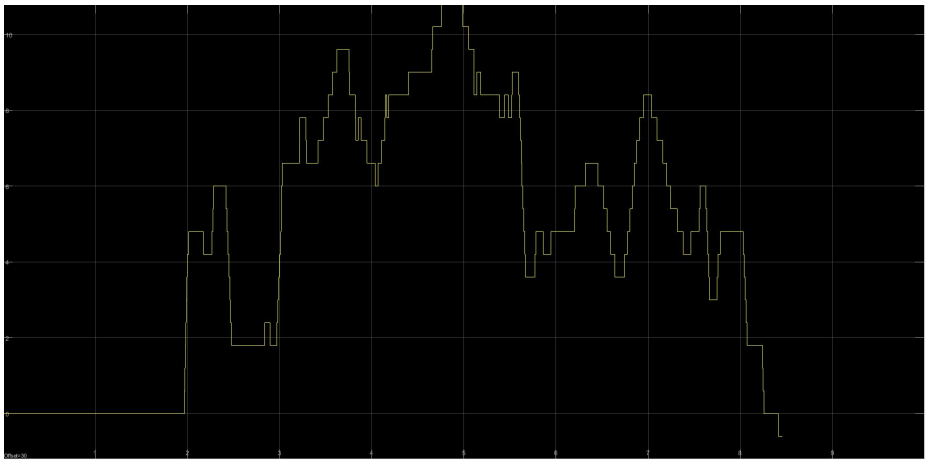


Figura 122: Encoder cabeceo A (Arranque motores) - CH1, CH2: 1500 / CH3, CH4: 1500

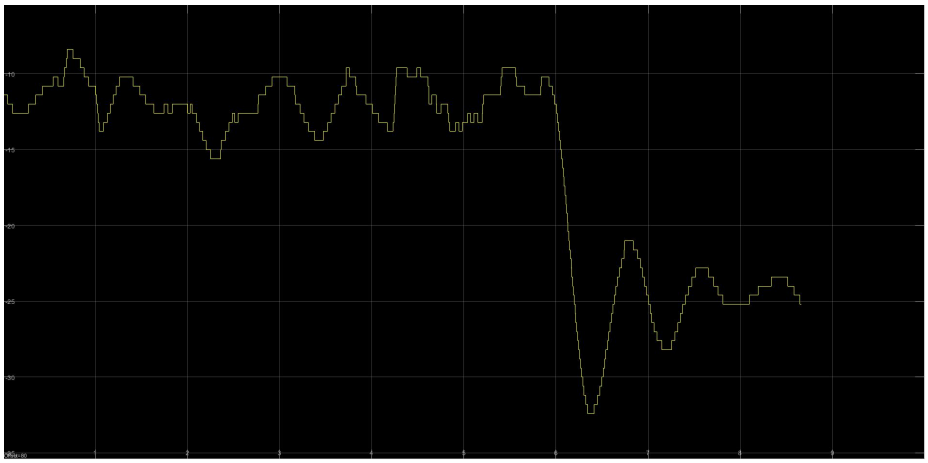


Figura 123: Encoder cabeceo A - CH1, CH2: 1400 / CH3, CH4: 1500.

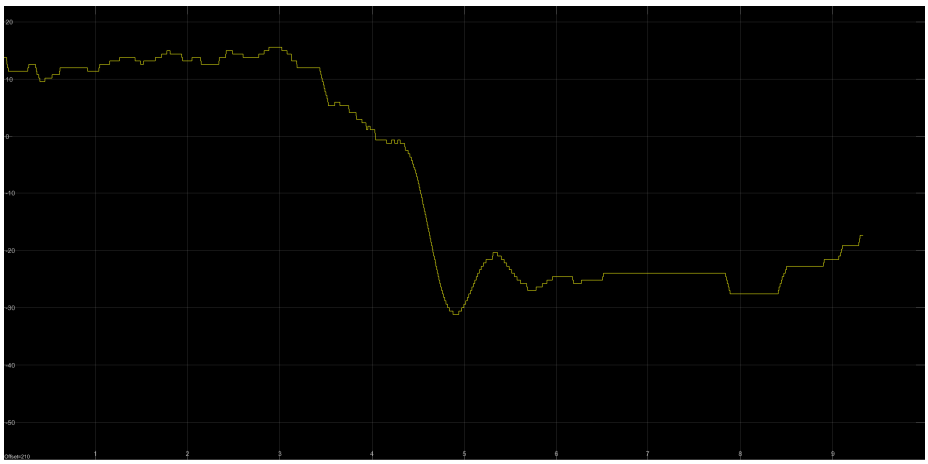


Figura 124: Encoder cabeceo A - CH1, CH2: 1300 / CH3, CH4: 1500.

Comportamiento dron - cabeceo B

Figura 125: Movimiento cabeceo B - CH1, CH2: 1500 / CH3, CH4: 1400.

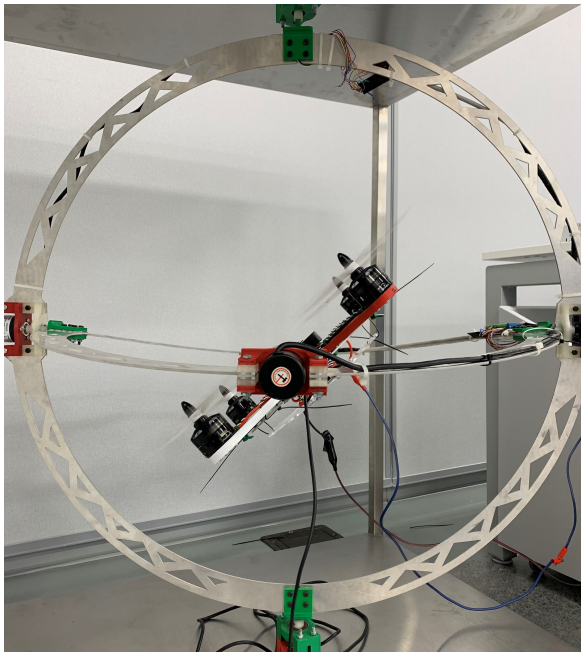


Figura 126: Movimiento cabeceo B - CH1, CH2: 1500 / CH3, CH4: 1300.

Resultados cabeceo B: Lectura Encoder

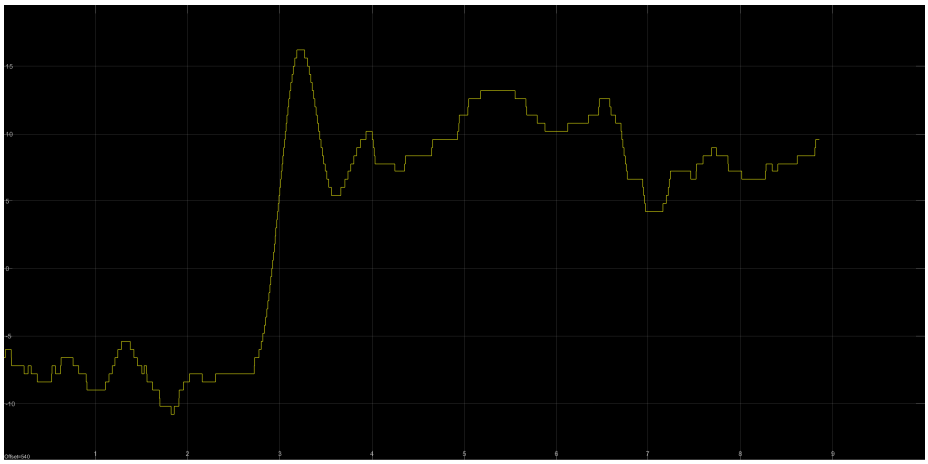


Figura 127: Encoder cabeceo B - CH1, CH2: 1500 / CH3, CH4: 1400.

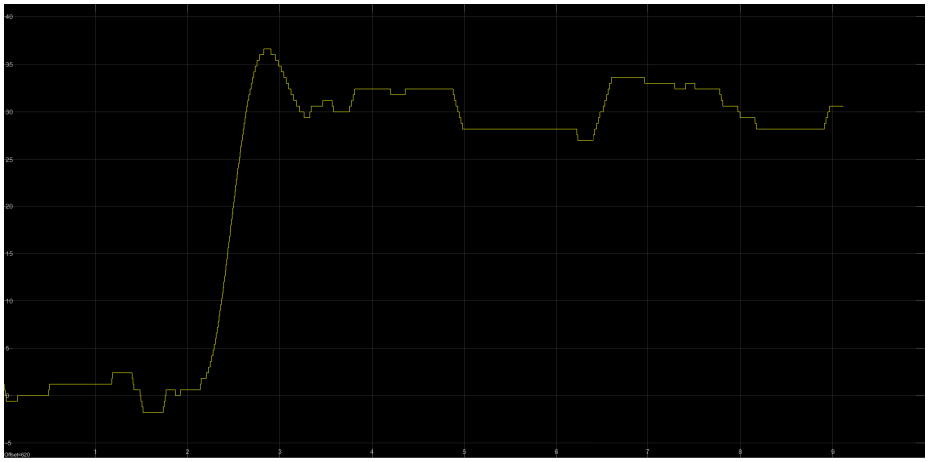


Figura 128: Encoder cabeceo B - CH1, CH2: 1500 / CH3, CH4: 1300.

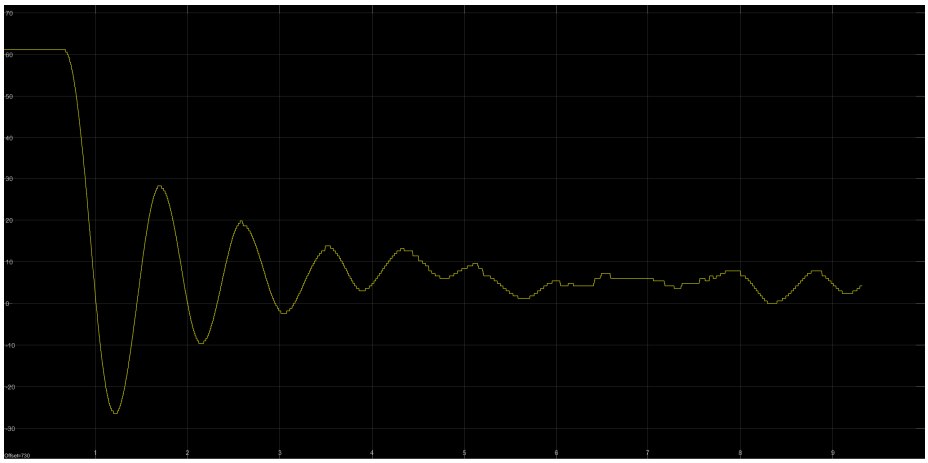


Figura 129: Encoder cabeceo B - CH1, CH2: 1500 / CH3, CH4: 1500.

8.3. Lectura de los codificadores

8.3.1. Código de arduino

Figura 130: Código Arduino Uno - lectura de los codificadores rotacionales [40].

8.3.2. Codigo Python y plot de matlab

C:\Users\josue\Desktop\Serial2Excel_Encoder\Serial2Excel_EncoderTesis.py

Serial2Excel_EncoderTesis.py

```python
import serial

arduino_port = "COM6" #serial port of Arduino
baud = 9600 #arduino uno runs at 9600 baud
fileName="data3_4.csv" #name of the CSV file generated
samples = 1000 #how many samples to collect
print_labels = False

ser = serial.Serial('COM6', baud)
#print("Connected to Arduino port:" + arduino_port)
file = open(fileName, "w")
#print("Created file")

line = 0 #start at 0 because our header is 0 (not real data)
while line <= samples-1:

    if print_labels:
        if line==0:
            print("Printing Column Headers")
        else:
            print("Line " + str(line) + ": writing...")

    getData=str(ser.readline().decode('ascii'))
    data=getData[0:][:-2] #quita basura
    print(data)

    file.write(data + ",") #write data with a newline
    line = line+1

print("Data collection complete!")
file.close()
```

Figura 131: Código Python-`Serial2Excel` - toma de datos de los codificadores rotacionales.

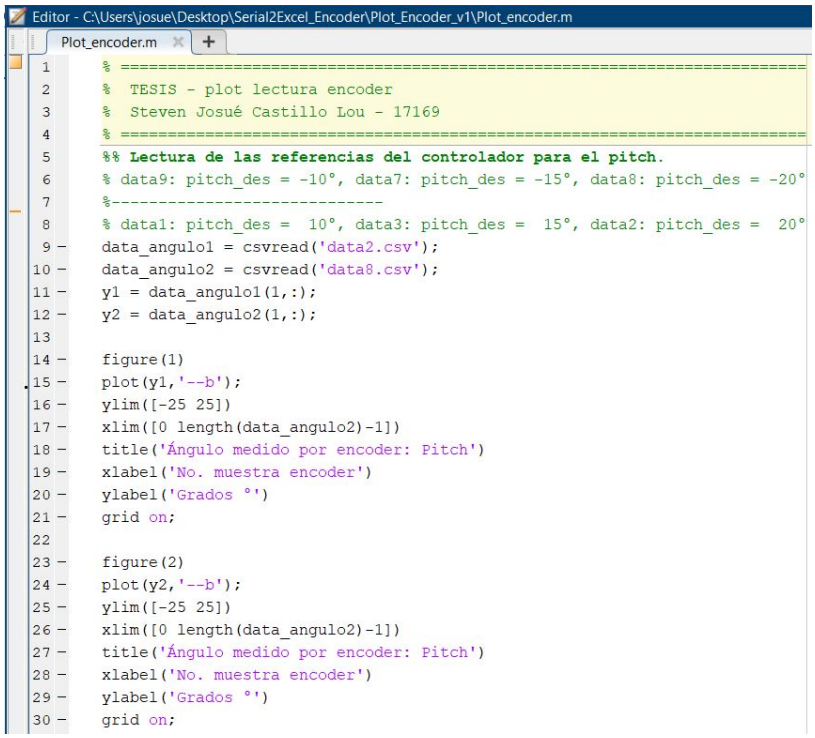


Figura 132: Código Matlab-`Plot_encoder` de la toma de datos de los codificadores rotacionales.

Con los tres códigos anteriores se generaron las gráficas finales para la validación del control de orientación del cuadricóptero ensamblado en la plataforma. El primero de ellos `Encoder_UNO_Tesis_DRON.ino`, básicamente tiene la función de leer los datos del codificador rotacional ensamblado en la plataforma para los tres ángulos de Euler (roll, pitch y yaw respectivamente), esto se logró con la programación del Arduino Uno mostrado en la Figura 130, luego este manda los datos leídos por el puerto serial de la computadora.

El código de Python `Serial2Excel_EncoderTesis.py`, de la Figura 132 se utilizó para leer los datos recibidos en el puerto serial que manda el Arduino, para que luego de 1000 muestras se logré almacenar la corrida de las distintas referencias de los ángulos del controlador de vuelo para llegar al ángulo deseado, esto se almancena de manera eficiente en un archivo de Excel `data#.csv`. Por último la data guardada en dichos archivos de Excel fue importada en un script de Matlab `Plot_encoder.m`, para luego generar los respectivos gráficos como se observan en las figuras 150, 151 y 152 de la sección 9.0.4.

Enlace video: Tesis 2021- Lectura encoders rotacionales.

El procedimiento de la lectura, almacenamiento y la obtención de las gráficas generados con los tres códigos anteriores se puede observar en el siguiente video. En dicho video se relata con un mayor detalle lo que realiza cada uno de estos tres códigos.

- `https://youtu.be/y6aAyKdTmF8`

CAPÍTULO 9

Control de orientación implementado

En este último capítulo se presenta la derivación del control de orientación para la configuración del cuadricóptero diseñado en capítulos anteriores, el cual corresponde a la configuración en equis del mismo.

Esta derivación se basó en dos referencias principales, las cuales corresponden a dos grandes universidades del extrajero sumamente avanzadas en proyectos de esta índole. Dichas referencias de estas universidades corresponden a la Universidad de Pennsylvania [41] y al Instituto Federal de Tecnología de Zurich (en alemán Eidgenössische Technische Hochschule Zürich) [42].

Cabe mencionar que la referencia que se utilizó de manera principal fue la de la Universidad de Zurich ya que esta utiliza la misma convención de ejes que posee el controlador de vuelo de PX4 (Pixhawk 1). Dichos ejes se muestran a continuación en la Figura 133.

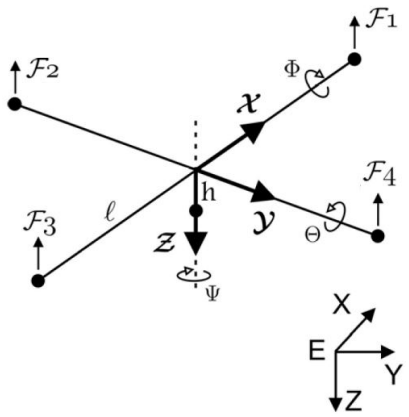


Figura 133: Sistema de coordenadas - cuadricóptero en configuración + [42].

Como vemos los tres ejes principales(xyz) de la Figura 133 coinciden en la misma dirección que el sistema de coordenadas {NED} de el controlador de vuelo utilizado de la Figura 114 con la pequeña diferencia que utilizan la configuración en cruz en vez de la configuración en equis para el cuadricóptero. Por lo tanto esa transformación de rotación del marco de referencia en cruz a equis será la que se tiene que ajustar.

Por último cabe mencionar que este modelo dinámico para el cuadricóptero también está basado en la formulación de Newton-Euler definido en la sección 6.2.3 del marco teórico y que se utilizó las herramientas de matemática simbólica de Matlab para derivar y simplificar las expresiones del control de orientación de este modelo.

9.0.1. Definición de los marcos de referencia

Teniendo en cuenta las diferencias de los sistemas de coordenadas entre ambas configuraciones del cuadricóptero se define el siguiente sistema de coordenadas que relaciona los 3 marcos de referencia.

1. Marco de referencia global de la tierra, Marco de referencia del inercial: $\{E\}$
2. Dron Configuración en + (NED-XYZ), Marco de referencia: $\{O\}$
3. Dron Configuración en X (NED-XYZ), Marco de referencia del $body frame$: $\{B\}$

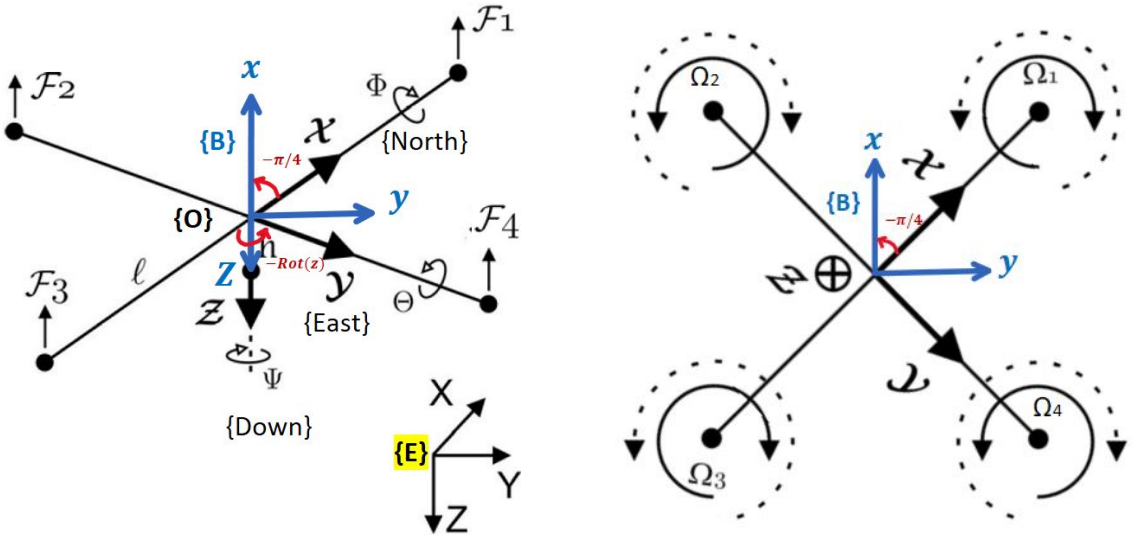


Figura 134: Marcos de referencia de interés (vista isométrica) / (vista en planta).

9.0.2. Derivación del control de orientación

Recordando el vector de las variables de estado y el vector de las entradas que representan las fuerzas de empuje de cada motor.

$$x = \begin{bmatrix} \boldsymbol{x}[\mathrm{m}] \\ \boldsymbol{y}[\mathrm{m}] \\ \boldsymbol{z}[\mathrm{m}] \\ \emptyset_{\mathrm{roll}}\ [\mathrm{rad}] \\ \boldsymbol{\theta}_{\mathrm{pitch}}\ [\mathrm{rad}] \\ \boldsymbol{\psi}_{yaw}[\mathrm{rad}] \\ \boldsymbol{dx}[\mathrm{m/s}] \\ \boldsymbol{dy}[\mathrm{m/s}] \\ \boldsymbol{dz}[\mathrm{m/s}] \\ \boldsymbol{d\emptyset}[\mathrm{rad}/s] \\ \boldsymbol{d\theta}[\mathrm{rad}/s] \\ \boldsymbol{d\psi}[\mathrm{rad}/s] \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \emptyset_{\mathrm{roll}} \\ \theta_{\mathrm{pitch}} \\ \psi_{yaw} \\ dx \\ dy \\ dz \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix}, u = \begin{bmatrix} k_{f1}\left(\omega_{M1}\right)^2 \\ k_{f2}\left(\omega_{M2}\right)^2 \\ k_{f3}\left(\omega_{M3}\right)^2 \\ k_{f4}\left(\omega_{M4}\right)^2 \end{bmatrix} \tag{56}$$

El campo vectorial de la dinámica del sistema para el control, se representa por medio de la siguiente ecuación.

$$\dot{x} = f(x, u) \tag{57}$$

Por lo tanto derivando el vector de estados para encontrar $f(x, u)$ tenemos lo siguiente:

$$\dot{x} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\emptyset} \\ \dot{\theta} \\ \dot{\psi} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\emptyset} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{x} = x_7 \\ \dot{y} = x_8 \\ \dot{z} = x_9 \\ \dot{\emptyset} = x_{10} \\ \dot{\theta} = x_{11} \\ \psi = x_{12} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\emptyset} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}, \quad \text{Ec. Newton } = \begin{cases} \ddot{x} \\ \ddot{y}\ , \\ \ddot{z} \end{cases} \quad \text{Ec. Euler } = \begin{cases} \ddot{\phi} = \dot{p} \\ \ddot{\theta} = \dot{q} \\ \ddot{\psi} = \dot{r} \end{cases} \tag{58}$$

Teniendo definido lo anterior y sabiendo que las variables del vector de estado derivado faltantes las obtenemos a partir de la ecuación de Newton-Euler 17 las cuales corresponden a las aceleraciones lineales y angulares del cuadricóptero respectivamente, se plantean los siguientes pasos para transformar el modelo de la dinámica de la configuración + a la configuración X del cuadricóptero.

Paso 1: Definir la matriz de inercia diagonal en el marco de referencia $\{O\}$.

$$I_O = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{59}$$

Paso 2: Definir la orientación de $\{B\}$, con respecto de $\{O\}$
(Marco + transformado al marco X).

$${}^{O}R_B = \text{rot}(-\pi/4) = \begin{bmatrix} \cos(-\pi/4) & -\sin(-\pi/4) & 0 \\ \sin(-\pi/4) & \cos(-\pi/4) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{60}$$

Si se quiere regresar del marco X al marco +, se utiliza la rotación opuesta, la cual corresponde a la traspuesta de la rotación anterior.

$${}^{B}R_O = {}^{O}R_B^T = \text{rot}\, z(+\pi/4) = \begin{bmatrix} \cos(-\pi/4) & \sin(-\pi/4) & 0 \\ -\sin(-\pi/4) & \cos(-\pi/4) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{61}$$

Paso 3: Se transforma la inercia al marco del body frame $\{B\}$ para la configuración X.

$$I_B = I_O \times {}^{O}R_B \tag{62}$$

Paso 4: Orientación del marco $\{O\}$ con respecto del inercial fijo $\{E\}$
Convención ZYX.

$${}^{E}R_O = rotz(yaw) \times roty(pitch) \times rotx(roll) \tag{63}$$

donde $rotz(yaw), roty(pitch), rotx(roll)$ son las matrices de rotación definidas en (7), (6), (5), respectivamente. Vemos que esta matriz de rotación del marco $\{E\}$ al $\{O\}$ queda igual a la ec. 10.

```
>> ER_O = rotz(yaw)*roty(pitch)*rotx(roll)

ER_O =

[ cos(pitch)*cos(yaw), cos(yaw)*sin(pitch)*sin(roll) - cos(roll)*sin(yaw), sin(roll)*sin(yaw) + cos(roll)*cos(yaw)*sin(pitch)]
[ cos(pitch)*sin(yaw), cos(roll)*cos(yaw) + sin(pitch)*sin(roll)*sin(yaw), cos(roll)*sin(pitch)*sin(yaw) - cos(yaw)*sin(roll)]
[         -sin(pitch),                              cos(pitch)*sin(roll),                              cos(pitch)*cos(roll)]
```

Figura 135: Matlab - matriz de rotación del marco $\{E\}$ al $\{O\}$.

Vemos que se obtiene lo mismo desde Matlab.

Paso 5: Orientación del body frame $\{B\}$ visto desde el inercial $\{E\}$.

$$^{E}R_{B} = {}^{E}R_{O} \times {}^{O}R_{B} \tag{64}$$

Si se quiere regresar del marco $\{B\}$ al marco $\{E\}$, se utiliza la traspuesta nuevamente:

$$^{B}R_{E} = {}^{E}R_{B}^{T} \tag{65}$$

Paso 6: Jacobiano que transforma las derivadas del roll, pitch y yaw a las velocidades angulares del drone wx, wy y wz en el marco $\{O\} \rightarrow$ configuración $+$.

$$^{O}J_{rpy(3x3)} = \begin{bmatrix} 1 & & \\ 0 & rotx(\phi) \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & roty(\theta)rotx(\phi) \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\ 0 & & \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\text{ pitch }) \\ 0 & \cos(\text{ roll }) & \cos(\text{ pitch })\sin(\text{ roll }) \\ 0 & -\sin(\text{ roll }) & \cos(\text{ pitch })\cos(\text{ roll }) \end{bmatrix} \tag{66}$$

Paso 7: Jacobiano que transforma las derivadas del roll, pitch y yaw a las velocidades angulares del drone wx, wy y wz en el marco del body frame $\{B\} \rightarrow$ configuración X.

$$^{B}J_{rpy(3x3)} = {}^{B}R_{O} \times {}^{O}J_{rpy} \tag{67}$$

Paso 8: Recordando el sistema de ecuaciones de Newton-Euler de la ec. 17, si se analiza por separado ambas ecuaciones tenemos lo siguiente.

- Newton

$$\begin{gathered} mI_{(3x3)}\dot{V} = \sum_{i}^{4} {}^{B}F_{i} + {}^{B}F_{gravedad} \rightarrow ma_{COM} = \sum_{i}^{4} {}^{B}F_{i} + {}^{B}F_{gravedad} \\ r_{E} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow m\ddot{r}_{E} = \sum_{i}^{4} {}^{B}F_{i} + {}^{B}F_{\text{gravedad}} \end{gathered} \tag{68}$$

Trasladando las fuerzas de los 4 motores al marco de referencia B con la rotación de la ec. 64 y despejando para el vector de aceleraciones lineales $\ddot{r_{E}}$ tenemos

$$\begin{gathered} m\ddot{r_{E}} = {}^{E}R_{B} \times ({}^{B}F_{1} + {}^{B}F_{2} + {}^{B}F_{3} + {}^{B}F_{4}) + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \\ \ddot{r_{E}} = (1/m) \times {}^{E}R_{B} \times \begin{bmatrix} 0 \\ 0 \\ -({}^{B}F_{1} + {}^{B}F_{2} + {}^{B}F_{3} + {}^{B}F_{4}) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \end{gathered} \tag{69}$$

- Euler

Definiendo que la velocidad angular del drone con respecto del inercial $\{E\}$, pero visto desde el body frame $\{B\}$ corresponde a ${}^{B}\omega_{EB} = \begin{bmatrix} wx & wy & wz \end{bmatrix}^T$ (estas velocidades angulares son las medidas por el giroscopio abordo del Pixhawk).

$$
\begin{aligned}
I\dot{\omega} + \omega \times I\omega &= \sum_i M_i \\
(I_B)^B\dot{\omega_{EB}} + {}^{B}\omega_{EB} \times (I_B)^B\omega_{EB} &= \sum_i M_i
\end{aligned}
\tag{70}
$$

Definiendo los vectores de fuerzas generados por los motores, vistos en el body frame.

$$
\begin{aligned}
{}^{B}F_i &= \sum_i^4 \begin{bmatrix} 0 \\ 0 \\ -f_i \end{bmatrix} \\
f_i &= kf(\omega_i)^2 = b(\omega_i)^2
\end{aligned}
\tag{71}
$$

Definiendo los brazos de palanca para cada fuerza generada por los motores, expresados en el body frame.

$$
\begin{aligned}
{}^{B}L_1 &= \begin{bmatrix} +Lcos(\pi/4) \\ +Lsin(\pi/4) \\ 0 \end{bmatrix} \\
{}^{B}L_2 &= \begin{bmatrix} +Lcos(\pi/4) \\ -Lsin(\pi/4) \\ 0 \end{bmatrix} \\
{}^{B}L_3 &= \begin{bmatrix} -Lcos(\pi/4) \\ -Lsin(\pi/4) \\ 0 \end{bmatrix} \\
{}^{B}L_4 &= \begin{bmatrix} -Lcos(\pi/4) \\ +Lsin(\pi/4) \\ 0 \end{bmatrix}
\end{aligned}
\tag{72}
$$

Los vectores de momento generados por los motores, vistos en el body frame son:

$$
\begin{aligned}
M_1 &= ({}^{B}F_1 \times {}^{B}L_1) + \begin{bmatrix} 0 & 0 & -\tau_i \end{bmatrix}^T \\
M_2 &= ({}^{B}F_2 \times {}^{B}L_2) + \begin{bmatrix} 0 & 0 & +\tau_i \end{bmatrix}^T \\
M_3 &= ({}^{B}F_3 \times {}^{B}L_3) + \begin{bmatrix} 0 & 0 & -\tau_i \end{bmatrix}^T \\
M_4 &= ({}^{B}F_4 \times {}^{B}L_4) + \begin{bmatrix} 0 & 0 & +\tau_i \end{bmatrix}^T \\
\tau_i &= k_M(\omega_i)^2 = d(\omega_i)^2
\end{aligned}
\tag{73}
$$

Constantes para el cuadaricóptero		
Nombre	Símbolo	Valor [unidades]
Momento de inercia del cuerpo respecto al eje x	Ixx	0.000772508 [kg m^2]
Momento de inercia del cuerpo respecto al eje y	Iyy	0.000772508 [kg m^2]
Momento de inercia del cuerpo respecto al eje z	Izz	0.001701286 [kg m^2]
Largo brazos desde el COM al eje del motor	L	0.1830 [m]
Gravedad	g	9.81 [m/s^2]
Masa del cuadricóptero en conjunto	m	1.0560 [kg]
Resistencia del motor	R	0.2040 [ohms]
Constante del motor	Km	0.0542 [Nm/A]
Constante del par del motor	Kt	0.0245 [Nm/A]
Constante de empue del motor (estimada)	Kf = b	1
Constante de arrastre del motor (estimada)	Ktau = d	Km

Cuadro 6: Valores de constantes para el modelo del cuadricóptero X.

Cabe mencionar que las constantes de empuje (b) y de arrastre (d) fueron consideradas dentro de tuneo de las constantes del controlador PD. Finalmente despejando de la ecuación 70 la aceleración angular ${}^{B}\dot{\omega_{EB}}$ temenos lo siguiente.

$$ {}^{B}\dot{\omega_{EB(3x3)}} = (I_B^{-1}) \times (M_1 + M_2 + M_3 + M_4 - ({}^{B}\omega_{EB} \times (I_B){}^{B}\omega_{EB})) \qquad (74) $$

Evaluando las ecuaciones despejadas de Newton (69) y de Euler (74) con las constantes del cuadro 6 obtennos las siguientes aceleraciones lineales y angulares respectivamente con matlab.

```
>> Newton = (1/m)*ER_B*(BF_1+BF_2+BF_3+BF_4) + [0;0; -g]

Newton =

 -((125*sin(roll)*sin(yaw))/132 + (125*cos(roll)*cos(yaw)*sin(pitch))/132)*(wM1^2 + wM2^2 + wM3^2 + wM4^2)
   ((125*cos(yaw)*sin(roll))/132 - (125*cos(roll)*sin(pitch)*sin(yaw))/132)*(wM1^2 + wM2^2 + wM3^2 + wM4^2)
                                - (125*cos(pitch)*cos(roll)*(wM1^2 + wM2^2 + wM3^2 + wM4^2))/132 - 981/100
```

Figura 136: Ec. Newton: Aceleraciones lineales x, y, z

```
>> pretty(Euler)
-- --                                      2                            2
|  |  1202634141977023517118561147268940 8 wM1       96276732420409088 wM2        790578081719625214433899294 25232 wx wz
```

```
>> pretty(Euler)
-- --                                    2                           2
|  |  12026341419770235171185611472689408 wM1      96276732420409088 wM2      7905780817196252144338992 9425232 wx wz
|  |  ----------------------------------------- + -------------------------------- - ---------------------------------------
-- --     50767458784174125347918120339665        50767458784174125347918120339665     50767458784174125347918120339665

                                                                                2                            2     --
   2829034938778841178706552817 8081 wy wz   12026341419770235171185611472689408 wM3        96276732420409088 wM4      |
 - --------------------------------------- - ---------------------------------------- - -------------------------------- |,
      50767458784174125347918120339665          50767458784174125347918120339665       50767458784174125347918120339665 --

  --                          2                                             2
  |     2483295394456308224 wM3       12026341419770235171185611472689664 wM4    28290349387788411787065528178081 wx wz
  | -------------------------------- + ---------------------------------------- + ---------------------------------------
  -- 50767458784174125347918120339665     50767458784174125347918120339665          50767458784174125347918120339665

                                                               2                                              2 --
   79057808171962521443389929425232 wy wz      2483295394456308224 wM1      12026341419770235171185611472689664 wM2  |
 - --------------------------------------- - -------------------------------- - ---------------------------------------- |,
      50767458784174125347918120339665      50767458784174125347918120339665     50767458784174125347918120339665      --

  --                      2                        2                       2                      2                        2                         2 -- --
  | 250006570386763712 wM2    250006570386763712 wM4    2519113405153044 wx    2519113405153044 wy    250006570386763712 wM1    250006570386763712 wM3  |  |
  | ----------------------- + ----------------------- + -------------------- + -------------------- - ----------------------- - -----------------------  |  |
  --    7845796859546257          7845796859546257        7845796859546257       7845796859546257        7845796859546257          7845796859546257     -- --
```

Figura 137: Ec. Euler: Aceleraciones angulares dwx, dwy, dwz

Entradas virtuales de control

Ya que únicamente interesa controlar la orientación del cuadricóptero, podemos utilizar la ecuación de Euler 74 para definir las variables virtuales del control de actitud para establecerlas en el controlador de vuelo (Pixhawk 1).

A continuación se muestra un diagrama de bloques que representa los lazos de control anidados de la posición y de la actitud que corresponde a la orientación del multirrotor.

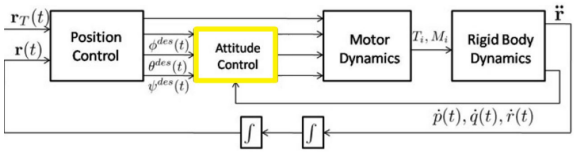


Figura 138: Lazos de control anidados para el control de posición y actitud [41].

Cabe mencionar que cualquier cuadricóptero se caracteriza por ser un robot de base flotante el cual no tiene ningún contacto que lo ancle al suelo (ya que se mueven en el aire libremente), por ende estos robots no tienen ningún actuador que modifiquen su posición directamente para las coordenadas x, y. Es por ello que en la Figura 138 vemos que el bloque de control de posición precede al control de orientación, ya que el primero depende del segundo. Dicho de otra manera no se puede controlar la posición del dron directamente en x, y sin antes controlar su orientación en los tres ángulos de Euler ϕ, θ, ψ.

Lo mencionado en el párrafo anterior es de suma importancia, ya que la no dependencia de la orientación con respecto de la posición es lo que permite desacoplar ambos lazos de control por separado. Si la orientación dependiera de la posición, no se podría plantear un control de actitud por separado e independiente al de la posición. Por lo tanto con el control adecuado de los tres ángulos de Euler además de poder establecer a que altitud se querrá colocar el cuadricóptero para volar, podemos tener un control total del dron en las coordenadas x, y, z.

A continuación se definirán las cuatro entradas virtuales según la contribución de cada uno de los cuatro motores, según la contribución a los cuatro movimientos básicos para controlar el dron, los cuales son el *thrust, roll, pitch, yaw*.

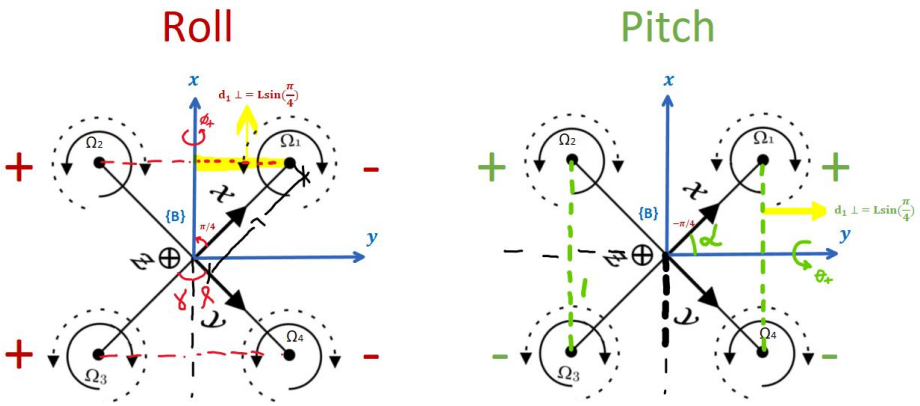


Figura 139: Contribución motores a movimientos en el *roll*, *pitch* respectivamente.

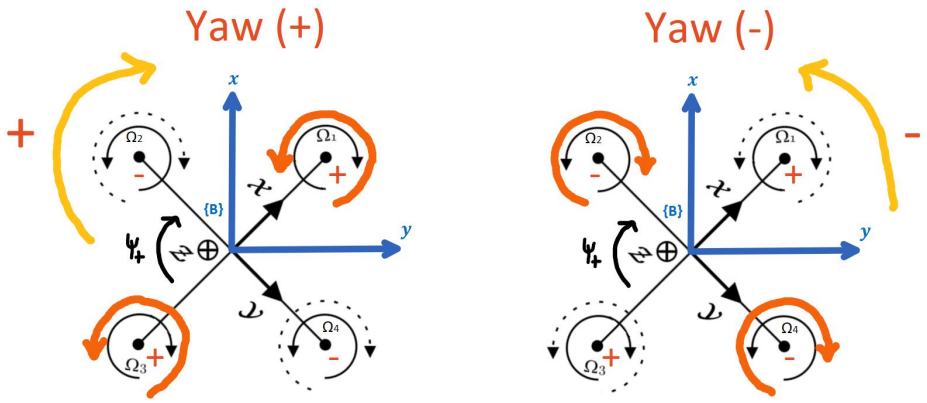


Figura 140: Contribución motores al movimiento en el *yaw* positivo y negativo respectivamente.

Con los diagramas de los movimientos para las tres maniobras básicas en la orientación mostradas en las figuras 139 y 140 se puede plantear la contribución de cada motor para cada uno de los tres grados de libertad.

Finalmente podemos establecer las cuatro entradas de contribución de los motores (en la configuración del dron X) para cada uno de los cuatro movimientos básicos del cuadricóptero siendo U_1, U_2, U_3, U_4 para las maniobras del *thrust, roll, pitch, yaw* respectivamente.

$$U = \begin{bmatrix} U1_{Fz} \\ U2_{\phi} \\ U3_{\theta} \\ U4_{\psi} \end{bmatrix} = \begin{bmatrix} b(+w_1^2 + w_2^2 + w_3^2 + w_4^2), \\ bLsin(pi/4) * (-w_1^2 + w_2^2 + w_3^2 - w_4^2), \\ bLsin(pi/4) * (+w_1^2 + w_2^2 - w_3^2 - w_4^2), \\ d(+w_1^2 - w_2^2 + w_3^2 - w_4^2) \end{bmatrix} \tag{75}$$

La ecuación anterior también se puede expresar de la manera $U = K\omega_i$,

$$\begin{bmatrix} U1_{Fz} \\ U2_{\phi} \\ U3_{\theta} \\ U4_{\psi} \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -bLsin(pi/4) & +bLsin(pi/4) & +bLsin(pi/4) & -bLsin(pi/4) \\ +bLsin(pi/4) & +bLsin(pi/4) & -bLsin(pi/4) & -bLsin(pi/4) \\ d & d & d & d \end{bmatrix} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} \tag{76}$$

Expandiendo la ecuación de Euler y sustituyendo los momentos (U2,U3,U4) generados por cada motor de la ecuación 75 en 70 tenemos lo siguiente.

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} U2_{\phi} \\ U3_{\theta} \\ U4_{\psi} \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{77}$$

$$\omega \times I\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} qr\,(I_{zz} - I_{yy}) \\ rp\,(I_{xx} - I_{zz}) \\ pq\,(I_{yy} - I_{xx}) \end{bmatrix} \tag{78}$$

La resta de las inercias de el último elemento del producto cruz de $\omega \times I\omega$, es cero debido a la simetría del marco del dron ($I_{xx} \approx I_{yy}$).

Por lo tanto sustituyendo valores, la ecuación (76) queda de la siguiente manera.

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} U2_\phi \\ U3_\theta \\ U4_\psi \end{bmatrix} - \begin{bmatrix} qr\left(I_{zz} - I_{yy}\right) \\ rp\left(I_{xx} - I_{zz}\right) \\ 0 \end{bmatrix} \tag{79}$$

Separando cada elemento de la ecuación 79 y reordenando, la expresión anterior nos queda de la siguiente manera.

$$\begin{aligned} I_{xx}\dot{p} &= qr\left(I_{yy} - I_{zz}\right) + U2_\phi \\ I_{yy}\dot{q} &= pr\left(I_{zz} - I_{xx}\right) + U3_\theta \\ I_{zz}\dot{r} &= U4_\psi \end{aligned} \tag{80}$$

Al linealizar la actitud de la expresión anterior cuando el dron está suspendido en el aire (flotando) en el punto de equilibrio: $\phi = \theta = \psi = p = q = r = 0$ obtenemos lo siguiente

$$\begin{aligned} \ddot{\phi} &= \frac{1}{I_{xx}} U2_\phi \\ \ddot{\theta} &= \frac{1}{I_{yy}} U3_\theta \\ \ddot{\psi} &= \frac{1}{I_{zz}} U4_\psi \end{aligned} \tag{81}$$

Con el modelo anterior de la ecuación 81 podemos plantear un controlador PD de manera individual para cada maniobra (las 3 son independientes entre sí) en la actitud del cuadricóptero.

$$\begin{aligned} U_1 &= T_{\text{des}} \\ U_2 &= \left(\phi_{\text{des}} - \phi\right) k_{p\ \text{Roll}} - \dot{\phi} k_{d\ \text{Roll}} \\ U_3 &= \left(\theta_{\text{des}} - \theta\right) k_{p\ \text{Pitch}} - \dot{\theta} k_{d\ \text{ditch}} \\ U_4 &= \left(\psi_{\text{des}} - \psi\right) k_{p\ \text{Yaw}} - \dot{\psi} k_{d\ \text{Yaw}} \end{aligned} \tag{82}$$

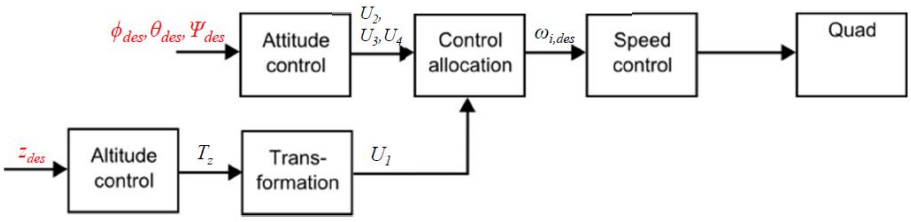


Figura 141: Control de actitud desacoplado [42].

9.0.3. Implementación en el controlador vuelo de PX4 con Matlab/Simulink

Paso 1: Obtener roll, pitch y yaw de la IMU.

La lectura de los tres ángulos de vuelo se leen directamente con el bloque del paquete de soporte de PX4 "***Vehicle Attitude***".

Dicho bloque otorga las lecturas de los ángulos en forma de cuaternión. Luego se transforma al tipo de dato que se necesita y por último se transforma de cuaternión a ángulos de euler con el bloque de "***Quaternions to Rotation Angles***".

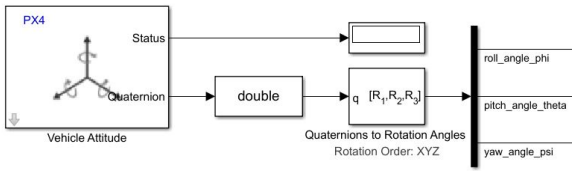


Figura 142: Ángulos de Euler medidos desde la IMU del Pixhawk1.

Paso 2: Obtener las derivadas del roll, pitch y yaw

Estas derivadas de los ángulos de Euler se obtienen a partir de las velocidades angulares del dron w_x, w_y, w_z (medidas dado el sensor abordo - Giroscopio) y el jacobiano de la ecuación 67.

$$\begin{bmatrix} d\phi \\ d\theta \\ d\psi \end{bmatrix} = {}^{B}J^{-1}_{rpy(3x3)} \times \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \tag{83}$$

Para implementar esto de manera física, en el script de Matlab de la derivación `Derivacion_dron.m` se creó una función simbólica de la ecuación anterior 83 llamada **jacobiano (*symfun*)**. Dicha función simbólica fue utilizada para crear la función numérica para la implementación física dentro del controlador de vuelo. Esta función numérica fue generada con el siguiente comando de matlab:

```
matlabFunction(jacobiano, 'file', 'jacobiano.m');
```

```
function [droll, dpitch, dyaw] = jacobiano(roll,pitch,yaw,wx,wy,wz)
%JACOBIANO
|
%    [DROLL, DPITCH, DYAW] = JACOBIANO(ROLL,PITCH,YAW,WX,WY,WZ)
    t2 = cos(pitch);
    t3 = cos(roll);
    t4 = sin(pitch);
    t5 = sin(roll);
    t6 = t3.^2;
    t7 = t5.^2;
    t8 = 1.0./t2;
    t9 = t6+t7;
    t10 = 1.0./t9;
    droll = t8.*t10.*(t2.*t6.*wx+t2.*t7.*wx+t4.*t5.*wy+t3.*t4.*wz);
    dpitch = t10.*(t3.*wy-t5.*wz);
    dyaw = t8.*t10.*(t5.*wy+t3.*wz);
end
```

Figura 143: Jacobiano numérico implementado.

Paso 3: Establecer U1

Establecer u1 igual al $Thrust_{des}$ (colocar un valor cualquiera buscando que el drone no oscile). De manera experimental se probó que el mejor valor para el control dentro de la plataforma fue de 45 unidades.

Paso 4,5 y 6: Calcular U2, U3 y U4.

Para calcular las cuatro variables de control para los motores se unificó todo dentro de un bloque de ***Matlab function*** de simulink. En donde los parámetros de entrada serian las tres mediciones de los ángulos de Euler del paso 1, los tres ángulos deseados (referencias de control) dado sliders en radianes, el *Thrust* deseado de 45 grados, el tuneo para las constantes del control PD para cada una de las tres orientaciones y la lectura de las velocidades angulares dado el bloque del sensor del giroscopio. Todo esto se resume en la siguiente Figura 9.0.3.

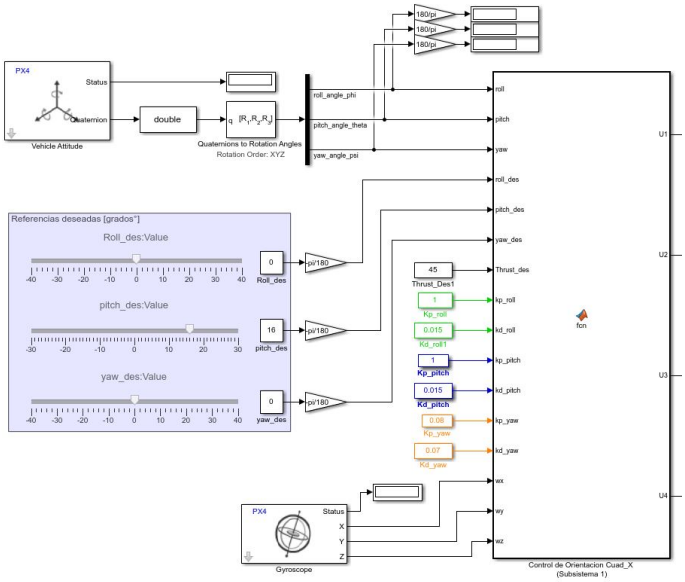


Figura 144: Control de orientación - Subsistema 1

Teniendo los parámetros de entrada establecidos, y utilizando el jacobiano numérico del paso 2. Se tiene definida por completo la función del bloque de Matlab para este primer subsistema 9.0.3.

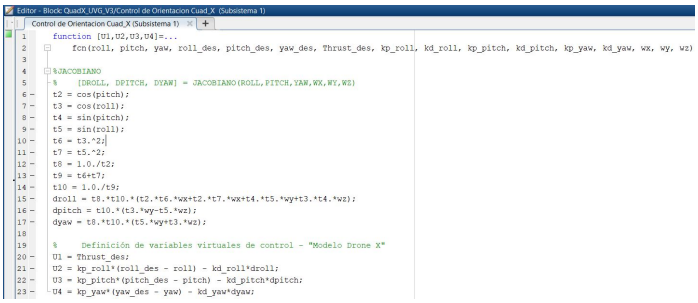


Figura 145: $Matlab function$ para el subsistema 1 del control de orientación.

Paso 7: Calcular las velocidades de los motores

Para calcular las velocidades de los motores se debe emplear la inversa de la matriz de mezclado K definida en la ecuación 76, para poder despejar las velocidades angulares deseadas.

Luego con el script de la derivación nuevamente se volvió a crear una función simbólica llamada "mixer", con el comando `mixer(u1, u2, u3, u4) = K2[u1; u2; u3; u4]`.

Luego de manera similar que para el subsistema 1, se obtiene la función numérica para el motor mixer, realizada con el comando de Matlab siguiente: `matlabFunction(mixer, 'file', 'mixer2.m');`

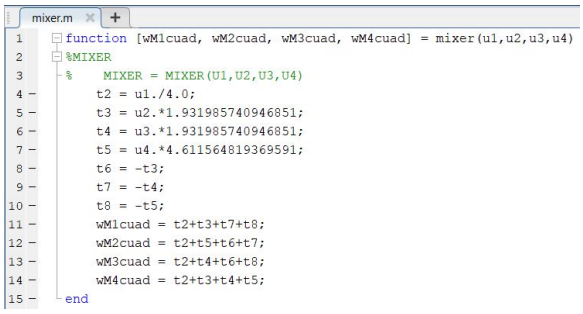


Figura 146: Función numérica para el motor mixer.

Ya que esta función numerica devuelve las velocidades de los motores en rad/s elevadas al cuadrado hay que hacer algunos ajustes para la implementación.

```
function [wM1_rad, wM2_rad, wM3_rad, wM4_rad]= fcn(u1,u2,u3,u4)
%MOTOR MIXER
t2 = u1./4.0;
t3 = u2.*1.931985740946851;
t4 = u3.*1.931985740946851;
t5 = u4.*4.611564819369591;
t6 = -t3;
t7 = -t4;
t8 = -t5;
wM1_cuad = t2+t3+t7+t8; %(rad/s)^2 doubles
wM2_cuad = t2+t5+t6+t7; %(rad/s)^2
wM3_cuad = t2+t4+t6+t8; %(rad/s)^2
wM4_cuad = t2+t3+t4+t5; %(rad/s)^2
if wM1_cuad < 0
    wM1_cuad  = wM1_cuad - wM1_cuad;
end
if wM2_cuad < 0
    wM2_cuad = wM2_cuad - wM2_cuad;
end
if wM3_cuad < 0
    wM3_cuad = wM3_cuad - wM3_cuad;
end
if wM4_cuad < 0
    wM4_cuad = wM4_cuad - wM4_cuad;
end
wM1_rad = sqrt(wM1_cuad); %(rad/s)
wM2_rad = sqrt(wM2_cuad); %(rad/s)
wM3_rad = sqrt(wM3_cuad); %(rad/s)
wM4_rad = sqrt(wM4_cuad); %(rad/s)

% Rad_seg2RPM = (60/(2*pi)) %(rad/s)--> RPM's
% wM1_rpm = wM1_rad*(60/(2*pi)); % RPM
% wM2_rpm = wM2_rad*(60/(2*pi)); % RPM
% wM3_rpm = wM3_rad*(60/(2*pi)); % RPM
% wM4_rpm = wM4_rad*(60/(2*pi)); % RPM

% Ecuacion de la recta  que mapea RPM a PWM(1000-2000uS) de los ESC's
% y = 25x/117 + 1000;
```

Figura 147: Matlab function del Motor mixer (subsistema 2).

Para que esto sea implementable lo primero que se debe realizar es aplicar programación

defensiva para asegurarnos que las velocidades de los motores siempre sean positivas (expresiones condicionales con los 4 *if*). Luego se sacan las raíces cuadradas a las velocidades de los motores para obtenerlas en [rad/s]. Luego se hace una conversión de unidades para los motores de rad/s a RPM. Por último se mapean esas RPM de cada motor a la señal PWM (por medio de una ecuación de la recta) para utilizarla en el bloque del paquete de soporte del controlador de vuelo ***PX4 PWM Output***.

Finalmente se obtiene el modelo del control de orientación (en bloques de simulink) para la configuración en X del cuadricóptero.

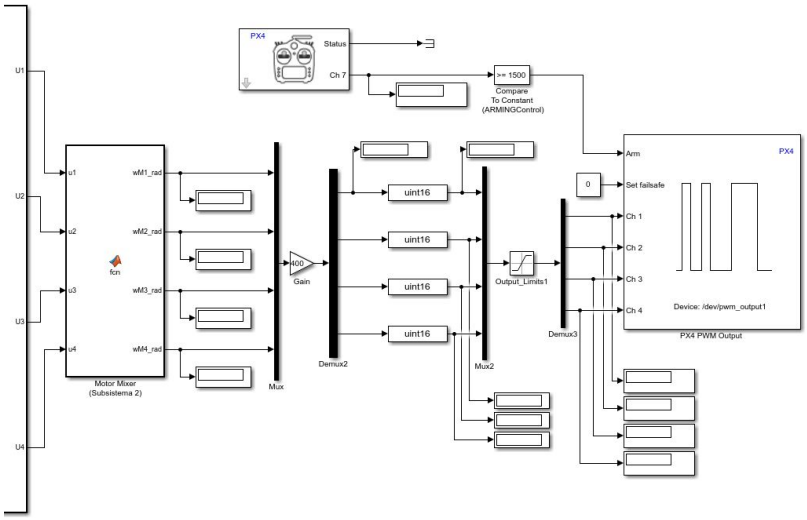


Figura 148: Control de orientación - Subsistema 2.

El controlador final de orientación presenta la forma siguiente.

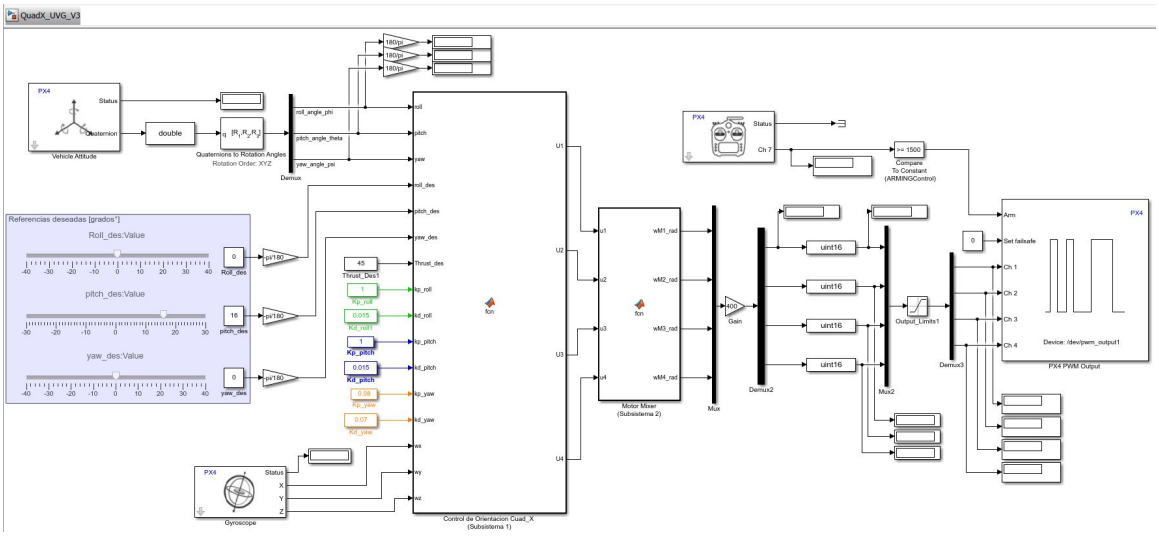


Figura 149: Control de orientación - Final v3.

9.0.4. Resultados finales de la orientación

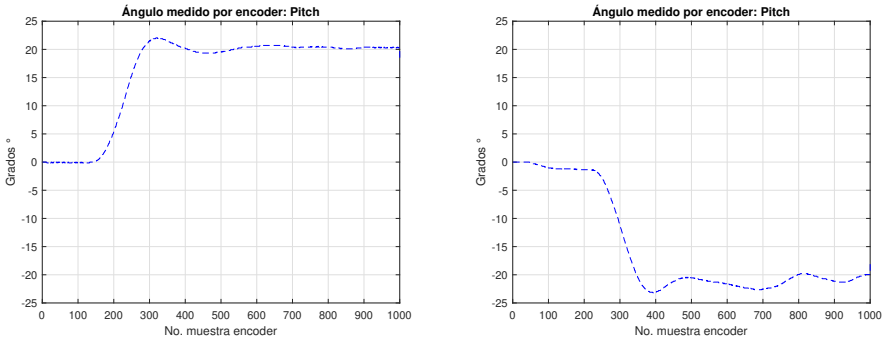


Figura 150: Plot de referencia pitch 20° y -20° respectivamente.

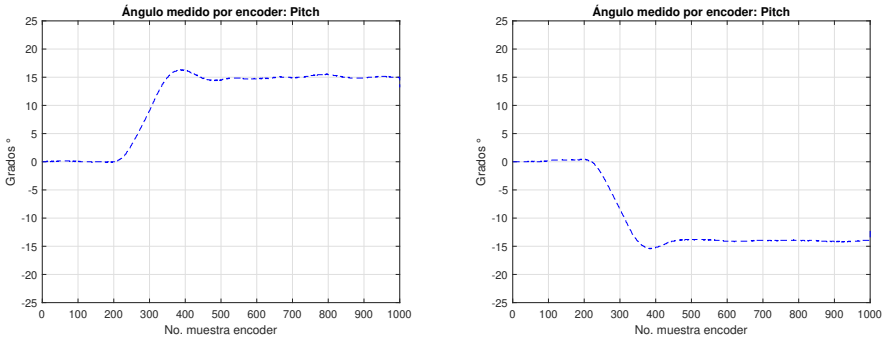


Figura 151: Plot de referencia pitch 15° y -15° respectivamente.

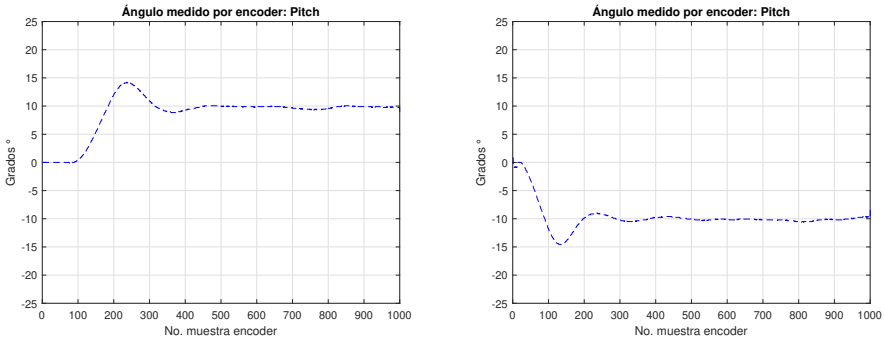


Figura 152: Plot de referencia pitch 10° y -10° respectivamente.

Enlaces videos: Tesis 2021- Control de orientación v1, v3 y v5

- Control de orientación v1: `https://youtu.be/fmmDkhxjccc`
- Control de orientación v3: `https://youtu.be/sRJqAOuGA-A`
- Control de orientación v5: `https://youtu.be/Ndtb-j2Scqk`

CAPÍTULO 10

Conclusiones

1. Se logró diseñar un marco adecuado para el cuadricóptero fijado a la plataforma de pruebas de tres grados de libertad.

2. Se determinó que es posible modificar el firmware y programar el controlador de vuelo Pixhawk 1 desde el entorno de programación visual Matlab/Simulink gracias al paquete de soporte para autopilotos de PX4.

3. Por medio de la lectura de posición de los encoders rotacionales se validó que el diseño final del marco del dron, puede maniobrar dentro de la estructura de la plataforma de manera eficaz únicamente para los ángulos de cabeceo y guiñada.

4. Por medio de control LQR, se logró implementar un controlador que estabiliza en cero grados al cuadricóptero en sus tres ángulos de vuelo (roll, pitch, yaw) respectivamente alineándose siempre al marco global de vuelo NED (Control de Orientación v1).

5. Por medio de control PD, se logró implementar un controlador que estabiliza la actitud del cuadricóptero en referencias variables deseadas para los ángulos del roll y del pitch (Control de Orientación v3).

6. Debido al pandeo y al peso de la estructura para el anillo del *roll*, el control mas adecuado fue únicamente para el anillo o eje del *pitch*.

7. Se logró diseñar una tarjeta de circuito impreso capaz de alimentar el consumo de los cuatro motores (12V-5.1A) sin escobillas del cuadricóptero, además de cumplir con la funcionalidad, de formar parte de base inferior del marco del dron.

8. Se determinó que la mejor forma de ensamblar de manera fija el cuadricóptero, dada la inestabilidad debido a la ubicación del CG y el pandeo de los anillos de aluminio de la plataforma, fue sujetando los cuatro brazos del multirrotor directamente en el eje de cabeceo de la plataforma.

9. Para los motores sin escobillas de $390Kv$ y $935Kv$ con una hélice de 8 pulgadas (8045), se determinó que la fuerza de empuje fue de 261gf y 430gf respectivamente.

10. Para el análisis de esfuerzos de los brazos del multirrotor, se determinó teóricamente que el esfuerzo de flexión dada la fuerza de empuje (261 gf) de cada motor generada por las hélices de 8 pulgadas, fue de 1.18MPa, con el cual se obtuvo un factor de seguridad de 11. Tanto para el análisis de elementos finitos realizado en ANSYS, como para el de estrés realizado en Inventor, se obtuvo aproximadamente el mismo factor de seguridad calculado teóricamente.

11. Por medio del criterio de falla de carga estática (energía de distorsión) y también por las simulaciones de análisis de elementos finitos de ANSYS e Inventor se determinó que la pieza no fallará debido a la carga generada por la fuerza de los motores.

CAPÍTULO 11

Recomendaciones

1. Es aconsejable rediseñar los perfiles de los anillos para los tres ángulos de vuelo en otro materia que no presente problemas de pandeo. Uno de los mejores material para esta aplicación en específico podría ser la fibra de carbono, ya que presenta una buena relación entre peso, pandeo y resistencia.

2. Se recomienda trabajar con las última versión de Matlab/Simulink (2021a al menos) así como también la versión mas reciente del controlador de vuelo de PX4 (Pixhawk 4).

3. Se recomienda rediseñar las dimensiones de la estructura de los anillos de para los 3 movimientos, ya que el espacio tan limitado no dejó aprovechar las características del motor Hitec 4108 de $390Kv$ con el tamaño de hélice recomendada por el fabricante (13 pulgadas mínimo).

4. Se recomienda utilizar los motores mas pequeños de $935Kv$, ya que a diferencia de los de los anteriores, el tamaño para estos de la hélice recomendadas si aplican las de 8 pulgadas, es por esto que generan una mayor fuerza de empuje (430gf).

CAPÍTULO 12

Bibliografía

[1] U. YÜZGEÇ, İ. ÖKTEN, H. ÜÇGÜN, A. R. GÜN, T. TÜRKYILMAZ, M. KESLER, C. KARAKUZU y U. Gökhan, "Development of the Test Platform for Rotary Wing Unmanned Air Vehicle," *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, vol. 3, n.º 2, págs. 18-24, 2016.

[2] E. B. Cortez Aguilar, "Control de orientación y altitud de un vehıculo aéreo no tripulado, del tipo cuadrirrotor.," Tesis doct., Universidad Autónoma de Nuevo León, 2016.

[3] F. dos Santos Barbosa, "4DOF Quadcopter: development, modeling and control," Tesis doct. DOI: 10.11606/d.3.2017.tde-23102017-144556. dirección: https://doi.org/10.11606/d.3.2017.tde-23102017-144556.

[4] G. Martínez, "Plataforma de pruebas y ajustes de sistemas de control para vehículos multirrotores," Tesis doct., Universidad del Valle de Guatemala, 2019.

[5] R. Lazzari, "Development of Simulation Technologies for Assessing Multi-Rotor Unmanned Aerial Vehicle Performance in Precision Agriculture Operations," Tesis doct., Politecnico di Torino, 2020.

[6] M. F. Santos, M. F. Silva, V. F. Vidal, L. M. Honorio, V. L. M. Lopes, L. A. Z. Silva, H. B. Rezende, J. M. S. Ribeiro, A. S. Cerqueira, A. A. N. Pancoti y B. A. Regina, "Experimental Validation of Quadrotors Angular Stability in a Gyroscopic Test Bench," en *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, IEEE, oct. de 2018. DOI: 10.1109/icstcc.2018.8540660. dirección: https://doi.org/10.1109/icstcc.2018.8540660.

[7] E. Dynamics. (sep. de 2018). "FFT GYRO series 450," dirección: https://eurekadynamics.com/fft-gyro-450/.

[8] Radiolink. (sep. de 2021). "DIY Quadcopter ARF Kit," dirección: https://www.radiolink.com/f450.

[9] E. D. N. Guerrero. (mar. de 2020). "Desarrollo de un modelo matemático, cinemático y dinámico con la aplicación de software, para modificar el funcionamiento de un dron, para que este realice monitoreo automático," dirección: https://www.recimundo.com/index.php/es/article/view/814/1323.

[10] S. Govindarajan. (mayo de 2014). "Design of Multicopter Test Bench." DOI: 10.7763/IJMO.2013.V3.276, dirección: http://www.ijmo.org/index.php?m=content%5C&c=index%5C&a=show%5C&catid=37%5C&id=293.

[11] H. Bolandi. (dic. de 2012). "Attitude Control of a Quadrotor with Optimized PID Controller." DOI:10.4236/ica.2013.43040, dirección: https://www.scirp.org/html/12-7900231_35654.htm#Figure%203.

[12] E. A. Paiva Peredo. (mayo de 2016). "Modelado y control de un cuadricóptero," dirección: https://pirhua.udep.edu.pe/handle/11042/2514.

[13] T. Bresciani. (oct. de 2008). "Modelling, Identification and Control of a Quadrotor Helicopter," dirección: https://pirhua.udep.edu.pe/handle/11042/2514.

[14] E. A. Paiva Peredo, "Modelado y control de un cuadricóptero," 2016.

[15] G. F. Franklin, J. D. Powell, A. Emami-Naeini y J. D. Powell, *Feedback control of dynamic systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 4.

[16] T. Wescott, "PID without a PhD," *Embedded Systems Programming*, vol. 13, n.º 11, págs. 1-7, 2000.

[17] Auterion. (). "The story of PX4 and Pixhawk," dirección: https://auterion.com/company/the-history-of-pixhawk/.

[18] PX4. (2021). "3DR Pixhawk 1 Flight Controller (Discontinued)," dirección: https://docs.px4.io/master/en/flight_controller/pixhawk.html#_3dr-pixhawk-1-flight-controller-discontinued.

[19] Mathworks. (2021). "Integration with General PX4 Architecture," dirección: https://la.mathworks.com/help/supportpkg/px4/ug/px4-capabilities-integration.html.

[20] Q. Quan, X. Dai y S. Wang, *Multicopter Design and Control Practice: A Series Experiments Based on MATLAB and Pixhawk*. Springer Nature, 2020.

[21] MathWorks. (2019). "UAV Toolbox Support Package for PX4 Autopilots," dirección: UAV%20Toolbox%20Support%20Package%20for%20PX4%20Autopilots.

[22] P. U. Guide. (2020). "Pixhawk Wiring Quick Start," dirección: https://docs.px4.io/master/en/assembly/quick_start_pixhawk.html.

[23] V. Oguntosin y A. Akindele, "Design of a joint angle measurement system for the rotary joint of a robotic arm using an Incremental Rotary Encoder," en *Journal of Physics: Conference Series*, IOP Publishing, vol. 1299, 2019, pág. 012 108.

[24] Argos. (2015). "ENCODER DE CUADRATURA," dirección: https://nomadaselectronicos.wordpress.com/2013/02/01/encoder-de-cuadratura/.

[25] R. Petrella y M. Tursini, "An embedded system for position and speed measurement adopting incremental encoders," *IEEE Transactions on industry applications*, vol. 44, n.º 5, págs. 1436-1444, 2008.

[26] G. Solchaga Pérez de Lazárraga, "Control motor brushless sensorless," 2015.

[27] P. Yedamale, "Brushless DC (BLDC) motor fundamentals," *Microchip Technology Inc*, vol. 20, n.º 1, págs. 3-15, 2003.

[28] J. Reid. (ene. de 2017). "Comprensión de las clasificaciones de Kv," dirección: https://www.rotordronepro.com/understanding-kv-ratings/.

[29] J. E. Shigley, C. R. Mischke, F. P. Bocanegra y C. O. Correa, "Diseño en ingeniería mecánica," 2008.

[30] Parrot. (sep. de 2010). "Parrot AR.Drone 2.0 Quadcopter (Blue/Orange)," dirección: https://www.bhphotovideo.com/c/product/999250-REG/parrot_pf721002_drone_2_0_quadricopter_control.html.

[31] Hitec. (sep. de 2021). "Energy Propel 4108 / 40 Integrated Power System," dirección: https://hitecrcd.com/products/airplanes/multirotor-motors-escs/energy-propel-410840-integrated-power-system/product.

[32] Emax. (sep. de 2020). "EMX-MT-1534- Motor multicóptero EMAX MT2213 (con combo Prop1045) 935KV," dirección: https://emaxmodel.com/collections/wholeoff/products/emx-mt-1534-emax-multicopter-motor-mt2213-with-prop1045-combo935kv?_pos=1%5C&_sid=54c9b8884%5C&_ss=r.

[33] S. N. (sep. de 2004). "ANSI PCB Trace Width Calculator," dirección: https://www.desmith.net/NMdS/Electronics/TraceWidth.html.

[34] NWC. (sep. de 2021). "Circuit Wizard," dirección: new-wave-concepts.com/ed/circuit.html.

[35] Altium. (sep. de 2021). "Board Shape Import Using a DXF/DWG Drawing," dirección: https://my.altium.com/altium-designer/getting-started/board-shape-import-using-dxfdwg-drawing.

[36] A. Inventor. (abr. de 2021). "Acerca del análisis de estrés," dirección: https://knowledge.autodesk.com/support/inventor/learn-explore/caas/CloudHelp/cloudhelp/2020/ENU/Inventor-Help/files/GUID-61F01A5D-7E54-45A1-9698-7BB11F0AEE94-htm.html.

[37] C. F. Urresta Pérez, "Caracterización de las propiedades mecánicas de materiales impresos mediante la técnica de impresión 3D fused deposition modeling (FDM)," B.S. thesis, 2020.

[38] R. C. Hibbeler, *Mecánica vectorial para ingenieros: estática*. Pearson Educación, 2004.

[39] MatWeb. (sep. de 2021). "Overview of materials for Acrylonitrile Butadiene Styrene (ABS), Extruded," dirección: http://www.matweb.com/search/DataSheet.aspx?MatGUID=3a8afcddac864d4b8f58d40570d2e5aa%5C&ckck=1.

[40] *Encoder Library, for Measuring Quadarature Encoded Position or Rotation Signals*, https://www.pjrc.com/teensy/td_libs_Encoder.html, (Accessed on 12/10/2021).

[41] D. Mellinger, *Trajectory generation and control for quadrotors*. University of Pennsylvania, 2012.

[42] R. S. M. Hutter, *ETH Zurich - 2015 Rotorcraft - Dynamic Modeling of Rotorcraft & Control.pptx*, https://ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/rsl-dam/documents/RobotDynamics2017/10-RotorcraftControl.pdf, (Accessed on 12/08/2021), oct. de 2015.

[43] Mathworks. (sep. de 2021). "UAV Toolbox Support Package for PX4 Autopilots User's Guide," dirección: https://www.mathworks.com/help/pdf_doc/supportpkg/px4/index.html.

CAPÍTULO 13

Anexos

13.1. Conexión Matlab/Simulink con el autopiloto PX4

En este capítulo se explicará detalladamente, paso a paso, el procedimiento que se llevó a cabo para poder instalar correctamente este paquete de soporte para el autopiloto PX4. Cabe mencionar que cada paso respectivamente se ilustrará con capturas de pantalla lo que se debería de ver para completar dichos pasos de la manera adecuada.

13.1.1. Paso 1: Descargar el Toolbox directamente de Matlab

Lo primero que se realizó fue buscar dicho toolbox desde la pestaña de *home*, en la sección de *enviroments* seleccionamos **Add-Ons** > **Get Hardware Support Packages** y buscamos **px4**.

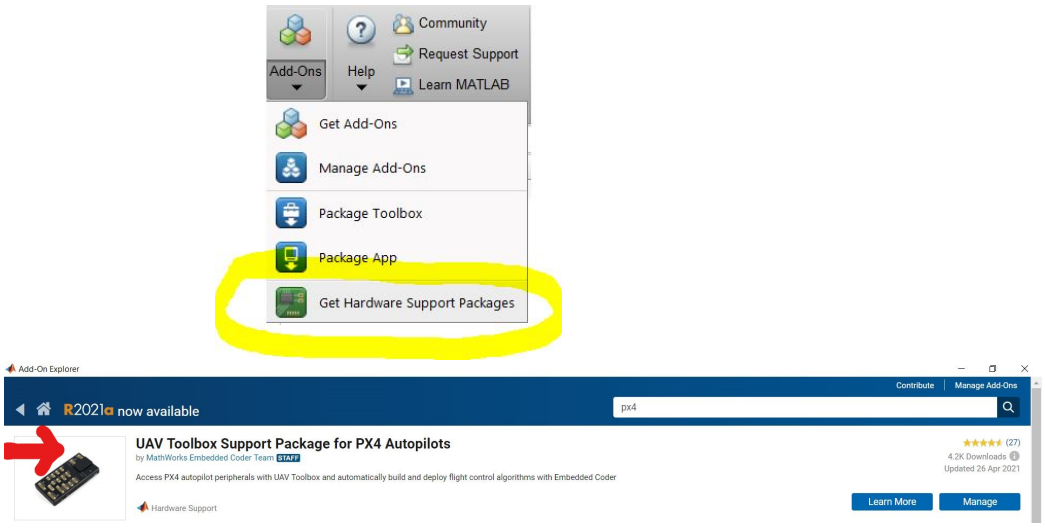


Figura 153: Paso 1 - Instalación UAV Toolbox de PX4.

13.1.2. Paso 2: Instalar el firmware del cuadricóptero X desde Mission Planner

Una vez instalado el paquete desde Matlab se debe de configurar el **hardware** desde la pestaña de *home*, en la sección de *enviroments* se selecciona **Add-Ons** > **Manage Add-Ons** y se busca el de **px4** previamente instalado. Luego selecciona el ícono del engranage y se presiona clic en **Setup**.

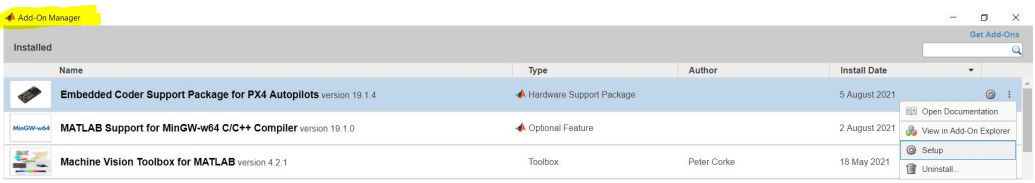


Figura 154: Paso 2a) - Configuración inicial del hardware.

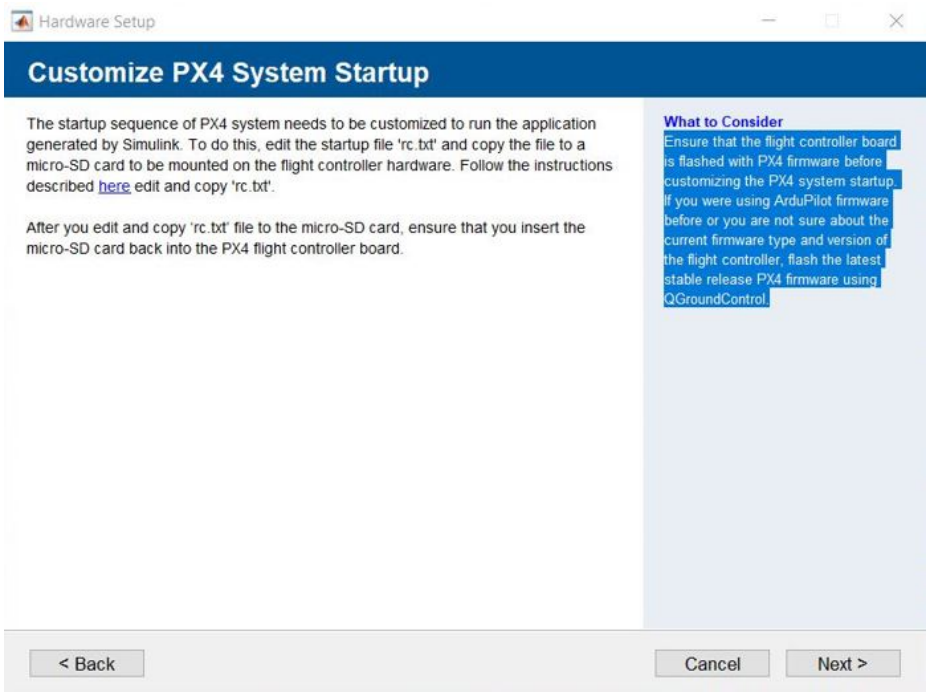


Figura 155: Paso 2b) - Configuración inicial del hardware.

Como se muestra en la Figura 155, el paquete de soporte no deja empezar a configurar el hardware del controlador de vuelo desde Matlab, si antes no se instaló el firmware del controlador al Pixhawk. Este procedimiento se debe de realizar en la plataforma comercial para el uso del autopiloto QGrounControl, pero debido a que la versión del Pixhawk con el cual se cuenta es la primera versión del mismo la plataforma para esta es MissionPlaner. A continuación se detalla el procedimiento en las siguientes figuras.

Figura 156: Paso 2c) - Carga del firmware desde Mission Planner.

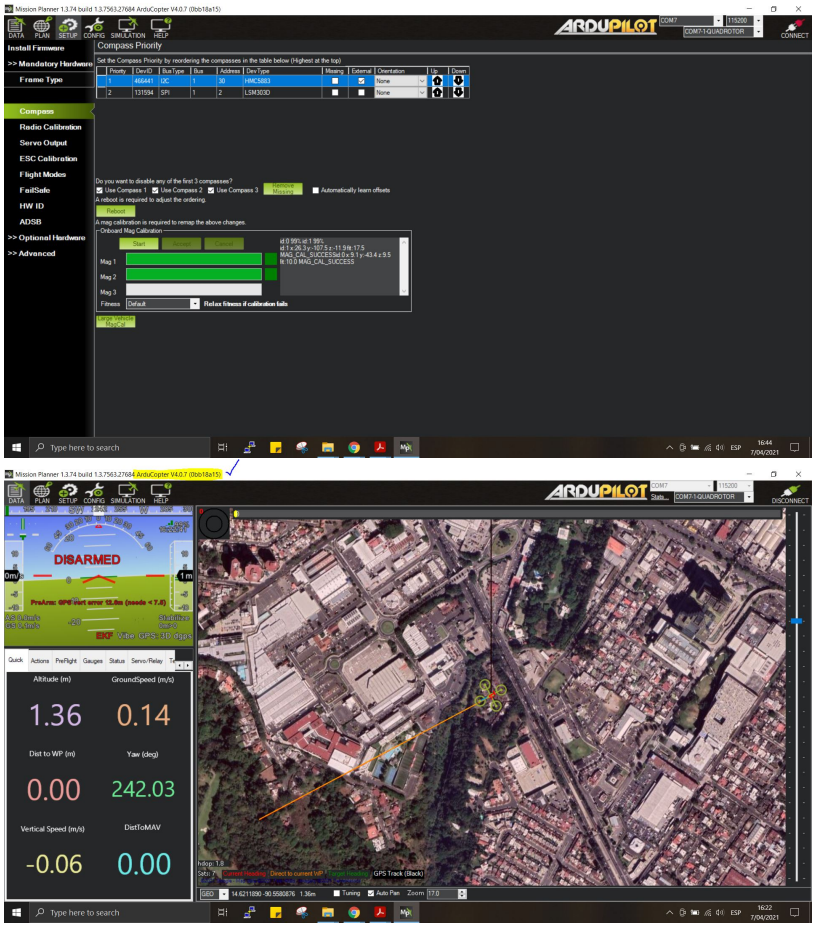


Figura 157: Paso 2d) - Calibración desde Mission Planner.

13.1.3. Paso 3: Instalar la línea de comandos de Ubuntu 18.04 LTS

Para los siguientes pasos se tuvo como referencia los siguientes enlaces de la documentación de Matlab. Dichos enlaces son:

- `https://la.mathworks.com/help/supportpkg/px4/ug/install-support-for-px4.html`
- `https://la.mathworks.com/help/pdf_doc/supportpkg/px4/index.htm` » "Embedded Coder Support Package for PX4 Autopilots User's Guide.pdf".

En cualquiera de esas dos referencias se detalla el procedimiento de instalación por realizar. A continuación se muestra la página no. 7 del pdf de la Guía para Usuarios del segundo link descrito anteriormente.

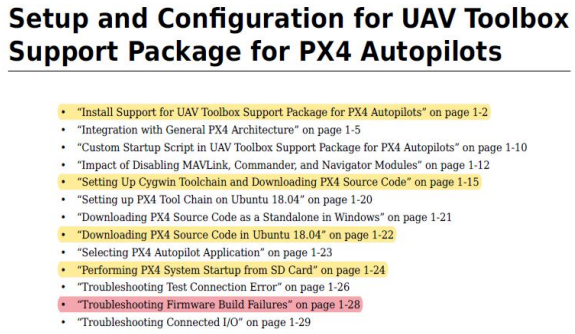


Figura 158: Resumen instalación Toolbox de PX4 - pág. 7 [43].

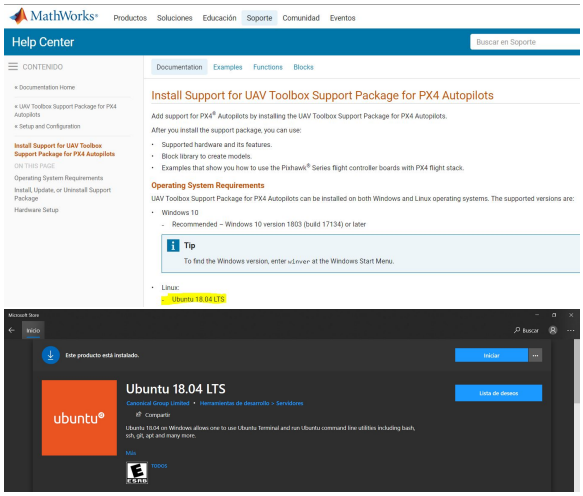


Figura 159: Paso 3 - Instalar Ubuntu 18.04 LTS.

Antes de proseguir con los siguientes pasos se debe de instalar la línea de comandos de Ubuntu para Windows 10. Dicha línea de comandos puede ser encontrada en Microsoft Store como "Ubuntu 18.04 LTS".

13.1.4. Paso 4: Toolchain de PX4 en Windows

Con el Ubuntu bash shell de Windows instalado, se prosigue a realizar el siguiente paso de nuestra instalación, el cual corresponde a la **configuración del Toolchain de PX4 en Windows** para poder así bajar el "código fuente" que utilizará el controlador para comunicarse con el paquete de soporte.

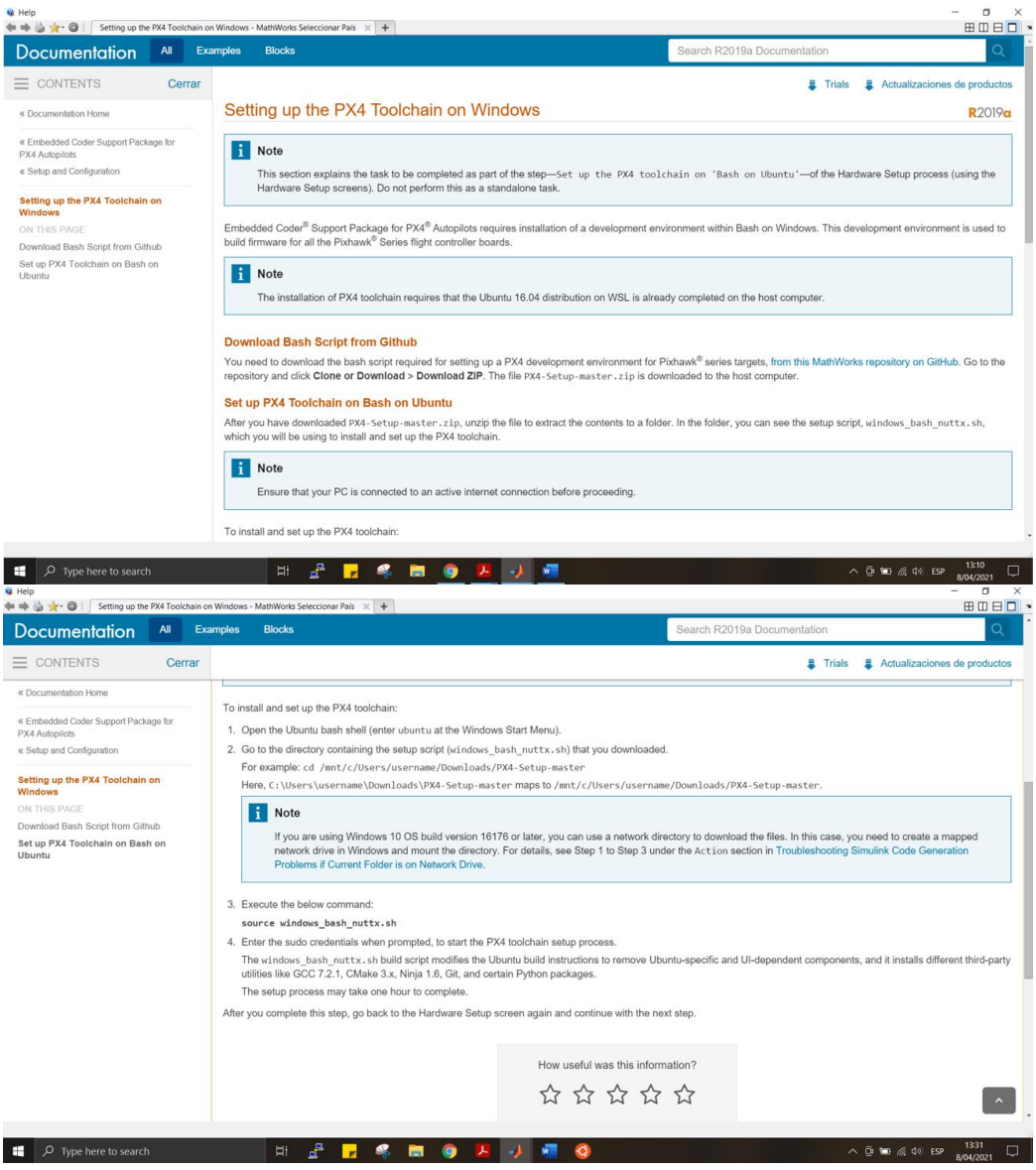
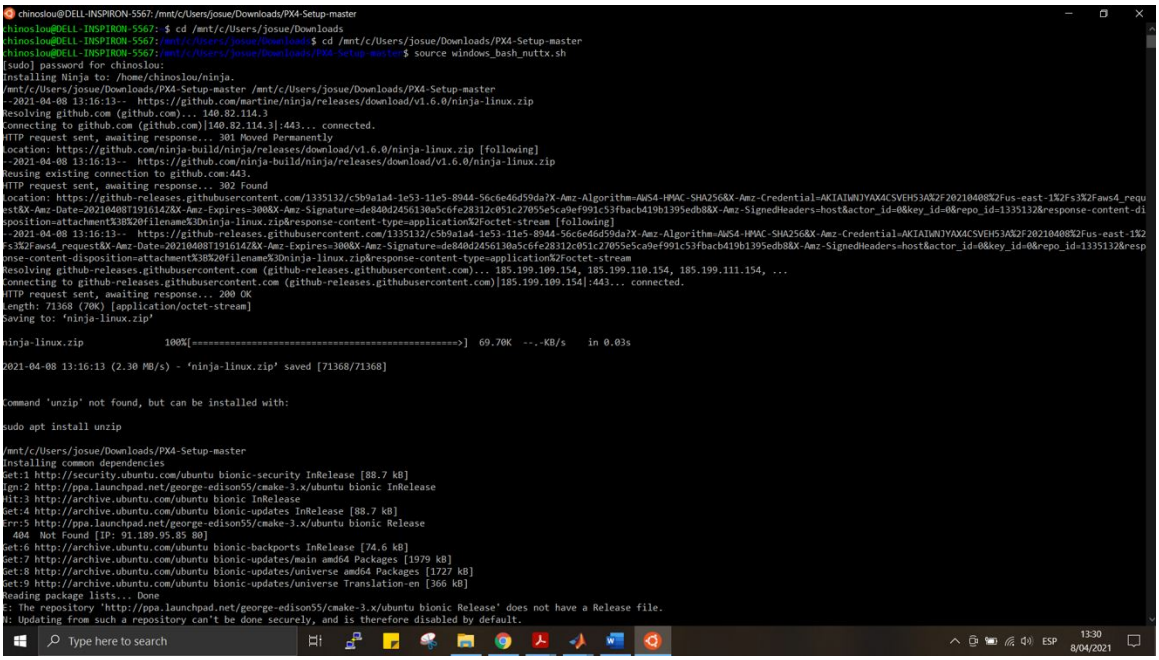


Figura 160: Paso 4 - Configuración del Toolchain de PX4.

Para ejecutar bien ese paso se debe de descargar el archivo zip que dice la página de Mathworks **PX4-Setup-master.zip**, se descomprime en descargas y desde el shell de Ubuntu se dirige al directorio que contiene el archivo **windows_bash_nuttx.sh** y se ejecuta desde allí. Este procedimiento se muestran en la Figura 161.



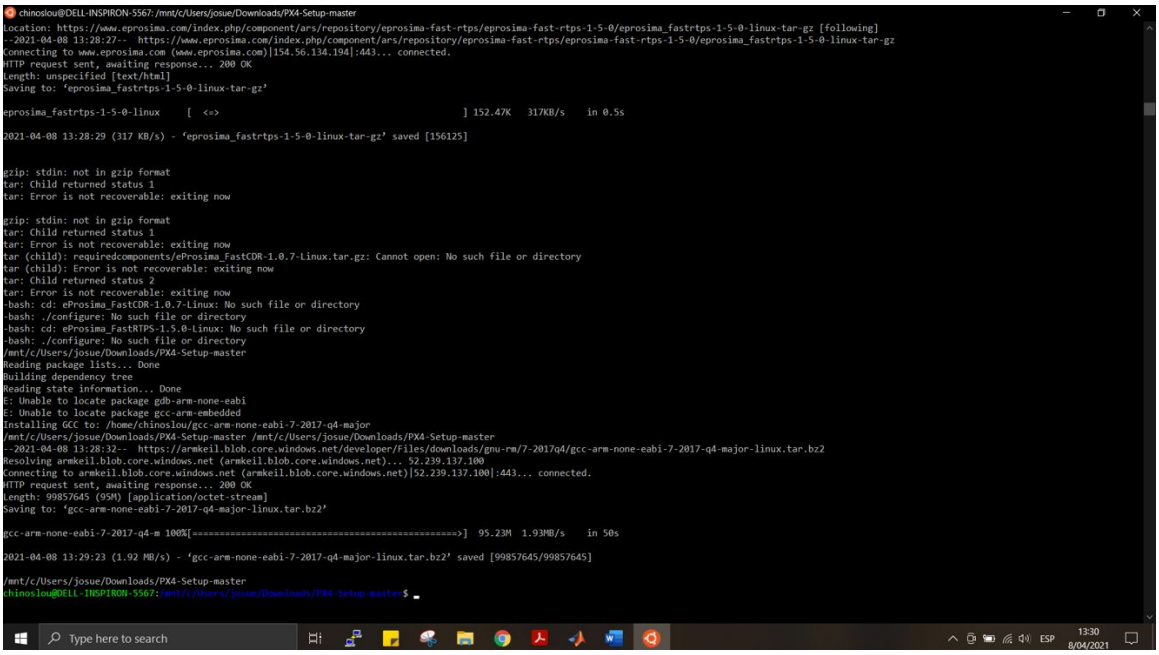


Figura 161: Paso 4 - Procedimiento desde el shell de Ubuntu.

13.1.5. Paso 5: PX4 *Source Code*

El siguiente paso en la ventana de configuración de hardware de Matlab corresponde a descargar el **PX4 source code**.

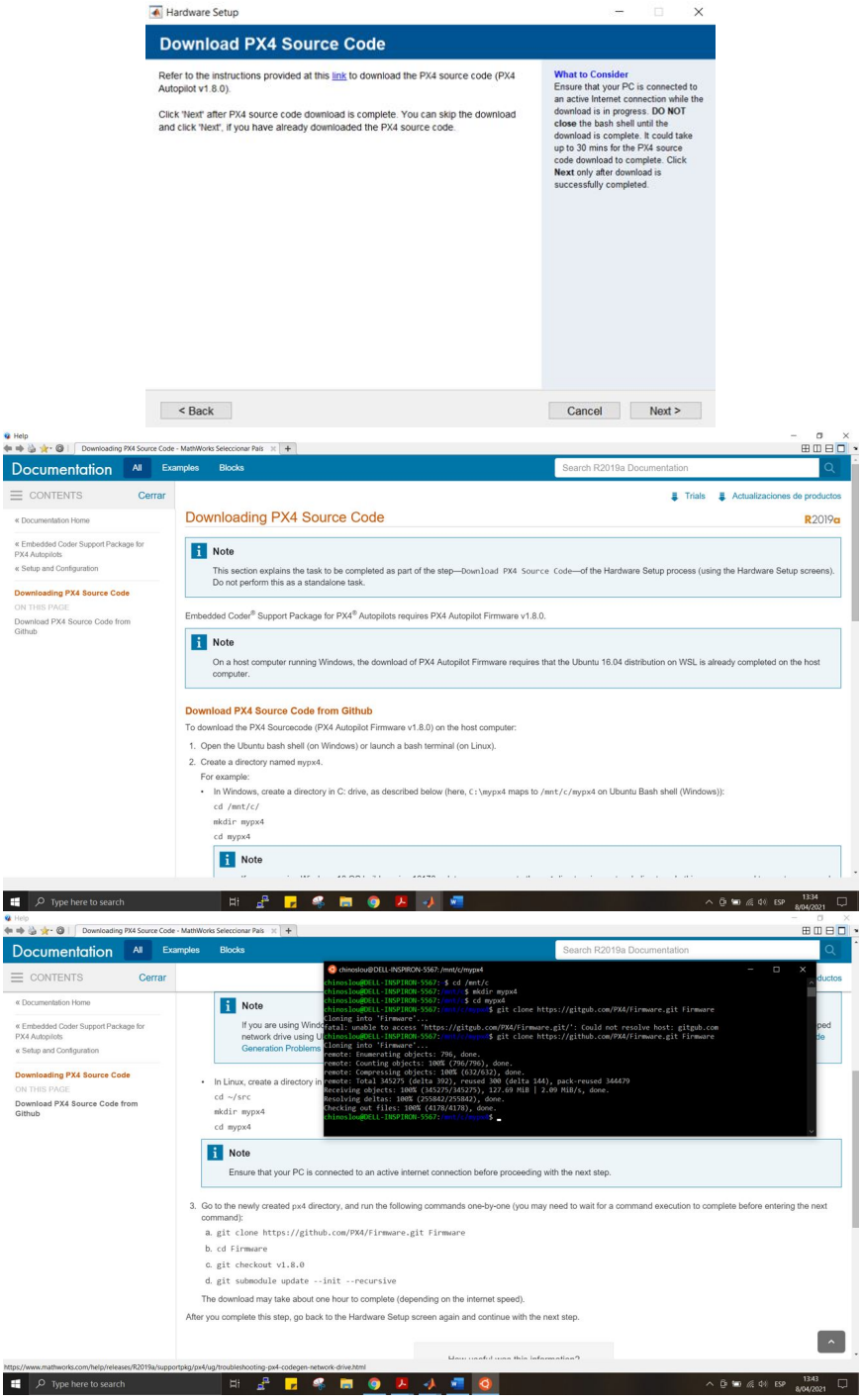


Figura 162: Paso 5 - Descarga del código fuente de PX4.

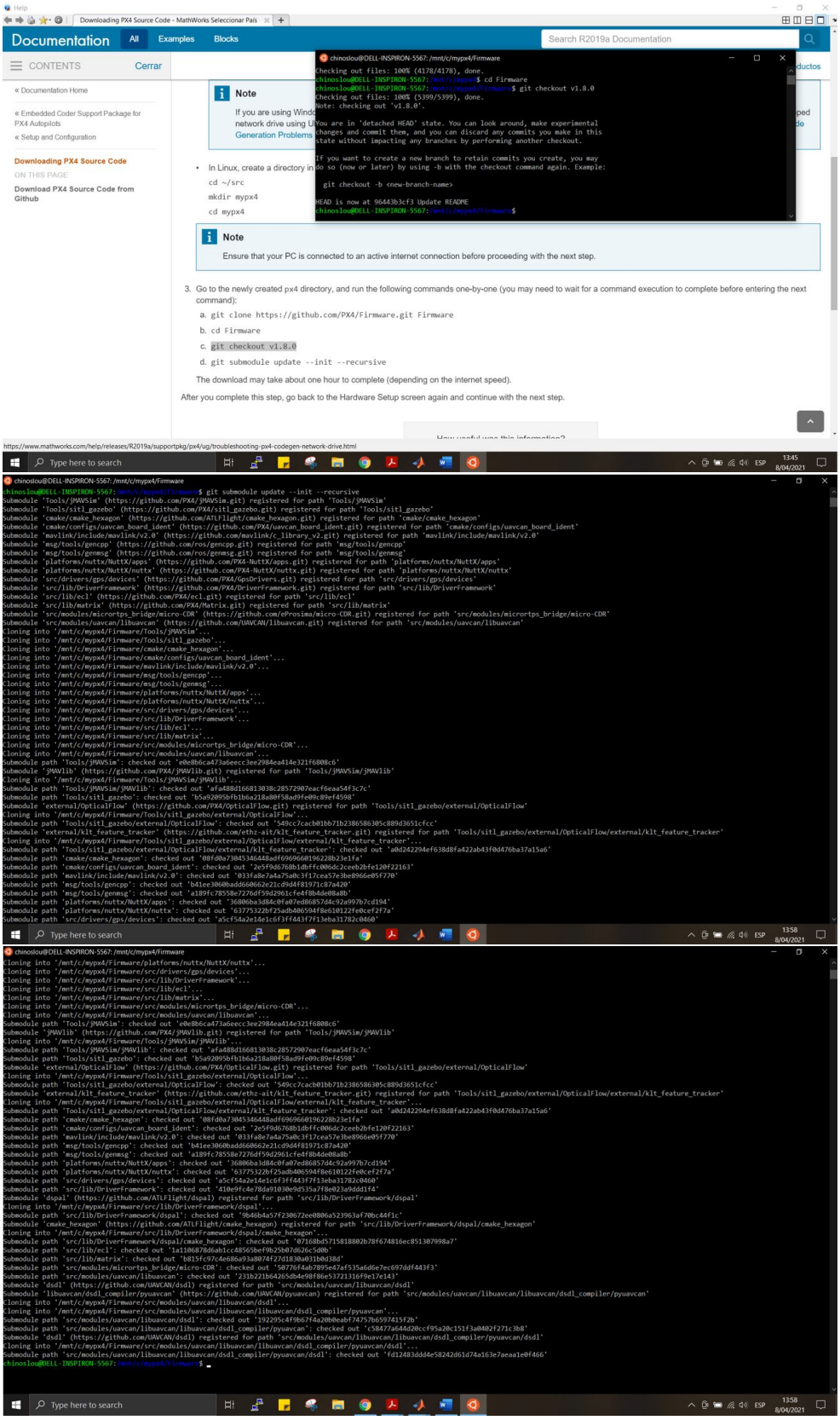


Figura 163: Paso 5 - finalizado.

13.1.6. Paso 6: Validación del PX4 *Source Code*

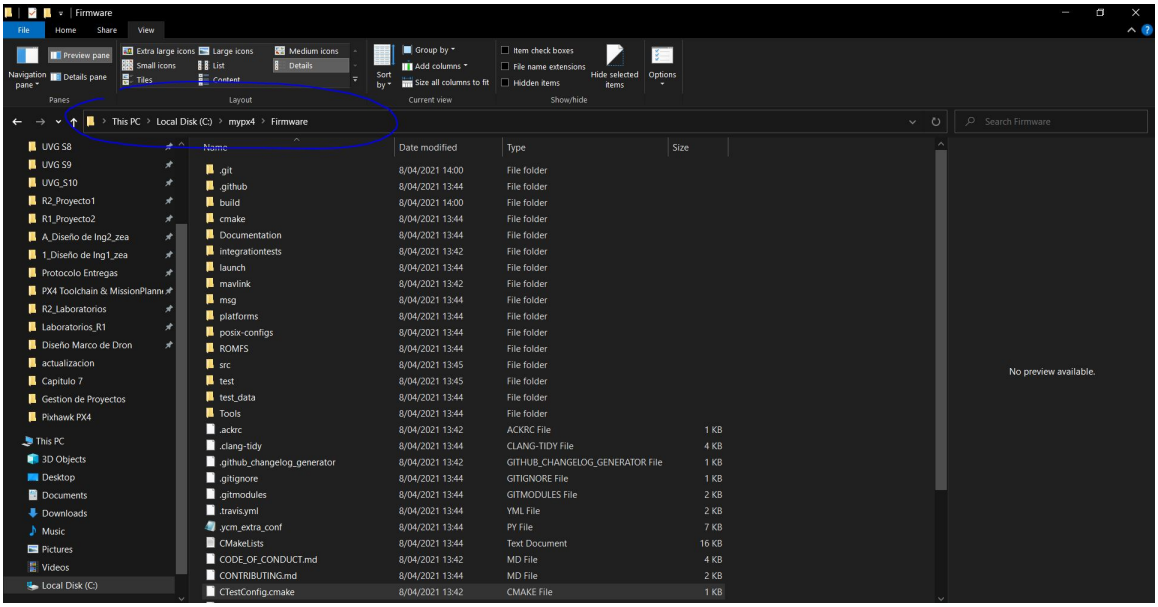


Figura 164: Paso 6 - Verificando paso anterior.

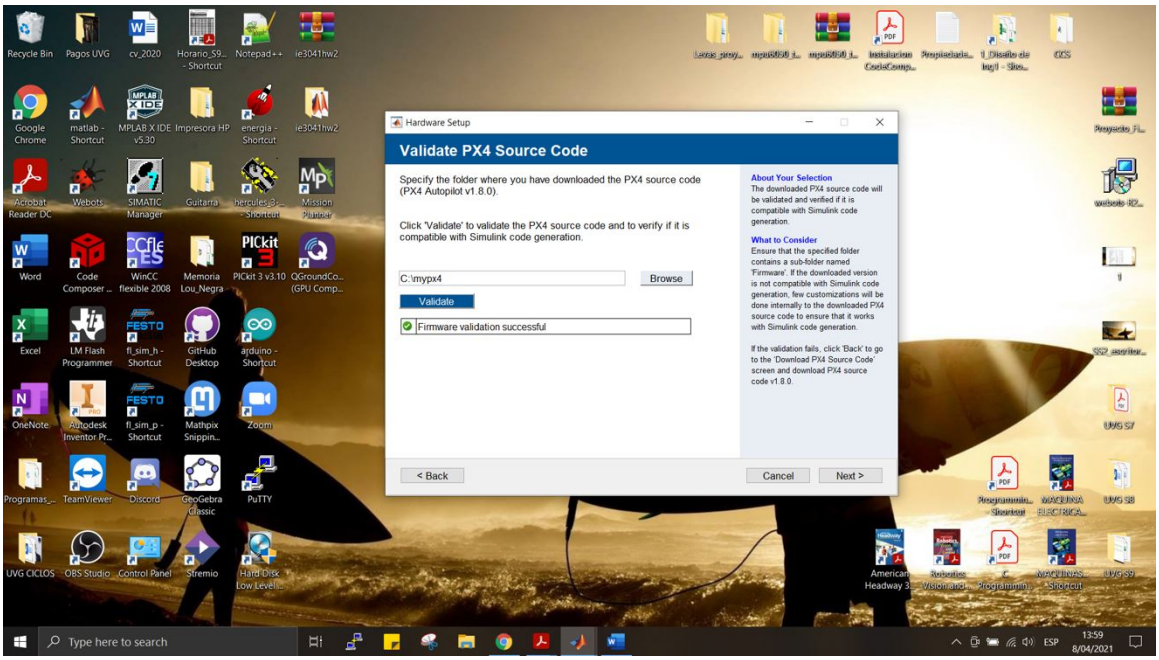


Figura 165: Paso 6 - Validación exitosa del source code.

13.1.7. Paso 7: Seleccionar la configuración CMake y compilar el firmware

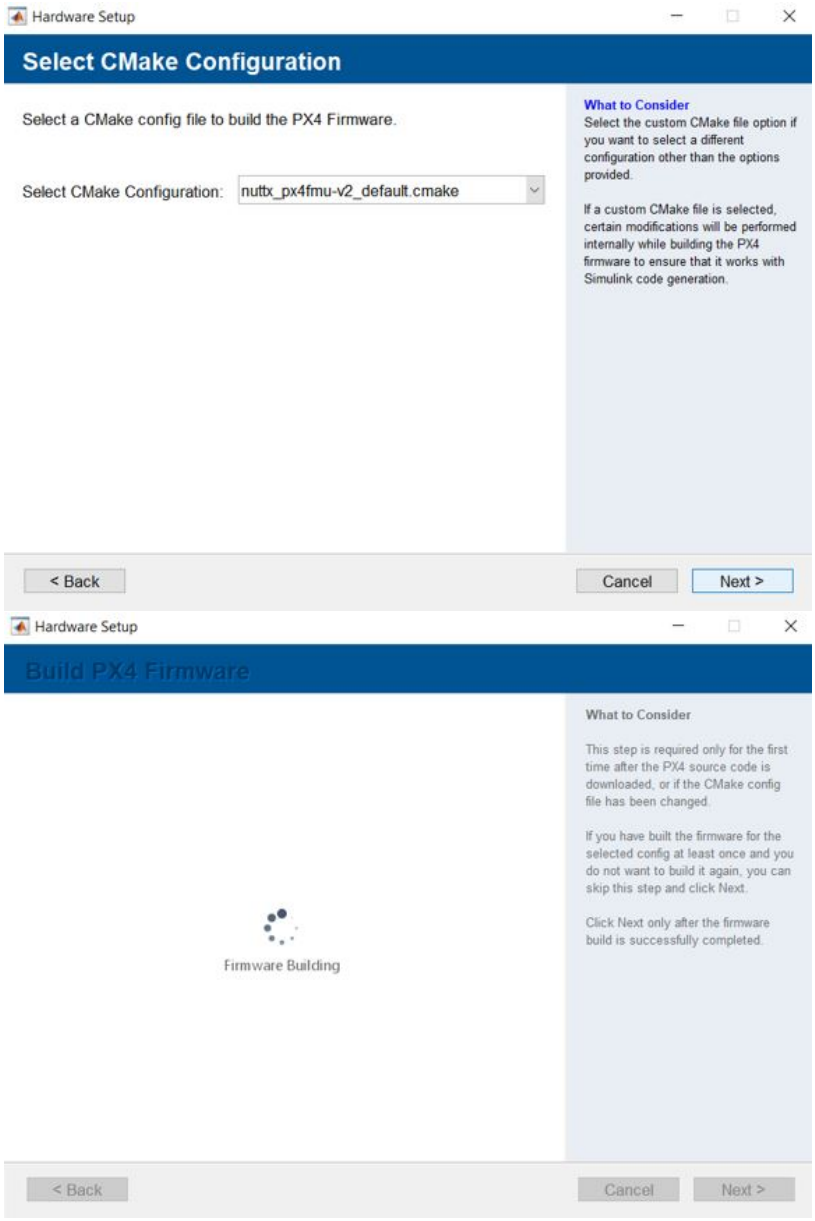


Figura 166: Paso 7 - Selección de configuración CMake para el PX4.

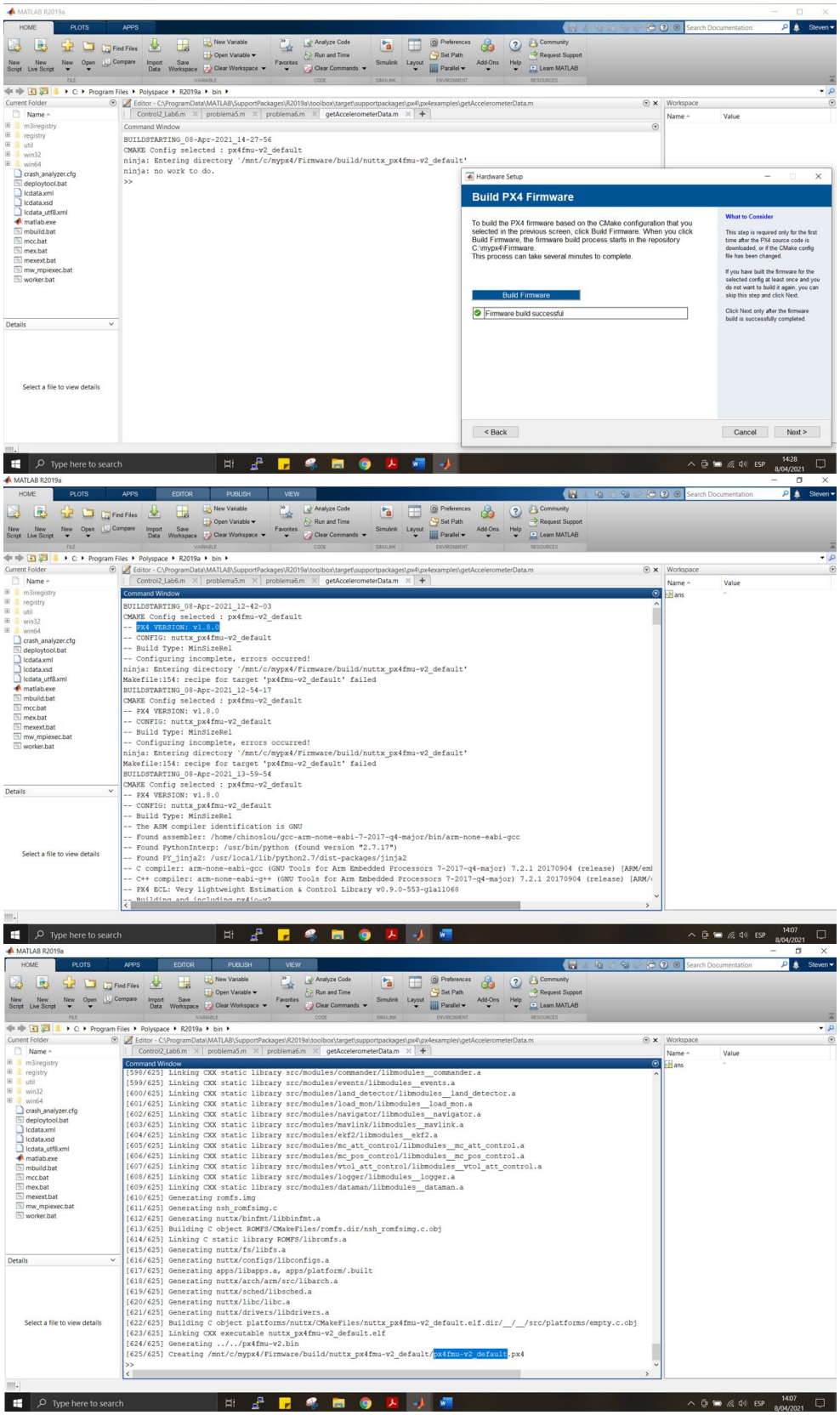


Figura 167: Paso 7 - Compilación del firmware exitoso.

13.1.8. Paso 8: Customizar el arranque del PX4 para Matlab

Este es un paso muy importante ya que de esto depende que el controlador de vuelo reconozca el entorno del paquete de Simulink/Matlab. En resumidas palabras de esto depende que se conecte o no el controlador de vuelo con Matlab.

Lo que se realiza es que se copia el archivo de arranque o "startup" en la raíz de la memoria SD que tiene incorporado el Pixhawk en donde también se encuentra la otra carpeta con el firmware de la configuración en X del cuadricóptero instalado previamente por el Mission Planner. Entonces, cuando este se conecte junto con Matlab el mismo controlador sabrá que tiene que funcionar con Simulink y no con el Mission Planner.

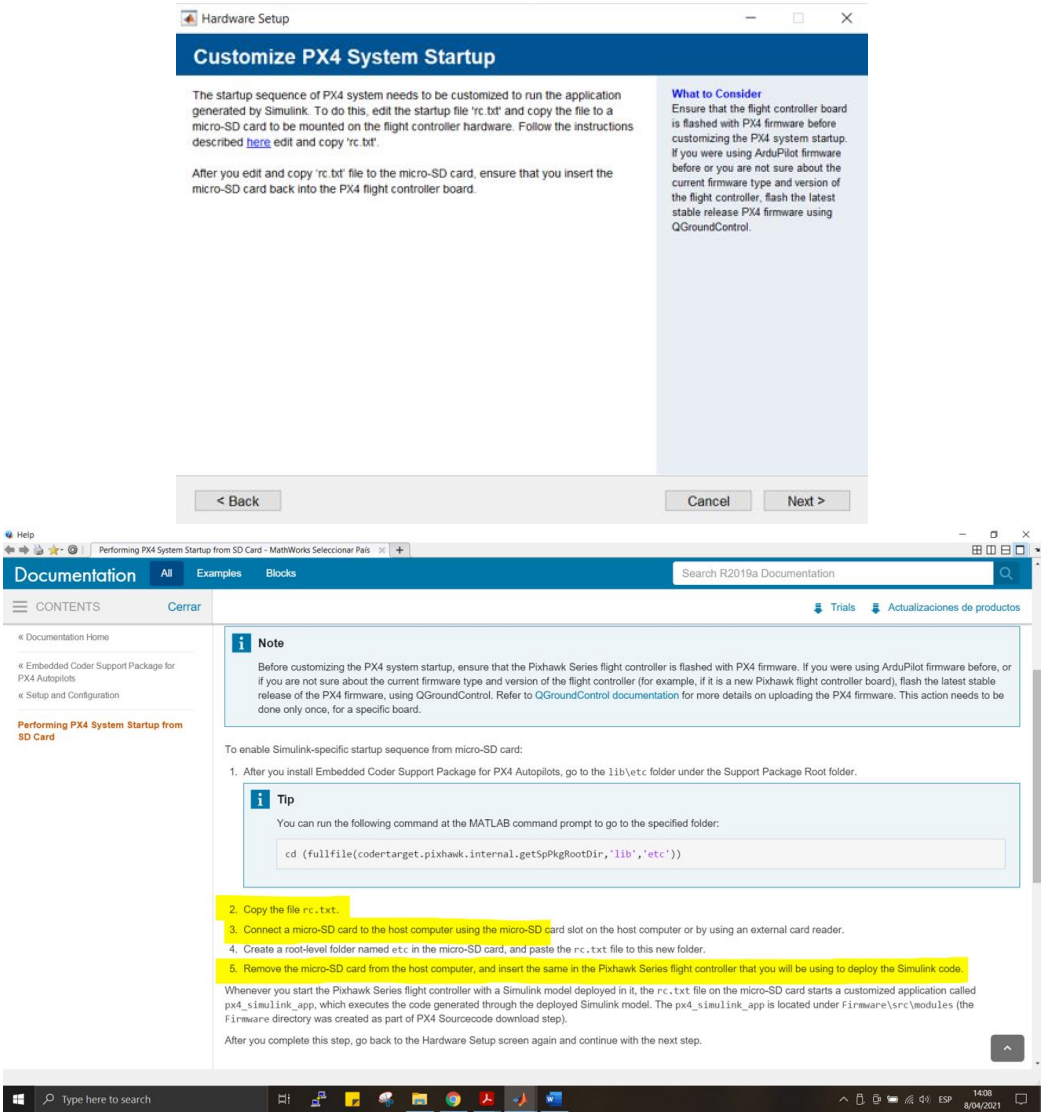


Figura 168: Paso 8 - Archivo de startup para SD Pixhawk.

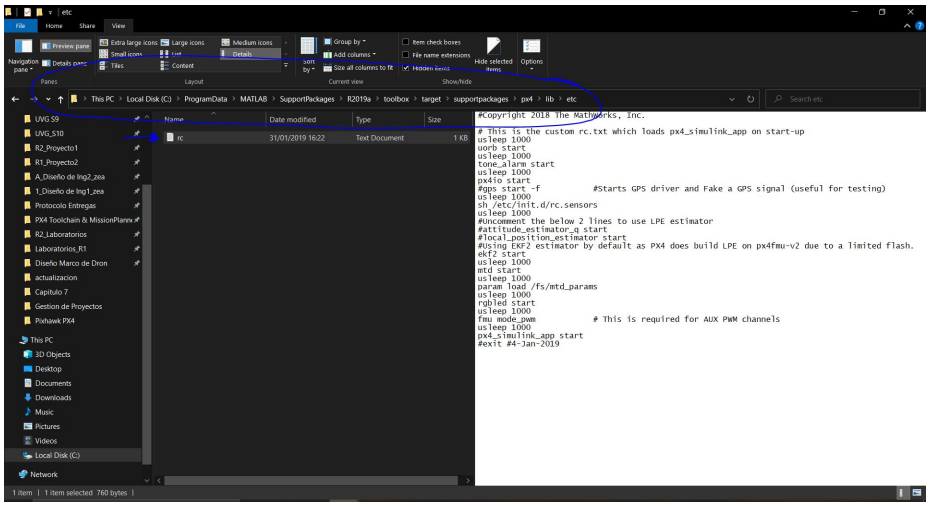


Figura 169: Paso 8 - Ubicación del archivo de texto a copiar.

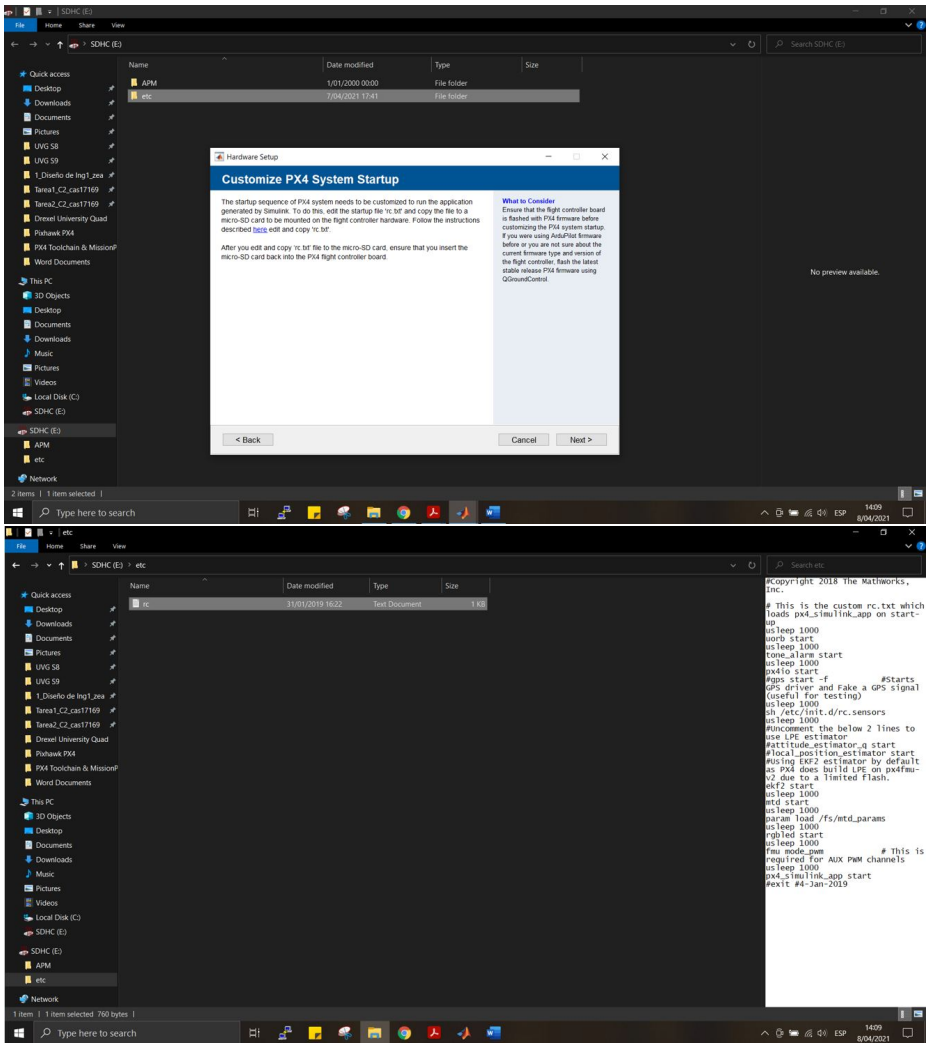


Figura 170: Paso 8 - Archivo rc.txt copiado correctamente en memoria SD.

13.1.9. Paso 9: Verificación de la conexión

En este último paso se tuvo la complicación que en el momento de seleccionar el puerto COM7 el cual era el mismo que se visualizó en el administrador de dispositivos la configuración en esta parte se quedaba cargando sin una respuesta adecuada (nunca finalizaba).

Probando lo que sugería la página de Mathworks, se procedió a desconectar y reconectar el controlador de nuevo y se notó que el Puerto COM7, no era el adecuado ya que por lo mismo del archivo de arranque (startup rc.txt) este puerto cambia para iniciar con otro no asignado (el COM7 era para el Mission Planner) es por esto que se solicita reconectar varias veces el controlador hasta que ese puerto se actualice a uno nuevo que en este caso fue el COM6. Es por esto que en al terminar la configuración del Mission Planner sale el mismo puerto COM7 como se puede ver en la Figura 157.

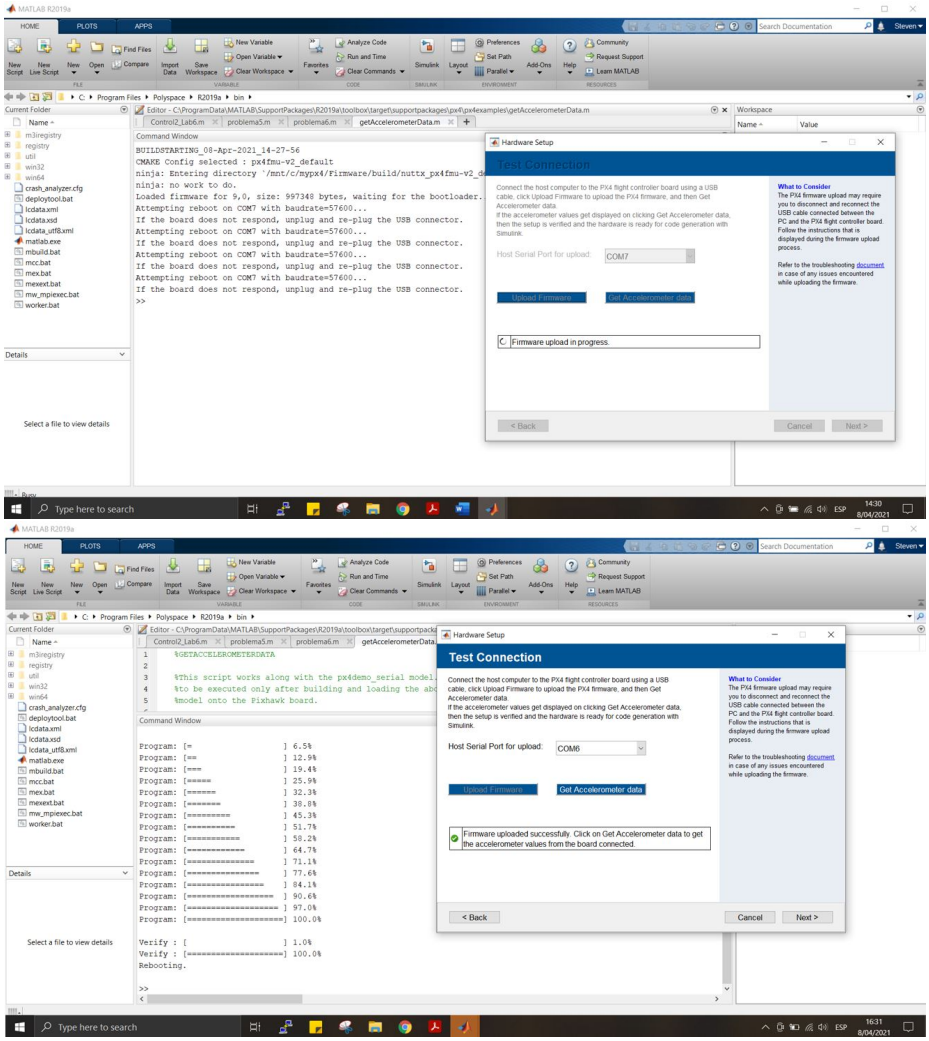


Figura 171: Paso 9 - Verificación de conexión Matlab/Pixhak.

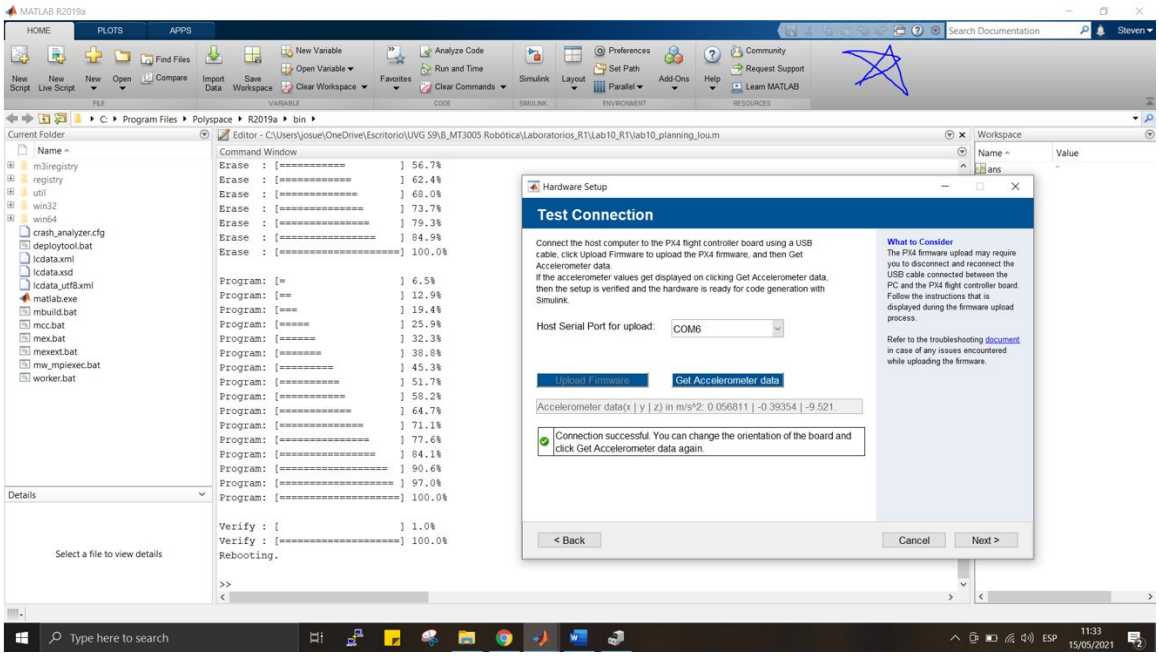


Figura 172: Paso 8 - Medición de prueba data/acelerómetro.

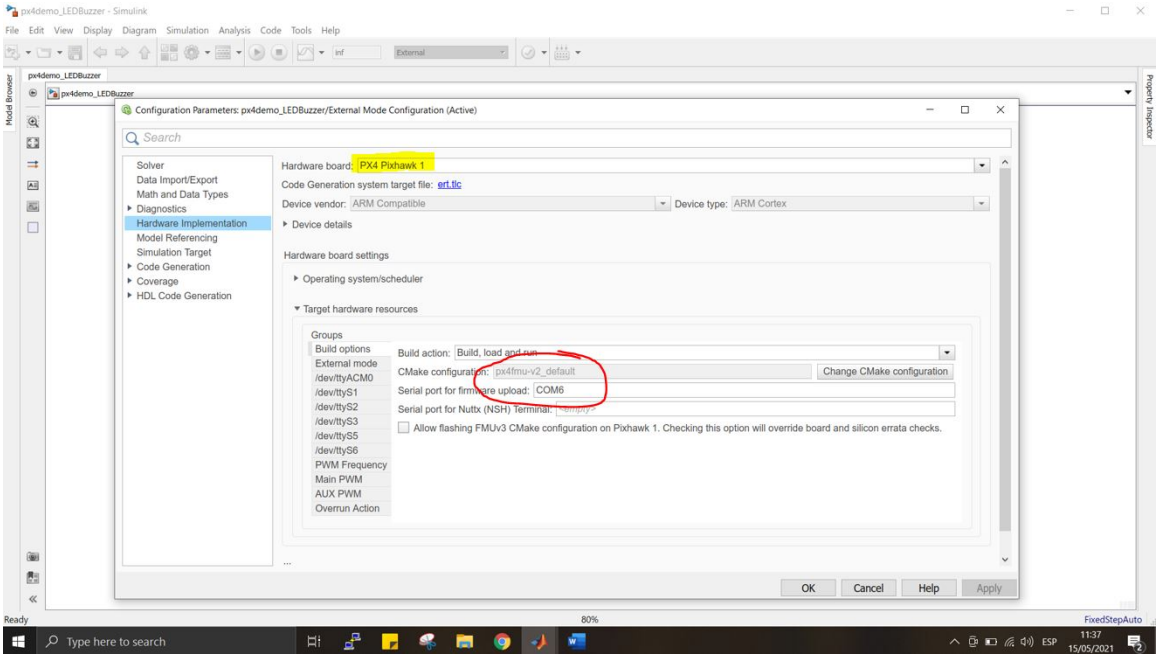


Figura 173: Paso 8 - Configuración puerto COM6 desde Simulink.

13.2. Exportar CAD (contorno del PCB) de Inventor a Altium

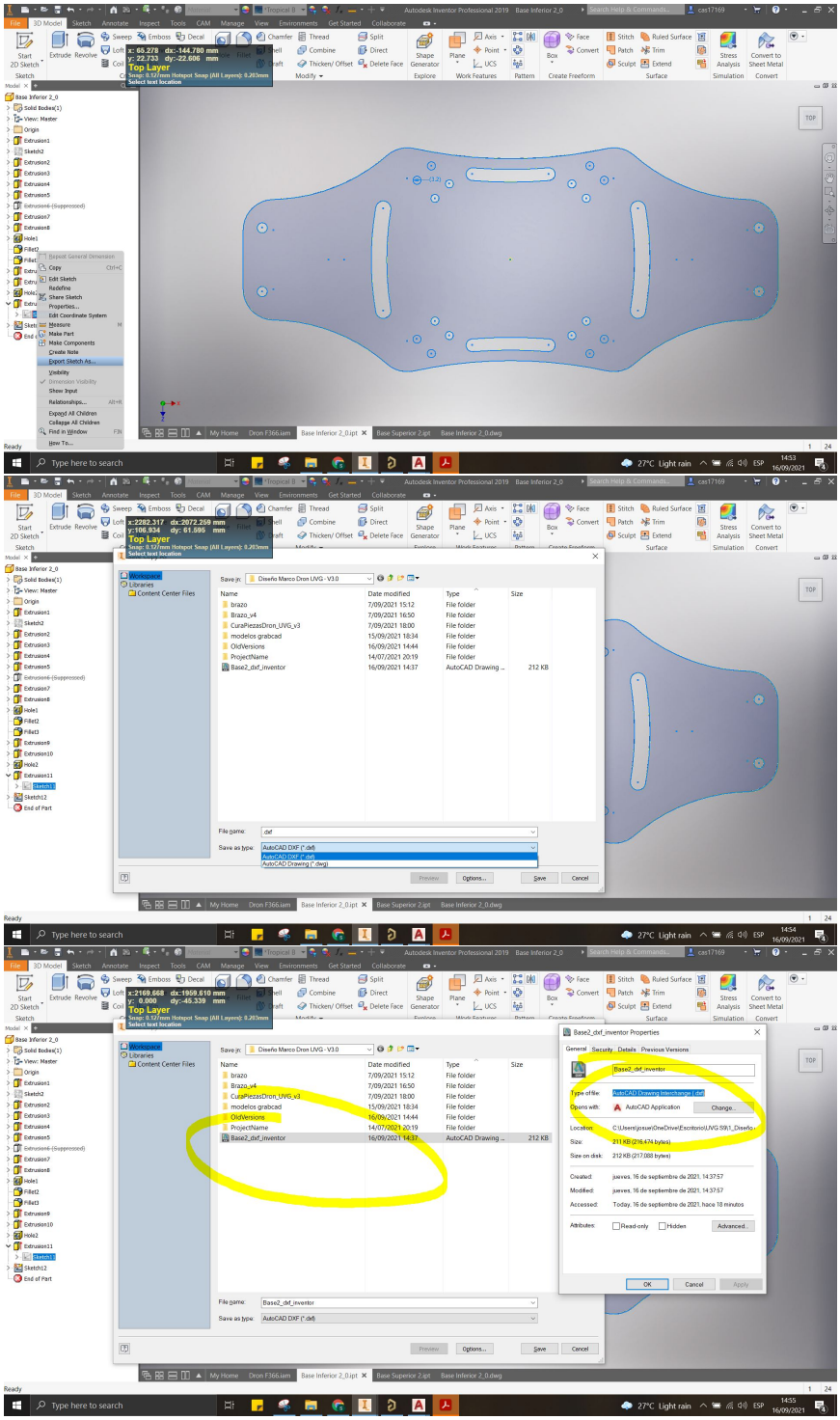


Figura 174: Paso 1 - Exportar CAD de Inventor a Altium.

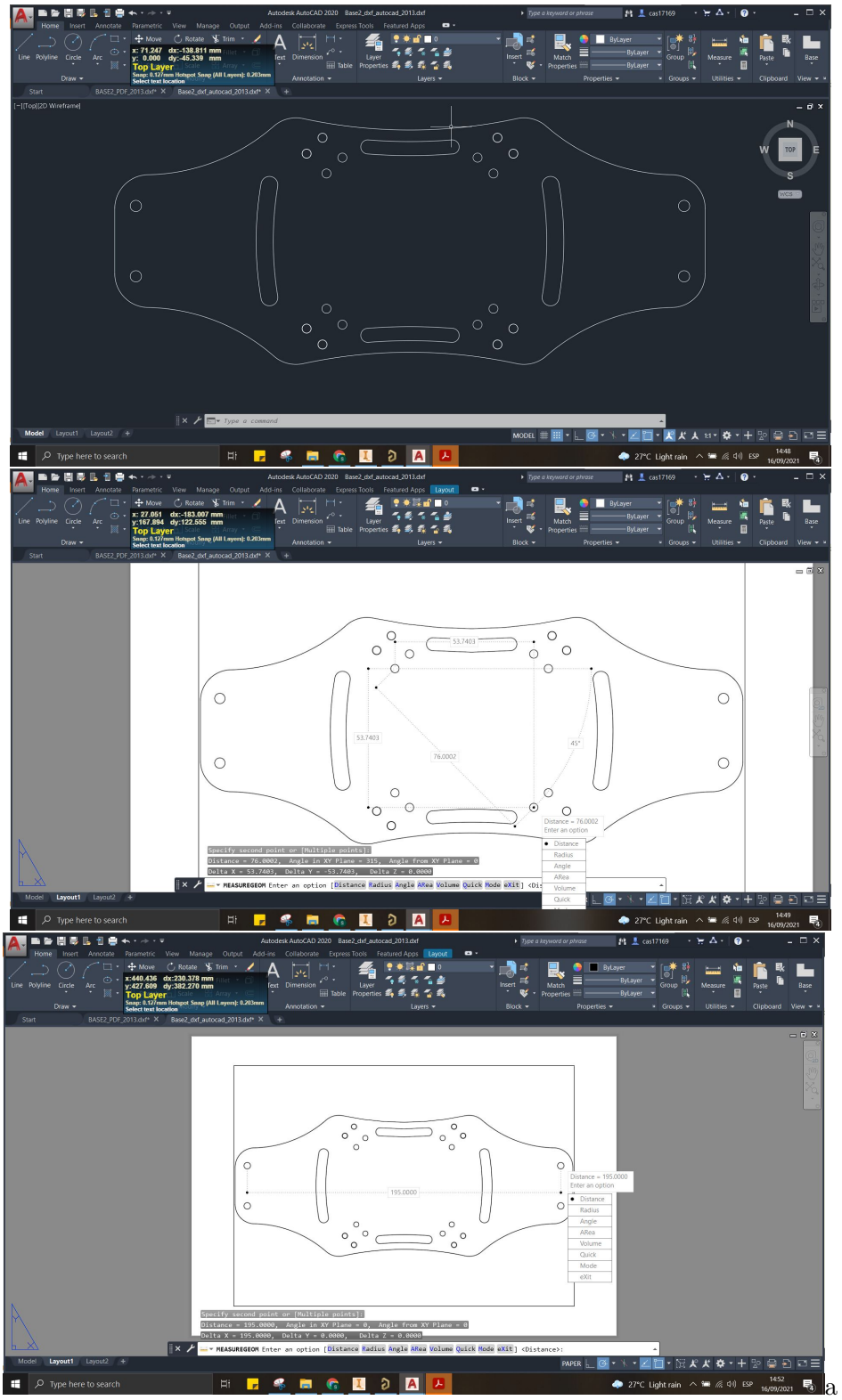


Figura 175: Paso 2 y 3 - Exportar CAD de Inventor a Altium.

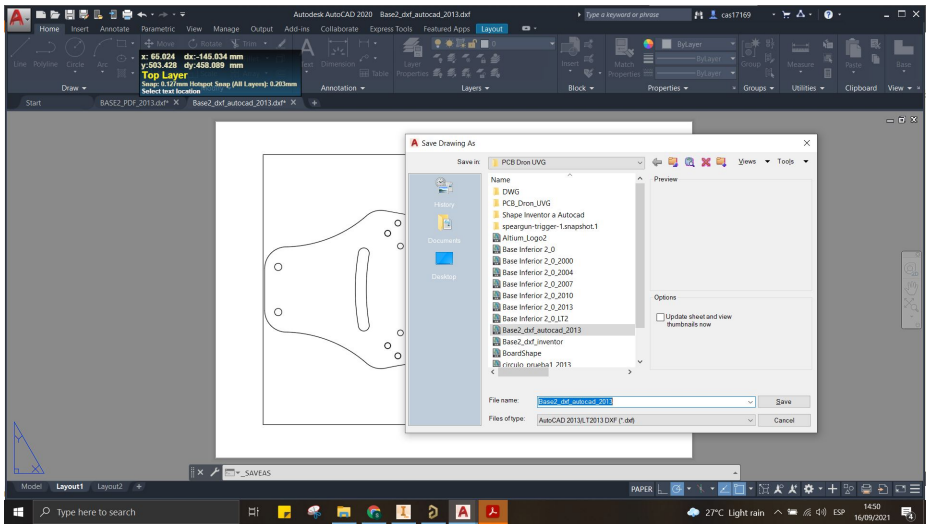


Figura 176: Paso 4 - Exportar CAD de Inventor a Altium.

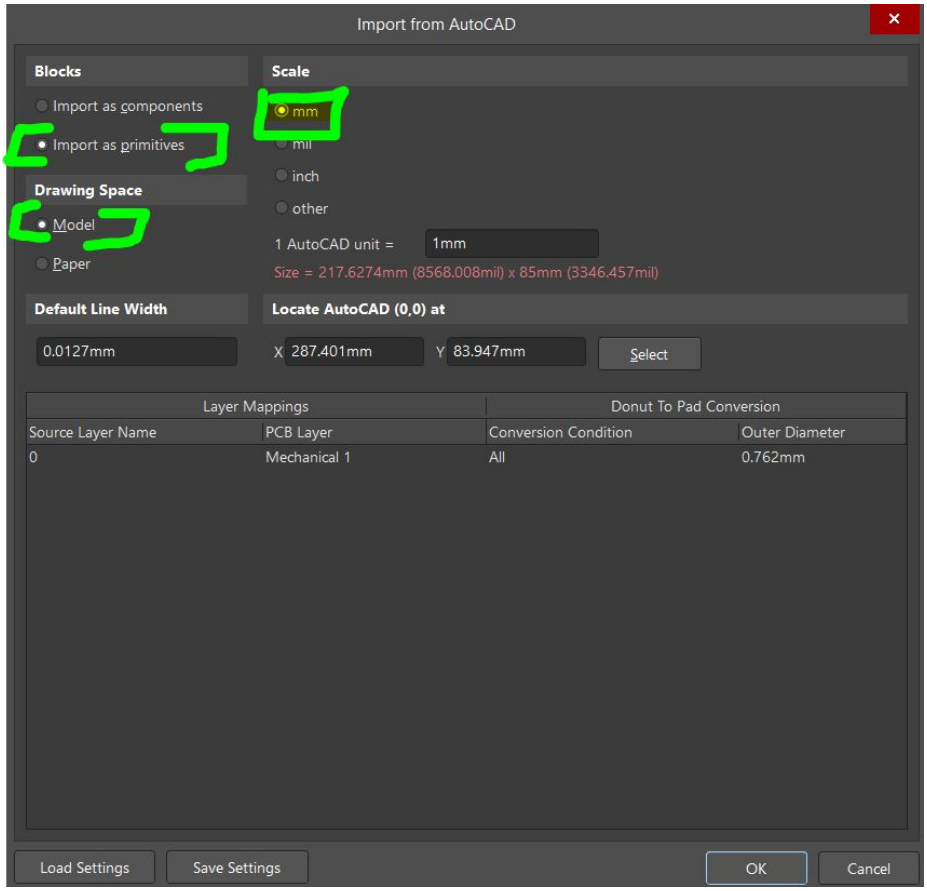


Figura 177: Paso 5 y 6 - Exportar CAD de Inventor a Altium.

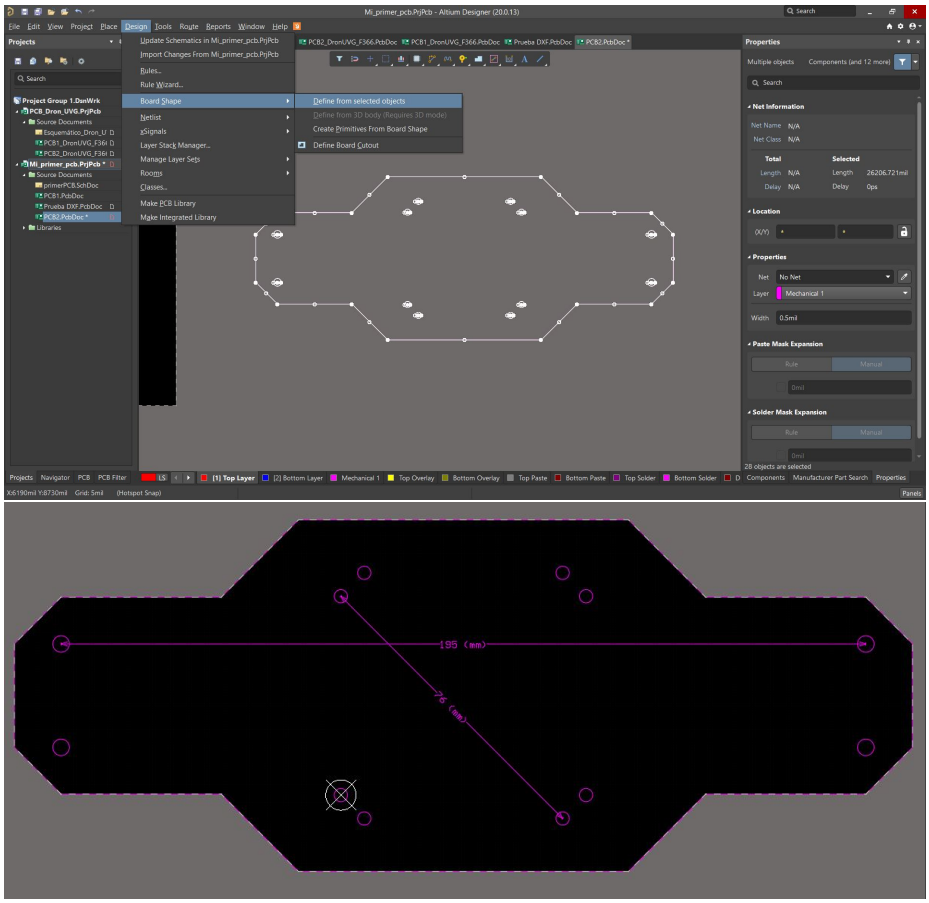


Figura 178: Paso 7 - Exportar CAD de Inventor a Altium.

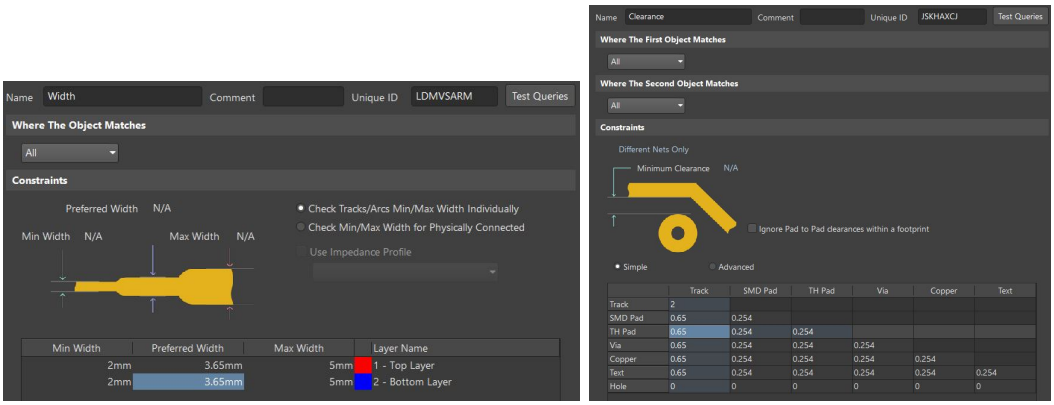


Figura 179: Reglas de diseño - Altium PCB.

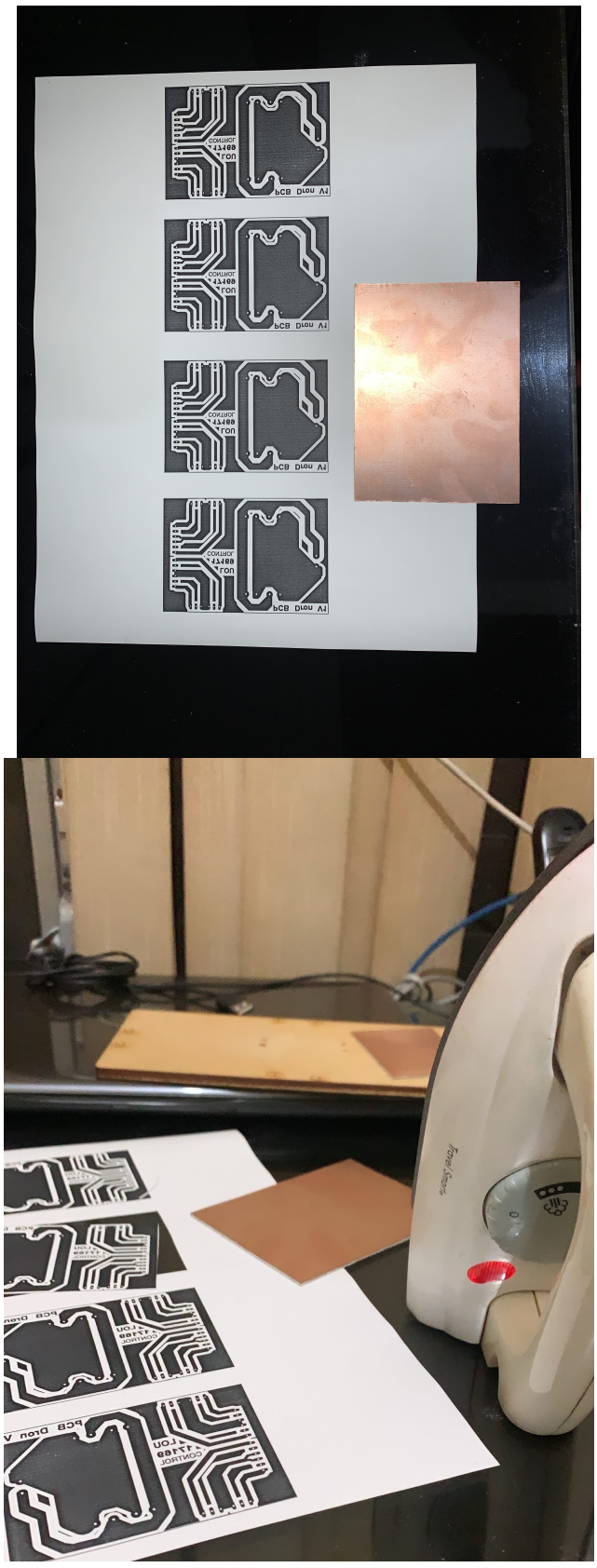


Figura 180: Placa versión 1 (manufactura).