

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

### **DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA MULTI – ROTOR DE TRES GRADOS DE LIBERTAD COMANDADA POR MEDIO DE GESTOS REALIZADOS CON LA MANO**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y CONTROL**

**RICARDO ANDRÉS ROMERO PAREDES**

**[ricardorompar@hotmail.com](mailto:ricardorompar@hotmail.com)**

**DIRECTOR: Ing. PATRICIO JAVIER CRUZ DÁVALOS, Ph.D.**

**[patricio.cruz@epn.edu.ec](mailto:patricio.cruz@epn.edu.ec)**

**Quito, febrero 2022**

## **AVAL**

Certifico que el presente trabajo fue desarrollado por Ricardo Andrés Romero Paredes, bajo mi supervisión.

---

**Ing. Patricio Javier Cruz Dávalos, Ph.D.**  
**DIRECTOR DEL TRABAJO DE TITULACIÓN**

## DECLARACIÓN DE AUTORÍA

Yo Ricardo Andrés Romero Paredes, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.



---

Ricardo Andrés Romero Paredes

## DEDICATORIA

*Todo el esfuerzo y tiempo destinados a la elaboración de este trabajo van dedicados principalmente a mi familia: mis papás y mi hermano, a quienes les debo todo lo que soy y han sido mi inspiración, mi guía, mi apoyo y motivación en cada momento de mi vida. Incluido, sobre todo, este.*

*A mis amigos, que son la familia que uno escoge.*

*De manera especial le dedico este trabajo a la futura versión del autor. Confío en que logrará todo lo que se proponga, como lo está demostrando ahora mismo.*

*A mi abuelita, que esperaba con ansias el día que me gradúe. Sé que está conmigo.*

*Finalmente, le dedico este trabajo al niño curioso que habita en mi interior y en el de cada uno de nosotros; que sigan soñando, aprendiendo y eventualmente, creando el mundo en el que todos viviremos.*

## AGRADECIMIENTO

En primer lugar, quiero agradecerles a mis padres Luis y Sonia que son la razón de todo lo que soy. A mi mamá, que es un ángel en la tierra y con su ejemplo de infinito amor, cariño y esfuerzo me ha guiado a este momento y lo sigue haciendo. A mi papá, que es un verdadero ingeniero. Su ejemplo de trabajo, perseverancia y sabiduría son los cimientos sobre los que mis pies andan. A mi hermano, el Luigi, a quien siempre he visto con profunda admiración y es, en gran medida, la razón por la que realicé este proyecto. Gracias *brow*. Gracias infinitas a los tres. Les amo.

A mis amigos, sin quienes la elaboración de este trabajo habría sido absolutamente imposible. De manera especial le agradezco al Lucho, al Perro, al Mario, a los Gabrieles, Pablin, Nohe, Gustavo, etc. (realmente es una lista muy larga). A veces me cuesta entender cómo me he llegado a encontrar con tantas personas tan excepcionales. Gracias a todos.

Le agradezco a la Cris por apoyarme y celebrar conmigo cada logro conseguido en toda esta etapa, además de brindarme esa paz y entendimiento que solo ella sabe dar.

A mis primos, especialmente la Verito y el Andy que no dejaron de preguntar cómo iba la tesis, hasta llegar al borde del cansancio. Sin ellos estas páginas se habrían escrito unos cuantos meses más tarde.

De manera especial agradezco al Ing. Patricio J. Cruz Ph.D., por su guía, apoyo y paciencia. Difícilmente habría podido escoger a un mejor director de proyecto. De igual manera agradezco la ayuda del Dr. Juan Pablo Vásquez por tomarse el tiempo de reunirse innumerables martes y ayudar a mejorar este proyecto con sus consejos. Agradezco también a todos quienes conforman el Laboratorio de Inteligencia y Visión Artificial por sus consejos y palabras de apoyo.

Agradezco a la Escuela Politécnica Nacional por ser un segundo hogar durante tantos años. Además, por el apoyo financiero proporcionado para el desarrollo del proyecto PIGR-19-07 y, especialmente, del presente trabajo de titulación.

Le agradezco al de Arriba, porque para hacer una tesis primero debes crear el universo.

Finalmente, le agradezco al Ricky, por no haberse conformado con lo fácil.

# ÍNDICE DE CONTENIDO

AVAL.....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VIII
ABSTRACT.....	IX
1. INTRODUCCIÓN.....	1
1.1. OBJETIVOS.....	1
1.2. ALCANCE .....	2
1.3. MARCO TEÓRICO .....	3
1.3.1. PLATAFORMAS MULTI-ROTOR DE TRES GRADOS DE LIBERTAD..	5
1.3.1.1. Grados de libertad (GDL) .....	5
1.3.1.2. Introducción a los UAVs multi-rotor.....	6
1.3.1.3 Tipos de plataformas multi-rotor de 3-GDL.....	9
1.3.2. HELICÓPTERO DE 3-GDL DE ROTORES EN TÁNDEM.....	11
1.3.3. CONTROL DEL HELICÓPTERO DE 3-GDL.....	13
1.3.3.1 Control PID.....	13
1.3.3.2 Lazo de control en cascada.....	15
1.3.4. INTRODUCCIÓN A LOS SISTEMAS DE RECONOCIMIENTO DE GESTOS DE LA MANO.....	16
1.3.4.1 Clasificación de los sistemas HGR .....	16
1.3.4.2 Características de las señales electromiográficas. ....	19
1.3.4.3 Myo Armband .....	20
1.3.4.4 Sistema de reconocimiento de gestos de la mano del Proyecto de Investigación Grupal PIGR-19-07.....	22
2. METODOLOGÍA.....	27
2.1. ARQUITECTURA DEL SISTEMA .....	27
2.2. IMPLEMENTACIÓN DEL HELICÓPTERO DE 3-GDL .....	29
2.2.1. HARDWARE DE CONTROL ELECTRÓNICO .....	29
2.2.1.1. Elementos de realimentación y control .....	30
2.2.1.2. Comunicación.....	44

2.2.1.3. Alimentación .....	49
2.2.1.4. Diseño de tarjetas electrónicas .....	51
2.2.2. CONSTRUCCIÓN DEL PROTOTIPO MECÁNICO .....	56
2.2.2.1. Cuerpo del helicóptero.....	57
2.2.2.2. Eje principal .....	60
2.2.2.3. Base .....	63
2.3. SISTEMA DE CONTROL .....	66
2.3.1. MODELO MATEMÁTICO DEL HELICÓPTERO DE 3-GDL .....	67
2.3.1.1. Variables del modelo .....	67
2.3.1.2. Parámetros del sistema .....	69
2.3.1.3. Modelo matemático .....	74
2.3.2. DISEÑO DE CONTROLADORES .....	85
2.3.2.1. Controlador Pitch .....	85
2.3.2.2. Controlador Elevation .....	86
2.3.2.3. Controlador Travel .....	87
2.3.2.4. Discretización de Controladores .....	88
2.4. INTEGRACIÓN CON EL SISTEMA DE RECONOCIMIENTO DE GESTOS.....	89
2.4.1. FUNCIONAMIENTO DEL SISTEMA DE HGR .....	90
2.4.1.1. Utilización del sistema de reconocimiento .....	90
2.4.2. INTEGRACIÓN DE COMPONENTES .....	91
2.4.2.1. Instancia 1 de Matlab.....	92
2.4.2.2. Instancia 2 de Matlab.....	93
2.4.2.3. Conexión entre las dos instancias de Matlab .....	97
2.4.2.4. Conexión entre Matlab y Simulink .....	97
2.4.3. CONTROL DE MOVIMIENTO MULTIMODO .....	98
2.4.3.1. Modo Manual.....	99
2.4.3.2. Modo Myo.....	99
3. RESULTADOS Y DISCUSIÓN .....	104
3.1. PROTOTIPO CONSTRUIDO .....	104
3.1.1. RESUMEN DE COSTOS .....	105
3.2. PRUEBAS DE CONTROLADORES.....	107
3.2.1. CONTROLADOR PITCH .....	107
3.2.1.1. Respuesta transitoria.....	108
3.2.1.2. Rechazo a perturbaciones .....	108

3.2.2. CONTROLADOR ELEVATION .....	110
3.2.2.1. Respuesta transitoria.....	110
3.2.2.2. Rechazo a perturbaciones .....	111
3.3.3. CONTROLADOR TRAVEL .....	112
3.3.3.1. Controlador PID de Travel sintonizado mediante el método de Ziegler-Nichols .....	114
3.3.3.2. Modificaciones al controlador Travel: algoritmo de anti wind-up y reset .....	117
3.3.3.3. Rechazo a perturbaciones .....	119
3.3. PRUEBAS DE COMANDOS MULTIMODO .....	120
3.3.1. MODO MYO .....	121
3.3.1.1. Modo Gesto .....	121
3.3.1.2. Modo IMU.....	123
3.3.2. MODO MANUAL.....	125
3.4. PUBLICACIÓN Y VIDEO .....	126
4. CONCLUSIONES Y RECOMENDACIONES .....	127
4.1. CONCLUSIONES .....	127
4.2. RECOMENDACIONES .....	129
5. REFERENCIAS BIBLIOGRÁFICAS.....	130
ANEXOS.....	135
ANEXO A.....	136
A.1. PLACA PRINCIPAL.....	136
A.2. PLACA AUXILIAR .....	141
ANEXO B.....	144
B.1. CUERPO DEL HELICÓPTERO .....	144
B.2. EJE PRINCIPAL .....	149
B.3. BASE.....	155
B.4. HELICÓPTERO COMPLETO .....	161
ANEXO C.....	162
ANEXO D.....	178
ANEXO E.....	188

## RESUMEN

El presente proyecto detalla la aplicación del reconocimiento de gestos de la mano y movimientos del brazo para controlar un banco de pruebas basado en rotores duales. En específico, se ha desarrollado un método de control multimodo basado en sensores de electromiografía superficial y mediciones inerciales (IMU) incluidos en el sensor Myo Armband para un helicóptero tándem de 3 grados de libertad (GDL). El sistema de reconocimiento puede clasificar cinco diferentes gestos de la mano, los cuales se utilizan para cambiar entre los tipos de vuelo y para generar diferentes valores de referencia para manipular la plataforma. El banco de pruebas del helicóptero de 3-GDL se ha elaborado e implementado completamente como una alternativa de bajo costo para evaluar la efectividad de controles de vuelo para vehículos aéreos no tripulados basados en rotores. La posición del helicóptero se regula mediante un esquema de control en cascada con un PID dual, el cual permite un cambio rápido entre los modos de gesto e IMU. Los resultados experimentales muestran la efectividad del uso de reconocimiento de gestos de la mano y movimientos del brazo para controlar la plataforma multi-rotor con una respuesta rápida y exacta.

**PALABRAS CLAVE:** Reconocimiento de Gestos de la Mano, Helicóptero Tándem de 3-GDL, Control Multimodo, Myo Armband.

## **ABSTRACT**

This project presents the application of the recognition of hand gestures and arm movements to control a dual rotor testbench. In particular, a multimodal control method is developed for a 3-degrees-of-freedom (DOF) tandem helicopter based on surface electromyography sensors and an inertial measurement unit (IMU) included in the Myo Armband sensor. The recognition system can classify five different hand gestures which are used for switching between flight modes and generating set point values for the helicopter. The 3-DOF helicopter testbench is fully designed and implemented as a low-cost alternative for testing the effectiveness of flight controls for unmanned aerial vehicles. The position of the helicopter is regulated by a cascade-dual-PID control scheme that allows a fast switching between a gesture mode and an IMU mode. Experimental results show the effectiveness of using hand gesture recognition and arm movement for controlling the multi-rotor testbench with a fast and accurate response.

**KEYWORDS:** Hand Gesture Recognition, 3-DOF Tandem Helicopter, Multimodal Control, Myo Armband.

# 1. INTRODUCCIÓN

El uso de vehículos aéreos no tripulados (UAVs) ha experimentado un crecimiento muy notorio debido a la versatilidad y la gran cantidad de aplicaciones que ofrecen; entre estas se pueden mencionar: comercio, seguridad, vigilancia, etc. Debido a que los usos para estos vehículos son cada vez más extensos, los sistemas para interactuar con estos robots y maniobrarlos deben ser más intuitivos y amigables para el usuario. Por lo general, las interfaces mediante las cuales interactúan humano y máquina son rudimentarias, como botones, palancas, perillas y demás, por lo que su uso puede resultar poco intuitivo. Por esta razón se han desarrollado nuevos sistemas que faciliten esta interacción, los cuales se basan en las cualidades naturales del ser humano como por ejemplo los gestos de la mano.

La elaboración de interfaces humano-máquina (HMIs) basados en gestos de la mano es un tema que actualmente es activamente investigado. Se han utilizado diversas tecnologías de adquisición de datos entre las cuales destaca la técnica de visión artificial y otras como sistemas infrarrojos y hasta guantes sensorizados. Sin embargo, en el contexto del presente proyecto, la técnica de mayor interés es la de electromiografía de superficie (sEMG), la cual permite hacer reconocimiento de gestos en base a la detección de actividad muscular. De manera específica, en el proyecto de investigación PIGR-19-07 se ha desarrollado un sistema de reconocimiento de gestos de la mano basado en sEMG e inteligencia artificial con el sensor Myo Armband. El proyecto presentado a continuación demuestra el uso de este sistema aplicado al control de una plataforma aérea representada por un prototipo de helicóptero de 3 grados de libertad (GDL).

En el presente Proyecto Técnico se expone la elaboración de dicho prototipo y cada una de sus partes, así como su integración con el sistema de reconocimiento de gestos de la mano. Los gestos realizados sirven como comandos de vuelo que permiten controlar el movimiento del helicóptero de 3-GDL.

## 1.1. OBJETIVOS

### OBJETIVO GENERAL

Diseñar e implementar una plataforma multi-rotor de tres grados de libertad comandada por medio de un sistema de detección de gestos de la mano basado en EMGs.

## OBJETIVOS ESPECÍFICOS

- Recopilar y revisar bibliografía relacionada con plataformas de tres grados de libertad (3-GDL) basadas en helicópteros de rotores en tándem sostenidos por un brazo fijo; y del sistema de reconocimiento de gestos de la mano (HGR) basado en el sensor Myo Armband desarrollado dentro del Proyecto de Investigación PIGR-19-07.
- Ensamblar las piezas mecánicas de la plataforma de 3-GDL elaboradas y construidas con ayuda de un software CAD.
- Diseñar e implementar el sistema electrónico de sensado y control de pose de la plataforma de 3-GDL en base a un microcontrolador Pyboard que interactuará con una interfaz gráfica desarrollada en MATLAB.
- Integrar el sistema HGR basado en el sensor Myo Armband con el control de posición de la plataforma de 3 GDL.
- Verificar el funcionamiento de la plataforma, el sistema de control y su integración con el sistema de reconocimiento de gestos.

## 1.2. ALCANCE

El alcance del presente proyecto abarca los siguientes puntos:

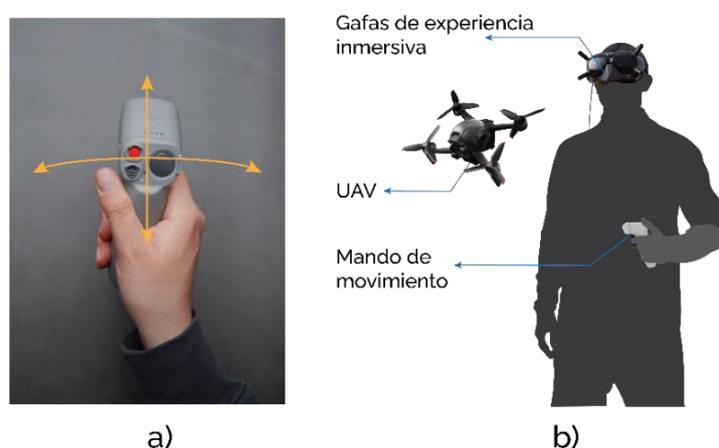
- Se hace una revisión bibliográfica correspondiente a plataformas de 3 GDL que emplean rotores, haciendo énfasis a aquellos basados en helicópteros de rotores en tándem sostenidos por un brazo fijo. Esto se realiza para conocer sus partes constituyentes, modelo matemático y técnicas de control.
- Se recopila información correspondiente a los sistemas de reconocimiento de gestos de la mano, especialmente los sistemas basados en señales EMG, y de manera particular del módulo de reconocimiento de gestos de la mano elaborado en el proyecto PIGR 19-07. Esto para conocer sobre su funcionamiento, la adquisición de señales con el sensor Myo, el procesamiento de las mismas y su uso en Matlab.
- Se construyen las piezas necesarias con ayuda del software CAD Fusion 360 para ensamblar la plataforma basada en el modelo comercial de la marca Quanser [6]. La elaboración de las piezas se realiza por medio de impresión 3D, corte láser y torneado.

- Se diseña el hardware electrónico y PCB para la implementación del sistema de control embebido y sensado para regulación de posición de la plataforma.
- Se diseña e implementa un esquema de control embebido necesario para la regulación y control de pose de la plataforma. Esto se realiza en el microcontrolador Pyboard, ya que posee los periféricos necesarios para el sensado y comunicación.
- Se determinan las acciones que la plataforma deberá realizar en función de los gestos reconocidos por el sistema de HGR. Esto significa otorgar una operación específica para los cinco gestos de la mano (wave out, wave in, fist, open y pinch) así como la utilización de la unidad inercial IMU.
- Una vez que se tienen el algoritmo de control implementado en la plataforma y las acciones que se realizan con cada gesto, se integra con el sistema de HGR. Los gestos y movimientos servirán para generar una señal de referencia para el controlador de posición del modelo de 3 GDL.
- Se realiza una interfaz gráfica en MATLAB que permita monitorear las variables de la plataforma y la respuesta ante los gestos realizados.
- Se comprueba el seguimiento de referencias de pose de la plataforma de 3GDL utilizando el sistema de HGR y también sin utilizarlo, es decir, generando las referencias directamente desde la interfaz gráfica desarrollada. En el contexto del proyecto, la pose hace referencia al valor de los tres grados de libertad de libertad que corresponden a los ángulos: travel, elevation y pitch.

### **1.3. MARCO TEÓRICO**

En los últimos años, el uso de vehículos aéreos no tripulados o UAVs (por sus siglas en inglés) ha experimentado un auge debido a la gran versatilidad que presentan; su utilización abarca desde aplicaciones militares hasta comerciales como vigilancia y transporte de bienes. Actualmente existen dos tipos de UAVs: autónomos y remotamente guiados. Los primeros se usan para aplicaciones como vigilancia y cuentan con sistemas internos de posicionamiento y control que comandan su movimiento de forma independiente. Por otro lado, los remotamente pilotados necesitan de un usuario que controle su movimiento, para lo cual se utilizan mandos a distancia cuyo uso suele ser complejo y poco intuitivo para el usuario general.

No obstante, el uso de nuevas tecnologías, como gafas de experiencia inmersiva, dan paso a una interfaz más intuitiva y cómoda entre el usuario y el UAV. Como ejemplo se puede mencionar el kit FPV (first person view) de la marca DJI [1]; este es un sistema que ofrece la posibilidad de controlar el vehículo aéreo al mismo tiempo que se visualiza en primera persona la imagen obtenida mediante cámaras de alta resolución a bordo del dron. Adicionalmente se puede utilizar el mando de movimiento que permite controlar el sistema mediante movimientos de la muñeca tal como se muestra en la Figura 1.1. Este tipo de sistemas de control de cuadricópteros mediante gafas inmersivas son muy utilizados en competencias de drones a nivel mundial. Esto se debe a que la vista de primera persona facilita el control del UAV porque el usuario no tiene que realizar un esfuerzo por determinar la posición del vehículo de manera visual, sino que puede posicionarse a partir de la imagen obtenida desde las cámaras colocadas en el dron [2].



**Figura 1.1.** DJI FPV con Motion Controller [1]: un UAV comandado mediante movimientos de la mano. a) Los movimientos de la muñeca son utilizados para comandar el dron. b) El sistema se compone de: mando de movimiento, gafas de experiencia inmersiva y el mando de movimiento.

Además de los UAVs remotamente controlados, se encuentran los autónomos. Estos vehículos son capaces de realizar tareas como vigilancia, reconocimiento, mapeo, transporte y demás, de manera independiente. En este caso, generalmente, no se cuenta con un usuario que pueda controlar el sistema o reaccionar ante una perturbación externa, por lo cual los sistemas de control deben ser efectivos y confiables. Por tal razón son necesarias plataformas de prueba en las que se pueda evaluar la efectividad de dichos controladores. En primera instancia, estos bancos de prueba podrían ser los mismos drones, sin embargo, resultan de gran utilidad las plataformas fijas (en tierra) ya que se

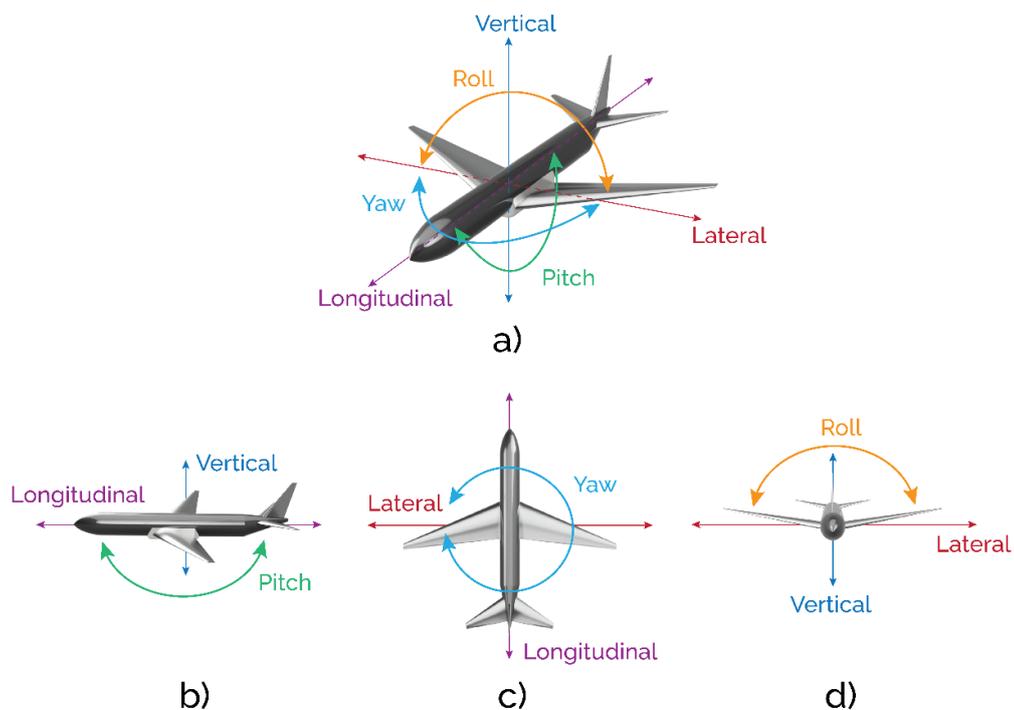
pueden hacer ajustes de manera rápida sin que eso represente un gran riesgo para el usuario o el equipo.

### 1.3.1. PLATAFORMAS MULTI-ROTOR DE TRES GRADOS DE LIBERTAD

El control de posición de UAVs autónomos basados en rotores es un problema que aún es activamente investigado. Cada vez los sistemas de control son más eficientes y para comprobar el funcionamiento de las diferentes estrategias se pueden utilizar plataformas de movilidad limitada con el objetivo de evaluar la efectividad de estos sistemas dentro de un ambiente controlado. Para esto se utilizan las plataformas multi-rotor de tres grados de libertad.

#### 1.3.1.1. Grados de libertad (GDL)

Los grados de libertad de un cuerpo se definen como el mínimo número de coordenadas independientes que son necesarias para describir completamente el estado del sistema mecánico [3]. Un vehículo aéreo, dado que es un sólido rígido en un espacio 3D, posee 6 grados de libertad que corresponden a las 3 direcciones de movimiento en el espacio más 3 ejes de rotación.



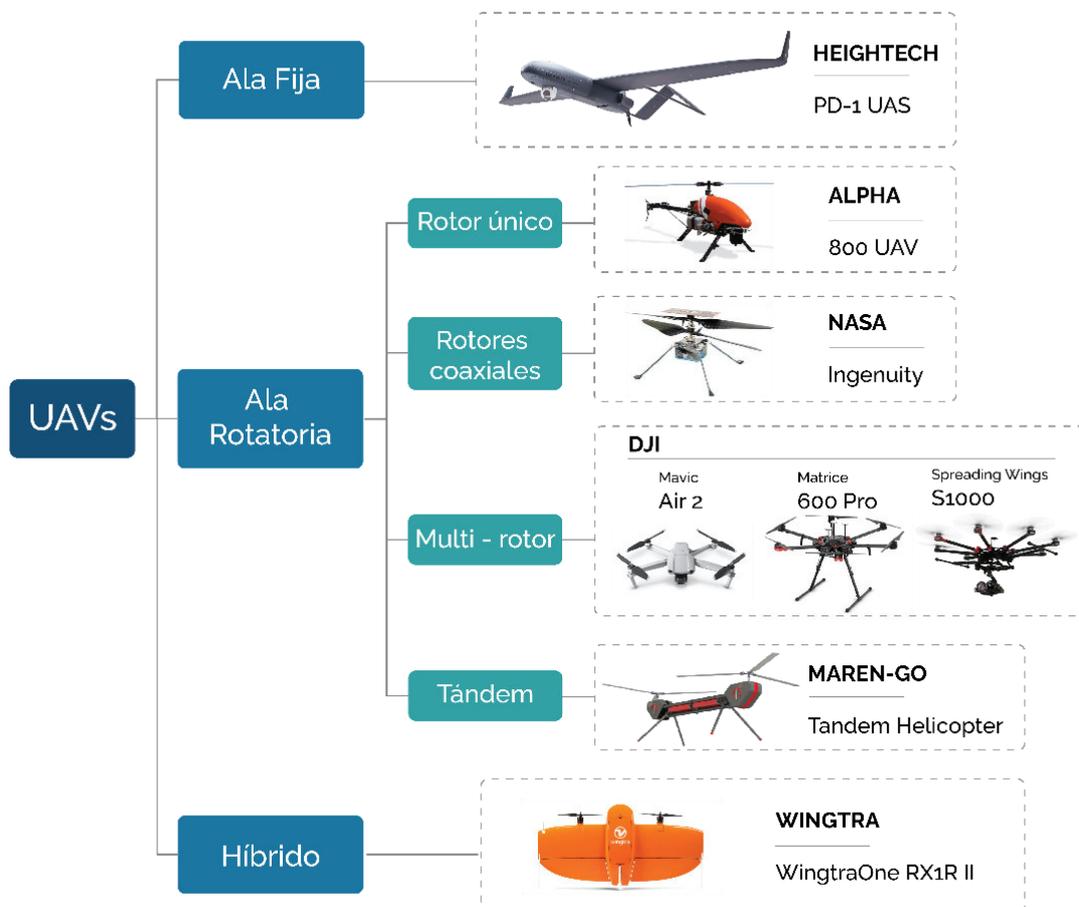
**Figura 1.2.** Grados de libertad de una aeronave. a) vista general con los 6 grados de libertad, b) vista lateral, c) vista superior, d) vista frontal.

Por otro lado, las plataformas de tres grados de libertad basadas en rotores son modelos de complejidad reducida y el propósito de su uso es probar la efectividad de los sistemas

de control. Dado que su movilidad es restringida, los grados de libertad se reducen a 3 que corresponden únicamente a los ejes de rotación. Como se puede observar en la Figura 1.2, para una aeronave convencional, estas posiciones angulares reciben el nombre de Pitch, Roll y Yaw. Estas variables se pueden controlar mediante diferentes actuadores dependiendo del tipo de aeronave que corresponda.

### 1.3.1.2. Introducción a los UAVs multi-rotor.

Al igual que las aeronaves tripuladas, los UAVs se clasifican en diferentes tipos que son: ala fija, ala rotatoria y un híbrido entre ambas, como se muestra en la Figura 1.3. La mayoría de las aeronaves motorizadas de ala fija generan su fuerza de propulsión en la dirección longitudinal como es el caso ejemplificado en la Figura 1.2. En este caso, el control de su orientación y movimiento se realiza por medio de las denominadas *superficies de mando*, que son alerones, timones, flaps, etc. Su funcionamiento se basa en el direccionamiento del flujo de aire dependiendo del ángulo de dichas superficies [4].



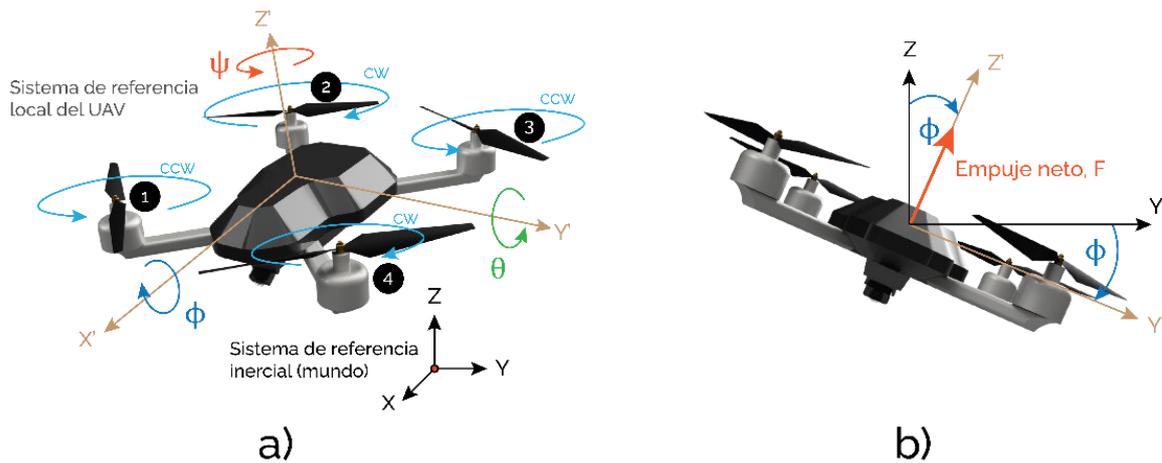
**Figura 1.3.** Clasificación de los vehículos aéreos no tripulados según su tipo.

Por otro lado, las aeronaves de ala rotatoria se clasifican en tres tipos que son autogiros, girodinós y helicópteros. Los dos primeros utilizan su ala rotatoria para generar la fuerza de empuje en la dirección vertical, mientras que la fuerza longitudinal se genera mediante un sistema de propulsión independiente [5]. En el caso de los helicópteros, las alas rotatorias se utilizan para generar el empuje en las tres direcciones de movimiento: vertical, longitudinal y transversal.

Para todas las aeronaves de ala rotatoria mencionadas anteriormente, el control de su movimiento se da mediante el direccionamiento del flujo de aire a través de las hélices, lo cual se logra variando su ángulo de incidencia [6]. Sin embargo, muchos UAVs de ala rotatoria poseen hélices rígidas sin partes móviles, este es el caso de las plataformas multi-rotor como cuadricópteros y hexacópteros. Para estas aeronaves, su posición y orientación es controlada mediante la variación de la velocidad de alguna o varias de sus hélices al mismo tiempo. Justamente los drones más utilizados son los de ala rotatoria debido a las numerosas funcionalidades que poseen. Uno de los ejemplos más famosos es el helicóptero Ingenuity de la NASA, que el 19 de abril de 2021 realizó su primer vuelo sobre la superficie de Marte a una altura de aproximadamente 3 metros [7].

Comparados con los vehículos de ala fija, los UAVs de ala rotatoria son los más ampliamente utilizados ya que presentan una mayor versatilidad, maniobrabilidad y facilidad de uso [8]. Esto se debe a que presentan algunas características que se mencionan a continuación:

- Despegue y aterrizaje vertical (VTOL por sus siglas en inglés): esto permite que la superficie de aterrizaje y despegue del UAV sea de dimensiones reducidas.
- Capacidad de permanecer suspendido (hover): esta cualidad le permite al vehículo realizar operaciones de vigilancia desde una posición fija.
- Costo reducido: los UAVs multi-rotor pueden llegar a costar hasta la mitad de un dron de ala fija con similares características [8].
- Mayor capacidad de carga: el uso de varios rotores permite levantar objetos de peso considerable, lo cual puede ser de gran utilidad para aplicaciones de reconocimiento con sensores especializados, transporte de bienes y productos, cinematografía, etc.
- Alta maniobrabilidad y facilidad de uso: el uso de multi-rotores permite controlar la dinámica del vehículo de manera relativamente sencilla. Adicionalmente, el amplio rango de tamaños de estos vehículos hace posible que se puedan realizar aplicaciones en lugares cerrados [9].



**Figura 1.4.** Movimiento de un cuadricóptero. a) la variación de las velocidades de las hélices produce una rotación en cada uno de los ejes roll  $\phi$ , pitch  $\theta$  o yaw  $\psi$ . b) Cuando se genera una variación en los ejes roll o pitch se produce una componente que modifica la posición del UAV.

El movimiento de este tipo de UAVs se controla mediante la variación de las velocidades en los rotores. Dado que cada rotor genera una fuerza independiente, el empuje neto  $F$  sobre el vehículo es la resultante de la suma de cada una de estas fuerzas, tal como se muestra en la Figura 1.4. Como se observa en el gráfico 1.4 a), cada rotor se puede identificar con un número del 1 al 4. La variación del ángulo Roll  $\phi$  se logra reduciendo la velocidad de los motores 1 y 2 y aumentando la de 3 y 4 o viceversa, mientras que para el ángulo Pitch  $\theta$  se consigue reduciendo la velocidad de los motores 1 y 4 y aumentando la de 2 y 3 o viceversa. Para el caso del ángulo Yaw  $\psi$  se debe considerar que el giro de cada hélice genera un torque contrario sobre el cuerpo del cuadricóptero. Por tal motivo es importante que un par de hélices gire en sentido horario (CW) y el otro en sentido antihorario (CCW). Dicho esto, la variación del ángulo  $\psi$  se consigue reduciendo la velocidad de los motores 2 y 4 y aumentando la de 1 y 3 o viceversa.

Como se muestra en la Figura 1.4. b), en el momento en que se produce una rotación sobre los ejes de Roll o Pitch, la dirección de la fuerza de empuje cambia y aparece una componente que hace acelerar al dron en las direcciones longitudinal  $Y$  o transversal  $X$  respectivamente. En el caso de producirse una variación en el eje Roll  $\phi$ , aparece una componente  $F \sin(\phi)$  en la dirección  $Y$  y por lo tanto el vehículo se acelera en esa dirección. De manera similar, se produce un movimiento en  $X$  con una variación en el ángulo  $\theta$ . Para variar la posición vertical o  $Z$  se aumenta la velocidad de todos los rotores, lo cual genera un mayor empuje y por lo tanto una aceleración en ese eje.

Con base a todo lo explicado anteriormente, se puede observar que el control del movimiento de un UAV se reduce a regular su posición angular en valores deseados de roll, pitch y yaw. Es decir, la variación de estos 3 grados de libertad determina el cambio de posición del vehículo en el espacio; es por esta razón, que los bancos de pruebas de 3-GDL son empleados para evaluar los sistemas de control en UAVs multi-rotor.

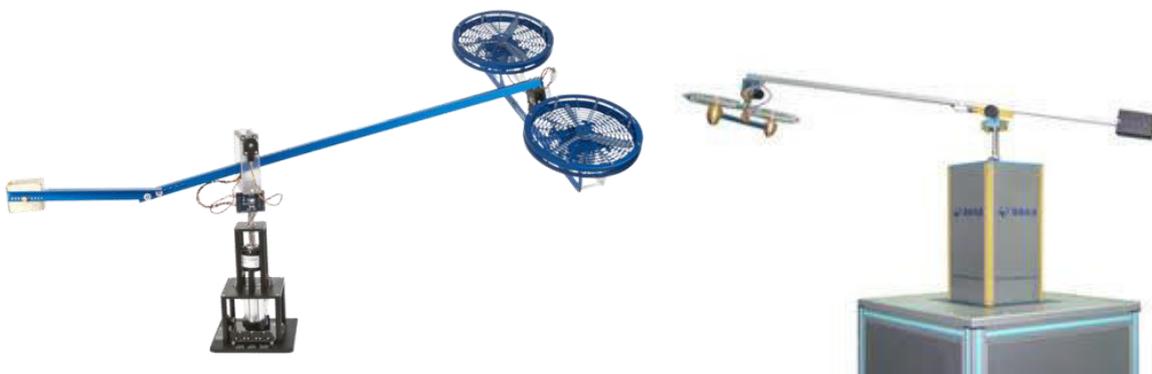
### 1.3.1.3 Tipos de plataformas multi-rotor de 3-GDL

A nivel comercial, las marcas que más destacan en la manufacturación de bancos de pruebas de 3 grados de libertad basados en rotores son Quanser [10] y Googol Technology [11]. Como se mencionó en la sección anterior, estas plataformas poseen movilidad reducida, por lo que solo se pueden modificar 3 de sus 6 variables. A continuación, se mencionan las plataformas más utilizadas.

- **3-DOF Helicopter**

Esta plataforma se basa en un modelo de un helicóptero con rotores en tándem, véase la Figura 1.5. Es utilizado para entender y desarrollar las leyes de control para un vehículo que posee la dinámica del cuerpo de un helicóptero rígido con rotores duales.

El cuerpo del helicóptero se encuentra suspendido con un rodamiento en el extremo de un brazo de soporte, lo que le permite realizar el movimiento de cabeceo (pitch). Este brazo en su parte media se encuentra sujeto a una articulación de dos grados de libertad, la cual le permite rotar alrededor del eje vertical y también moverse arriba y abajo. Finalmente, en el otro extremo del brazo se tiene un contrapeso que permite regular la masa efectiva del helicóptero, lo cual a su vez permite reducir el esfuerzo realizado por los rotores para generar el empuje necesario.



**Figura 1.5.** Helicópteros de 3-GDL. Izquierda: Quanser, derecha: Googoltech. Imágenes tomadas de [10] y [11].

Los ejes de rotación son comúnmente medidos con encoders, con lo cual se puede generar la realimentación de las posiciones angulares. Adicionalmente, se utiliza un anillo deslizante para transmitir datos y/o energía desde la base hasta la plataforma rotatoria, eliminando así la necesidad de tener cables que puedan enredarse. Los motores frontal y posterior son los encargados de todo el movimiento del helicóptero.

- **3-DOF Hover**

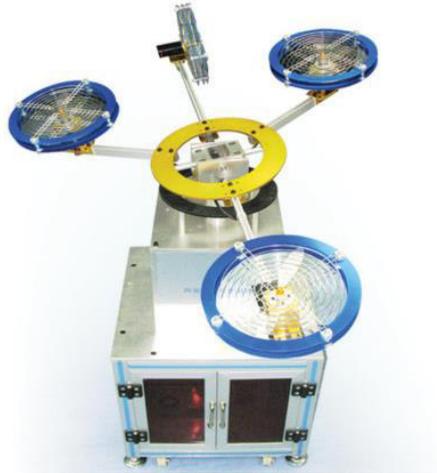
Esta plataforma se asemeja a un cuadricóptero con una articulación central para rotar libremente, véase la Figura 1.6. El sistema consiste de una estructura con cuatro hélices, se encuentra sostenido por una articulación de 3 grados de libertad que le permite girar alrededor de los ejes roll, pitch y yaw. Los rotores generan la fuerza de empuje necesaria para controlar los ángulos roll y pitch. Dos de las hélices giran en sentido contrario de tal forma que el torque neto aplicado sobre el cuerpo de la plataforma se encuentra balanceado.



**Figura 1.6.** Quanser 3DOF Hover: un cuadricóptero de 3-GDL. Imagen tomada de [12].

- **Four-Rotor Hover Vehicle**

Como su nombre lo indica, este sistema consta de 4 rotores, solo que, a diferencia del cuadricóptero, una de sus hélices gira sobre un plano diferente al de las otras 3, como se puede observar en la Figura 1.7. En este caso se tienen 3 motores que controlan el movimiento sobre los ejes roll y pitch, para lo cual se tiene una disposición en forma de Y con motores izquierdo, derecho y frontal. Finalmente, se tiene una hélice en la parte trasera del vehículo que sirve para contrarrestar el efecto del torque generado sobre el cuerpo de la plataforma. La velocidad de giro de este rotor además sirve para controlar el movimiento sobre el eje yaw.

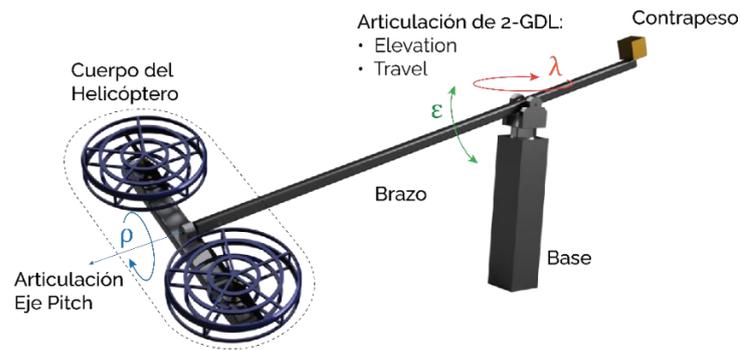


**Figura 1.7.** Vehículo suspendido de cuatro rotores de la marca Googoltech. Imagen tomada de [13].

### 1.3.2. HELICÓPTERO DE 3-GDL DE ROTORES EN TÁNDEM

El helicóptero de 3 grados de libertad es un banco de pruebas experimental para validar la efectividad de los sistemas de control en plataformas aéreas. Como su nombre lo indica, este sistema posee 3 variables que determinan su posición y movimiento. Esta plataforma está compuesta de 5 elementos fundamentales que se muestran en la Figura 1.8 y se explican a continuación.

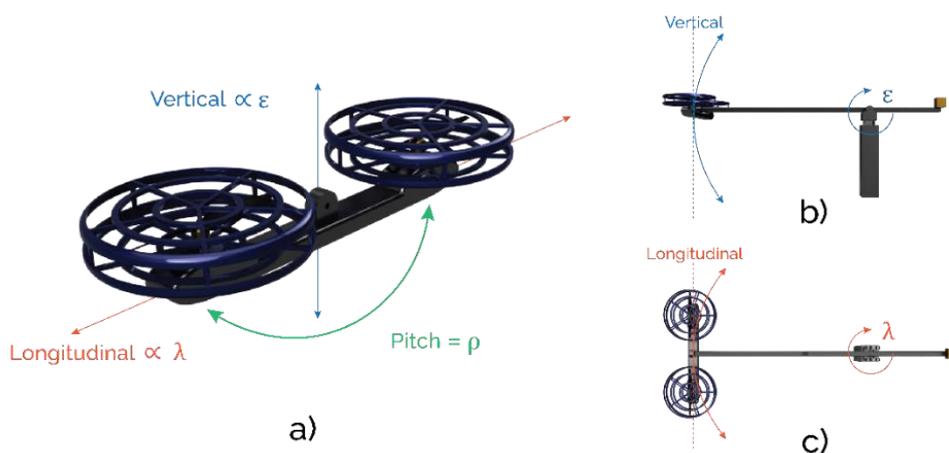
- Brazo de soporte: es la estructura que sostiene el contrapeso y el cuerpo del helicóptero.
- Cuerpo del helicóptero: se sitúa en un extremo del brazo de manera perpendicular. El helicóptero posee una configuración de rotores en tándem que se denominan frontal y posterior (o front y back en inglés, respectivamente).
- Contrapeso: en el otro extremo del brazo de soporte se encuentra un contrapeso que sirve para balancear el peso del cuerpo del helicóptero. Esto con el objetivo de reducir el esfuerzo generado por los motores para mover a la plataforma.
- Articulaciones: el sistema consiste de 3 articulaciones rotacionales que permiten los movimientos sobre sus 3 ejes de libertad denominados pitch, elevation y travel.
- Base: es la estructura que sostiene el brazo de soporte, la cual a su vez sostiene el cuerpo del helicóptero y el contrapeso. En la base se suele encontrar un anillo deslizante que conecta la parte rotacional con la parte inferior para evitar la utilización de cables que pueden enredarse.



**Figura 1.8.** Esquema del helicóptero de 3-GDL, sus partes constitutivas y sus tres grados de libertad: pitch  $\rho$ , elevation  $\epsilon$ , travel  $\lambda$ .

Como se mencionó en los puntos anteriores, los ejes de libertad de esta plataforma reciben los nombres de pitch, elevation y travel. Esto se debe a que estos nombres corresponden de manera más adecuada con el movimiento del cuerpo del helicóptero. El eje de travel se corresponde con el movimiento longitudinal del helicóptero, mientras que elevation se relaciona con el movimiento vertical.

Al igual que en el caso del cuadricóptero de la Figura 1.4, esta plataforma genera su movimiento longitudinal (travel) cuando existe una variación en el eje de pitch, mientras que su desplazamiento vertical (elevation) está relacionado con la magnitud de la fuerza de empuje. Como se observa en la Figura 1.9, el helicóptero puede desplazarse a lo largo del eje longitudinal, para lo cual se debe generar un ángulo de pitch, mientras que para el eje vertical se varía el empuje total generado por los rotores, que, como se explicará en el siguiente capítulo, recibe el nombre de  $F_{sum}$  (por la suma de las fuerzas generadas por cada hélice).



**Figura 1.9.** Relación entre travel  $\lambda$  y elevation  $\epsilon$  con los ejes longitudinal y vertical respectivamente. a) Vista del helicóptero aislado, b) Vista lateral, c) Vista superior.

Dado que el cuerpo del helicóptero se encuentra acoplado al brazo de la plataforma su movimiento está restringido a la longitud dada desde el extremo hasta el eje de rotación de travel y elevation, denominada  $L_m$ . Por esta razón, en realidad el eje longitudinal de la aeronave es tangencial a la trayectoria real que puede describir el helicóptero y de igual manera sucede con el eje vertical, tal como se muestra en las Figuras 1.9 b y c. En el Capítulo 2 se explica a mayor detalle el funcionamiento del helicóptero de 3-GDL elaborado en el presente proyecto.

### **1.3.3. CONTROL DEL HELICÓPTERO DE 3-GDL.**

Una vez revisada la estructura y funcionamiento básico del helicóptero de 3-GDL, se puede determinar una estrategia para controlar sus variables. Cabe recalcar que este sistema es de tipo sub-actuado, lo que quiere decir que no todas las variables pueden seguir trayectorias arbitrarias [14]. Esto se debe a que el número de actuadores en el sistema (2 rotores) es menor que el número de grados de libertad del mismo (3 variables: travel, elevation, pitch). Esta restricción en el sistema se manifiesta como un acoplamiento entre las variables travel y pitch, en particular, un movimiento en el eje pitch produce una variación en la variable travel y de manera correspondiente, la variable travel solo puede modificarse mediante un movimiento en pitch.

#### **1.3.3.1 Control PID**

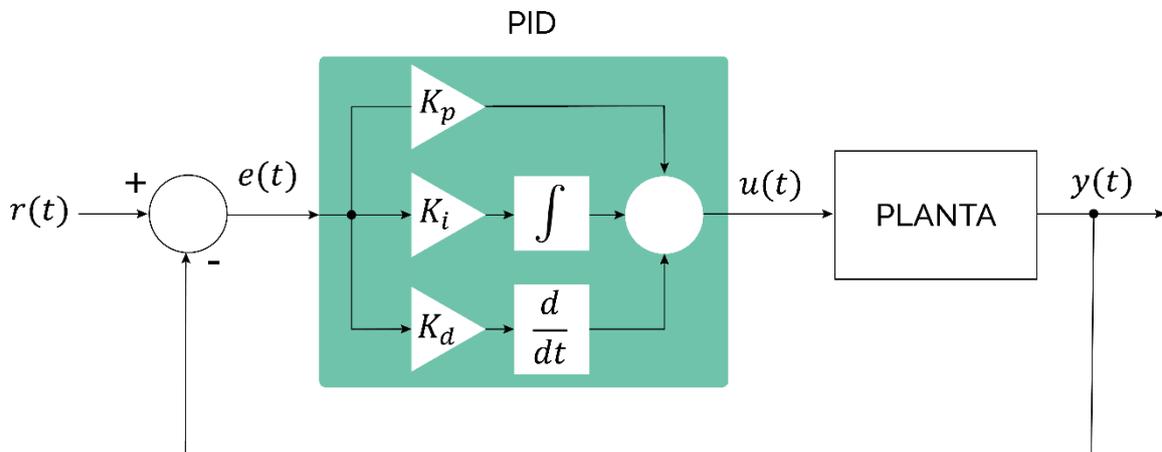
El controlador PID es la forma más común de realimentación. Desde su aparición ha sido un elemento esencial en aplicaciones de seguimiento y regulación y se convirtió en la herramienta estándar cuando apareció el control de procesos en la década de 1940. Actualmente, más del 90% de los lazos de control son de tipo PID de los cuales la mayoría tienen una estructura PI [15]. La popularidad de este esquema de control se debe a la gran versatilidad que ofrecen, pudiéndose implementar en la mayoría de los sistemas incluyendo lineales y no lineales, como es el caso de plataformas aéreas como cuadricópteros. En este caso, el controlador PID ofrece una solución simple pero efectiva para estabilizar al vehículo ya que hace posible considerar cada variable como independiente dentro de un rango limitado (ángulos pequeños), en el cual el comportamiento del cuadricóptero se aproxima a un sistema lineal [16].

Como se mostrará en el siguiente capítulo, el helicóptero de 3-GDL es un sistema no lineal, sin embargo, puede presentar características lineales para variaciones pequeñas de sus variables, es decir, alrededor del punto de operación. Por este motivo resulta viable la implementación de un sistema de control PID. La ley de control que describe de forma

matemática el funcionamiento de este tipo de controlador viene dada por la siguiente ecuación:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1.1)$$

Donde  $e(t)$  es el error dado por la diferencia entre  $r(t)$ , que es la variable de referencia (también llamada setpoint) y la variable medida del sistema representada por  $y(t)$ . La señal de control  $u(t)$  es una suma de tres términos: P (proporcional al error), I (proporcional a la integral del error) y el término D (proporcional a la derivada del error). Los parámetros de sintonización son la ganancia proporcional  $K_p$ , la integral  $K_i$  y la derivativa  $K_d$ . En la Figura 1.10 se muestra un lazo cerrado de control con un regulador de tipo PID y las variables mencionadas.



**Figura 1.10.** Representación en diagrama de bloques de un lazo cerrado de control con PID en forma paralela.

El efecto que produce cada uno de los términos P, I, y D es muy distinto y dependiendo de los valores de las constantes correspondientes se pueden modificar las características dinámicas del sistema hasta conseguir una respuesta deseada.

- Acción proporcional: al aumentar el error, esta acción aumenta, de modo que se intenta llevar al sistema al valor deseado de referencia. El error disminuye conforme se incrementa la constante proporcional  $K_p$  pero la tendencia a oscilar también aumentará.
- Acción integral: este término aumenta la acción de control cuando el error se mantiene durante un tiempo, es decir, mientras exista un valor de error distinto de 0, la acción integral seguirá aumentando. El efecto de añadir esta componente es que se elimina el error en estado estable, una acción proporcional (o proporcional

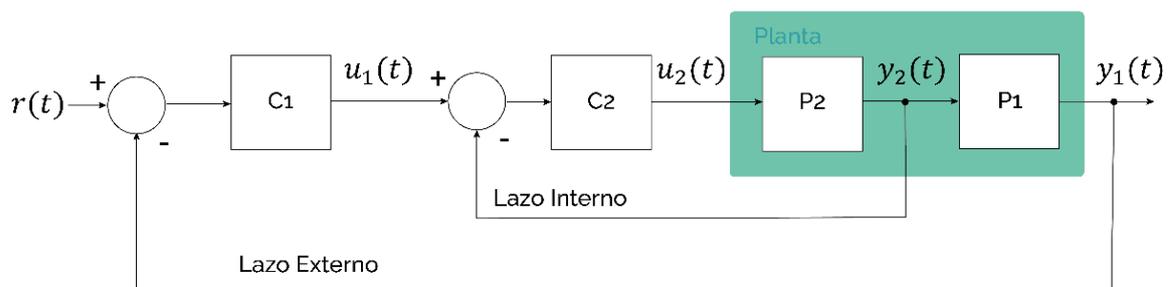
derivativa) por si sola no puede asegurar que el error se mitigará por completo, por tal motivo se añade la parte integral.

- Acción derivativa: al añadir los términos P e I, el sistema se puede hacer oscilatorio, la acción derivativa es la encargada de amortiguar estas oscilaciones proporcionándole mayor “rigidez” a la dinámica del sistema.

Como se puede notar, el uso del PID permite cumplir con los objetivos de regulación y seguimiento de manera efectiva. No obstante, tal como se muestra en la Figura 1.10, este esquema se aplica en sistemas de una entrada y una salida (SISO). Dado que el helicóptero de 3-GDL es un sistema de 2 entradas y 3 salidas (MIMO) el uso de esta estrategia de control requiere separar el sistema en 3 subsistemas en los cuales las 2 entradas se relacionan con 2 salidas (pitch y elevation) y la tercera salida (travel) es modificada por una de las otras (pitch). Esto se explica en detalle en la sección del modelo del sistema en el Capítulo 2, por ahora es importante tener en cuenta que esta estructura corresponde a un lazo de control en cascada.

### 1.3.3.2 Lazo de control en cascada

Como se mencionó en las secciones anteriores, en los drones multi-rotor la variación del ángulo pitch produce un movimiento en la dirección longitudinal, que para el caso del helicóptero de 3-GDL se traduce a un movimiento en el eje travel. Esto indica que el movimiento de la variable travel es controlada mediante el ángulo pitch, el cual a su vez se controla mediante un lazo interno que determina la variación de las velocidades de los rotores. Esta configuración corresponde a un lazo de control en cascada, representado en la Figura 1.11. Como se puede observar, la salida del lazo interno  $y_2(t)$  es la entrada de la planta P1 del lazo externo y la acción  $u_1(t)$  del controlador del lazo externo C1 ingresa como valor de referencia para el lazo interno.



**Figura 1.11.** Representación en diagrama de bloques de un esquema de control en cascada.

Como se puede verificar en la Figura 1.11, la planta está constituida de dos bloques P1 y P2 que, para el ejemplo mencionado del helicóptero de 3-GDL, el bloque P2 (lazo interno) corresponde al sistema de la variable pitch, mientras que el bloque P1 (lazo externo) es la variable travel, ya que al generar una variación en pitch se produce un movimiento en travel. Mayores detalles se explican en el siguiente capítulo.

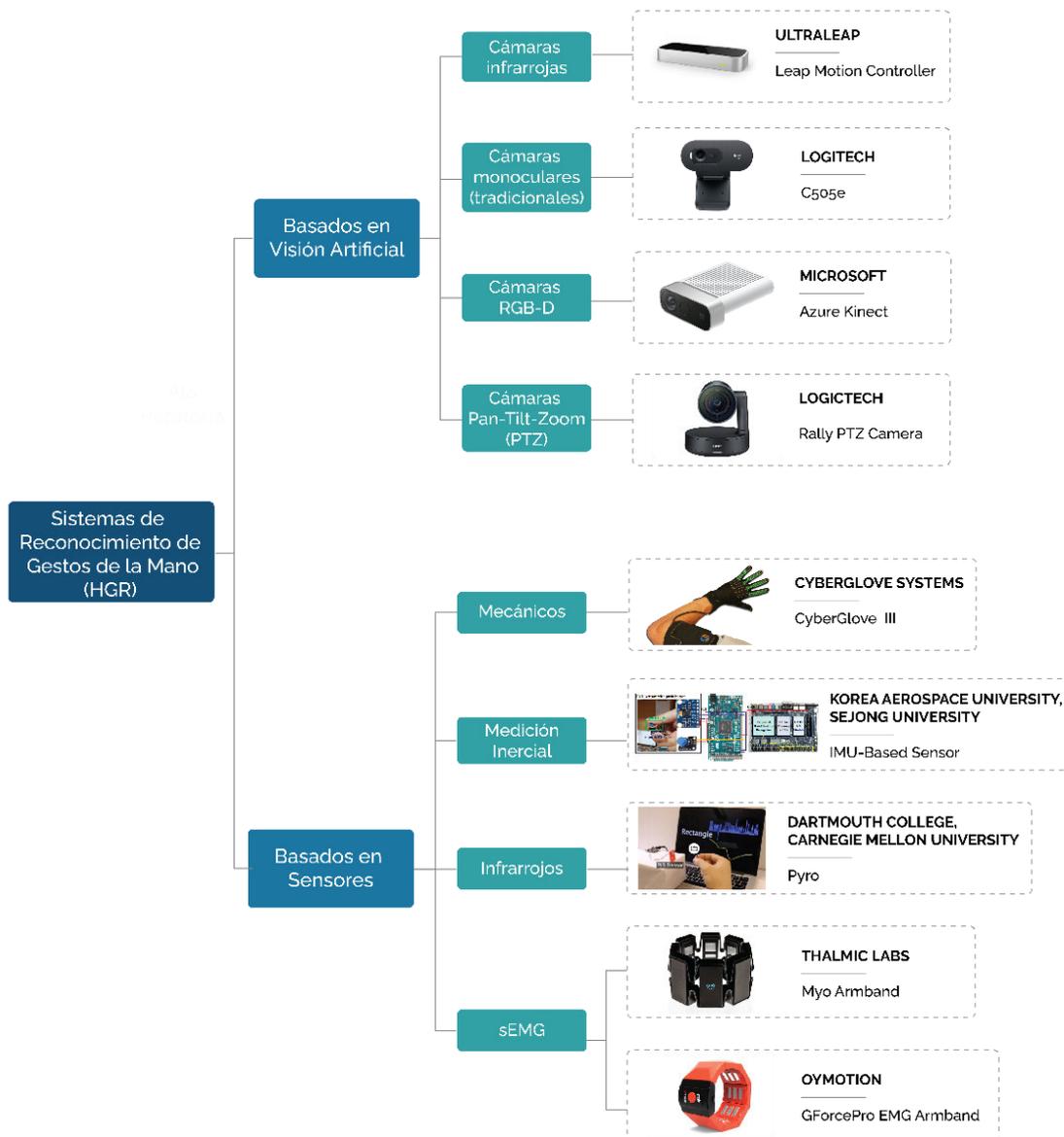
### **1.3.4. INTRODUCCIÓN A LOS SISTEMAS DE RECONOCIMIENTO DE GESTOS DE LA MANO**

El concepto de la interacción humano-computadora (HCI por sus siglas en inglés) apareció con el desarrollo de las primeras computadoras personales. Desde el inicio de la computación, este campo ha evolucionado desde el uso de dispositivos mecánicos como palancas e interruptores hasta la actualidad donde algunos de los elementos más usados son el teclado y mouse. De manera particular, en los últimos 20 años el método de HCI más común ha pasado a ser las pantallas táctiles ya que actualmente hay más de 6 billones de *smartphones* utilizados en el mundo [17], en los cuales se destaca el uso de este tipo de pantallas.

En este contexto, el uso de técnicas de comunicación no verbales ha demostrado numerosos beneficios para el desarrollo de interfaces humano-máquina (HMIs). De manera particular, el reconocimiento de gestos de la mano (o HGR por hand gesture recognition en inglés) ha sido utilizado en aplicaciones como reconocimiento de lenguaje de señas, control de prótesis y sistemas robóticos, realidad aumentada, manipulación de imágenes, entre otras [18]. El reconocimiento de gestos de la mano es un área de investigación amplia que puede ser subcategorizado de acuerdo con los métodos de clasificación y las tecnologías utilizadas para adquirir las señales de los gestos.

#### **1.3.4.1 Clasificación de los sistemas HGR**

Los sistemas de HGR pueden ser divididos en dos grupos de acuerdo con las tecnologías utilizadas para la adquisición de señales, estos son: basados en visión artificial y basados en sensores. En la Figura 1.12. se muestra esta clasificación con sus respectivas subcategorías y algunos ejemplos.



**Figura 1.12.** Clasificación de los sistemas de reconocimiento de gestos de la mano (HGR) por la tecnología para la adquisición de señales.

#### 1.3.4.1.1 HGR basado en visión

La gran mayoría de técnicas utilizan el método basado en visión, en el cual destacan el uso de dispositivos como Kinect de Microsoft [19] y Leap Motion (que utiliza una cámara en el espectro de infrarrojos) [20]. En este caso, el reconocimiento de gestos se basa en la obtención de un modelo de la mano como el esqueleto de los dedos o la silueta para lo cual se utilizan imágenes obtenidas de una cámara. Las principales tecnologías empleadas para el reconocimiento basado en visión artificial son:

- **Cámaras RGB-D:** se trata de cámaras que pueden captar imágenes en el espectro visible (RGB por red, green y blue) y además pueden generar un mapa de profundidad. Para esto utilizan un proyector y detector infrarrojo al lado de la cámara RGB. En esta tecnología destacan los dispositivos Kinect 2.0 y Azure Kinect de Microsoft.
- **Cámaras infrarrojas:** uno de los problemas que se puede encontrar en el uso de visión artificial es la interferencia generada por la luz ambiental, lo cual puede ser solucionado con una cámara de detección de espectro infrarrojo como es el caso del dispositivo Leap Motion de la marca Ultraleap.
- **Cámaras TPZ:** las siglas TPZ corresponden a tilt, pan y zoom que se refieren a los movimientos vertical, horizontal y acercamiento respectivamente. Estas son las acciones que se pueden realizar con estas cámaras y son de utilidad para rastrear y detectar los gestos de la mano.
- **Cámaras monoculares tradicionales:** la principal ventaja que presenta el reconocimiento de gestos basado en visión es la relativa facilidad con que se puede realizar. Ya que, además de los dispositivos mencionados antes, se puede emplear una webcam común, tal como se explica en [21].

En cualquiera de los casos mencionados, la principal dificultad que se presenta al realizar HGR basado en visión es la oclusión, que se refiere a la presencia de un objeto externo entre la cámara y el objetivo, que en este caso es la mano. Por esa razón se han desarrollado otros métodos empleando la detección de actividad muscular o inercial del brazo, para lo cual se usan otro tipo de sensores especializados.

#### *1.3.4.1.2 HGR basado en sensores*

En los sistemas de adquisición basados en sensores para reconocimiento de gestos, se pueden mencionar los de tipo infrarrojos, resistivos o capacitivos, electromiográficos, inerciales, etc. En estos casos el sistema genera señales que posteriormente deben ser procesadas mediante métodos de extracciones de características para luego clasificar los gestos de forma correspondiente. Algunas de las tecnologías más utilizadas son:

- **Mecánicos:** se basan en la deformación o movimiento mecánico de algún elemento que pueda generar una señal eléctrica como por ejemplo en forma resistiva. Tal es el caso del guante CyberGlove II [22] que utiliza sensores resistivos en cada dedo para detección de gestos.

- **Inerciales:** en este caso se hace uso de unidades de medición inercial (IMU por sus siglas en inglés) para la detección de gestos. En [23] se describe la utilización de una IMU para la detección de caracteres alfanuméricos realizados con el movimiento de la mano.
- **Infrarrojos:** utilizan sensores que detectan la radiación infrarroja generada por la temperatura del cuerpo. En este caso, el calor natural de las manos irradia ondas en el espectro infrarrojo que son detectadas por el sensor y se generan señales en función del movimiento, las cuales luego pueden ser procesadas para detección de gestos.
- **Electromiográficos:** para este tipo de sensores se utilizan electrodos que miden la actividad muscular del brazo. Al realizar los diferentes movimientos con las manos se generan señales que luego se ingresan al sistema de clasificación. A continuación, se explica a mayor detalle el funcionamiento de este tipo de sensores, ya que uno de este tipo es el empleado en el presente proyecto.

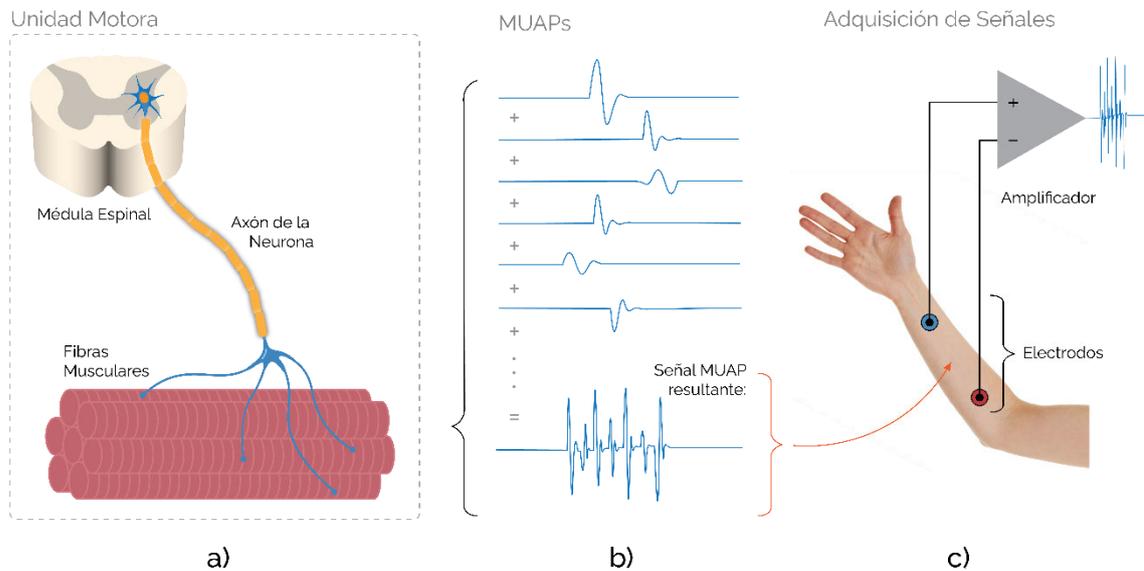
#### 1.3.4.2 Características de las señales electromiográficas.

Cada vez que se realiza un movimiento del cuerpo, los músculos involucrados generan señales eléctricas que pueden ser medidas. La medición de estas señales se denomina Electromiografía (EMG) que viene de las raíces *myo* que significa músculo, *graphos* que significa escritura y el prefijo *electro* por electricidad [24].

La contracción muscular es producida por las denominadas *Unidades Motoras* que se conforman por las diferentes estructuras que se muestran en la Figura 1.13.a). Cuando las neuronas motoras se activan, se envía una señal que modifica el potencial de la fibra muscular de -80mV a 30mV y recibe el nombre de *potencial de acción de la unidad motora* o MUAP (por *motor unit action potential* en inglés) [25]. Los MUAPs se generan de forma independiente por cada unidad motora y al final, la señal adquirida se conforma por la superposición de cada componente como se muestra en la Figura 1.13.b). El efecto de la piel y otros tejidos produce ruido y distorsiones en la medición por lo que siempre existirán diferencias con la señal real [26].

Para adquirir las señales producidas por la contracción muscular o EMGs se utilizan electrodos de dos tipos: superficiales y de inserción. Los electrodos de inserción utilizan sensores en forma de aguja que se introducen en el músculo con el beneficio de que se minimiza el ruido producido por la piel y otros tejidos. No obstante, este método no es práctico para su implementación en interfaces humano-máquina. Por esta razón se usan

electrodos superficiales (y se denomina electromiografía de superficie o sEMG), los cuales permiten adquirir estas señales de forma no invasiva. En ambos casos, las mediciones pasan a una etapa de amplificación de tipo diferencial, con lo cual se elimina el ruido común en ambos electrodos, tal como se indica en la Figura 1.13.c).



**Figura 1.13.** Proceso de generación y adquisición de señales EMG: a) Partes de la unidad motora, b) La señal EMG resulta de la superposición de los potenciales de acción de las unidades motoras, c) Se usa un amplificador diferencial para adquirir las señales.

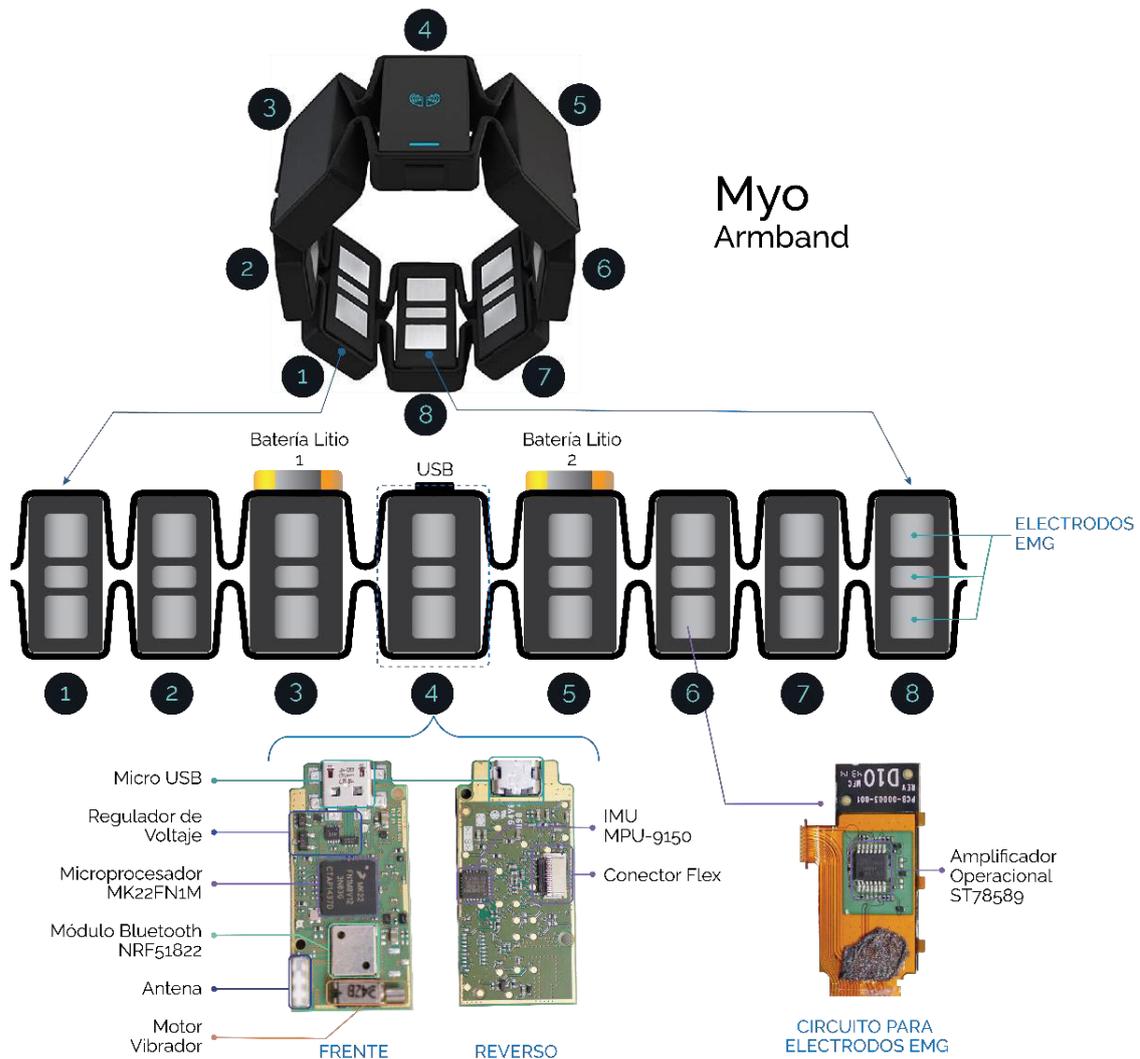
### 1.3.4.3 Myo Armband

Como se mencionó anteriormente, el uso de señales EMG para reconocimiento de gestos resulta de gran interés ya que se evitan los problemas causados por oclusión, iluminación deficiente o interferencias que presentan los sensores basados en visión. Entre las posibilidades que se pueden encontrar en la literatura para utilizar esta técnica muchas hacen uso de sensores fabricados [27] [28], dentro de los cuales, el sistema comercial de adquisición que más destaca es el sensor Myo Armband elaborado por la empresa Thalmic Labs [29]. El dispositivo fue discontinuado en 2018, sin embargo, sigue siendo de gran utilidad en grupos de investigación por su versatilidad y características.

- **Partes del dispositivo**

Este brazalete consta de 8 electrodos EMG, una unidad de medición inercial (o IMU por sus siglas en inglés) compuesta de: un acelerómetro, giroscopio y magnetómetro, de 3 ejes cada uno y la transmisión de datos al computador se realiza por medio de un módulo

bluetooth, véase Figura 1.14. Además, el dispositivo consta de un motor vibrador para alertar al usuario cuando un evento en particular ocurre.



**Figura 1.14.** El dispositivo Myo Armband y sus partes constituyentes.

Las especificaciones técnicas de los componentes y sensores del brazalete se resumen en la Tabla 1.1.

**Tabla 1.1.** Características y componentes del dispositivo Myo Armband.

Elemento	Modelo / Características	Cantidad
Electrodos EMG	Electrodos Superficiales conectados a un amplificador operacional ST78589	8
Bluetooth LE	NRF51822	1

IMU	InvenSense MPU-9150: 9 ejes (magnetómetro, acelerómetro, giroscopio).	1
Microprocesador	Freescale Kinetis ARM Cortex M4 120Mhz MK22FN1M	1
Batería	Batería de Litio: 3.7V, 260mAh	2

La operación del brazalete es controlada por el procesador ARM Cortex M4, el cual administra la transmisión por bluetooth con el chip BLE (Bluetooth de baja energía) NRF51822 [30]. El dispositivo consta de 8 piezas unidas por cables flexibles en las cuales se encuentran 3 electrodos superficiales para la adquisición de señales EMG. El acondicionamiento de las mismas se realiza por medio de amplificadores operacionales ST78589 y se comunican a la placa principal. Cada bloque del brazalete posee una numeración de 1 a 8 tal como se indica en la Figura 1.14, en la cual se muestra la distribución de los componentes mencionados en la Tabla 1. En la pieza 4 se encuentra la placa principal que incluye el conector USB, microprocesador, módulo bluetooth, vibrador e IMU. Los bloques 3 y 5 contienen las dos baterías de 3.7V que proporcionan la energía en el sistema y, por último, los electrodos diferenciales para medición de señales EMG y su circuito de acondicionamiento correspondiente se encuentran en todos los 8 bloques.

- **Señales del dispositivo**

Las señales que entrega el sensor Myo Armband corresponden a las EMG de 8 canales (uno por cada electrodo diferencial). Estas señales son muestreadas a una frecuencia de 200Hz. Además de estas señales, el dispositivo también entrega la información de la orientación en el espacio tridimensional. Para esto, se realiza un procesamiento interno que fusiona la información del giroscopio, magnetómetro y acelerómetro para poder entregar un valor de orientación de gran exactitud en forma de cuaterniones.

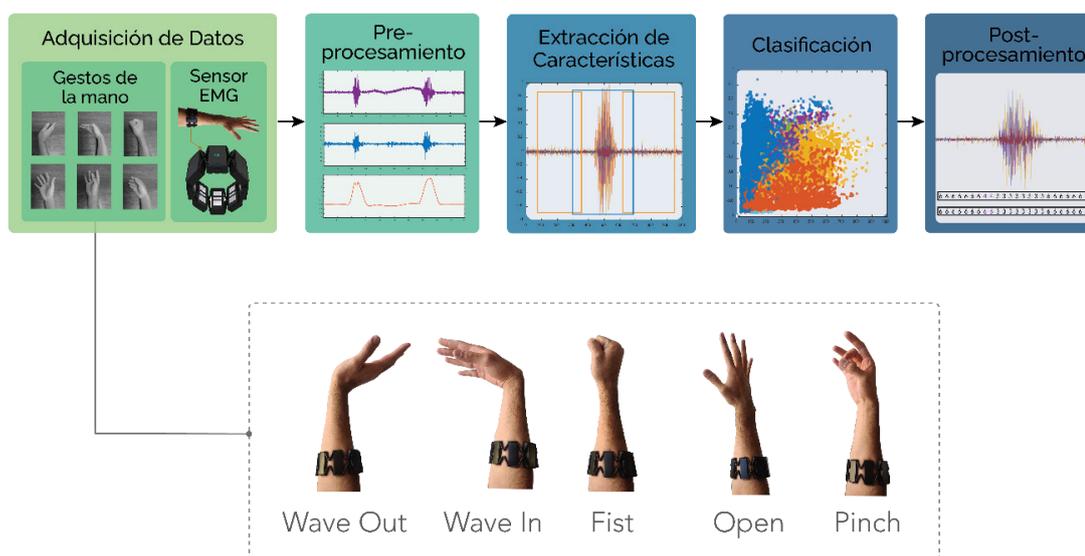
#### **1.3.4.4 Sistema de reconocimiento de gestos de la mano del Proyecto de Investigación Grupal PIGR-19-07**

El sistema de HGR utilizado en el presente trabajo ha sido desarrollado en el Laboratorio de Investigación en Inteligencia y Visión Artificial de la EPN como parte del proyecto *Reconocimiento de Gestos de la Mano Usando Señales Electromiográficas e Inteligencia*

*Artificial y su Aplicación para la Implementación de Interfaces Humano-Máquina y Humano-Humano*, cuyo código de identificación es PIGR-19-07.

El sistema se basa en la adquisición y procesamiento de señales EMG del antebrazo utilizando un Myo Armband. Como se explicó anteriormente, este es un dispositivo que reúne una gran cantidad de sensores para entregar señales de tipo EMG y de medición inercial mediante una IMU, las cuales son transmitidas a un computador vía bluetooth. El sistema utilizado es capaz de reconocer 5 diferentes gestos que se indican en la Figura 1.15. Cada uno posee un nombre distinto dependiendo de la acción que representan:

- *Wave Out* corresponde a la palma abierta hacia fuera (extensión de la muñeca),
- *Wave In* es lo contrario, con la palma abierta hacia adentro (flexión de la muñeca),
- *Fist* corresponde a la mano cerrada haciendo puño,
- *Open* es la palma y los dedos abiertos sin mover la muñeca y
- *Pinch* hace referencia al doble tope de las yemas del dedo anular (o medio) con el pulgar.



**Figura 1.15.** Diagrama de bloques del sistema de reconocimiento de gestos y los 5 gestos que se pueden detectar.

La gran mayoría de los sistemas de reconocimiento de gestos utilizan algún método de aprendizaje para determinar el gesto realizado en función de las señales detectadas, ya sean imágenes, mediciones inerciales o de tipo EMG. En este contexto, los sistemas de HGR poseen 5 etapas que son: adquisición, pre-procesamiento, extracción de

características, clasificación y post-procesamiento, tal como se indica en la Figura 1.15. A continuación se describen cada una de estas etapas:

#### 1.3.4.4.1 Adquisición

Para el sistema en cuestión, la adquisición de datos se realiza por medio del dispositivo Myo Armband, el cual entrega señales EMG de 8 canales que corresponden al número de electrodos que están en contacto con el antebrazo del usuario del brazalete.

Para el sistema de reconocimiento utilizado, el proceso de adquisición abarca tanto la toma de señales en línea (durante la utilización), como la toma de señales y su tratamiento para la generación del modelo de entrenamiento. Por esta razón, en [31], se menciona la estructura de dataset, la distinción entre usuario general y específico y la corrección de rotación del sensor como parte del proceso de adquisición. A continuación, se menciona brevemente cada una de estas características.

- **Usuario general**

El sistema utilizado es denominado *de usuario general* lo que significa que cualquier usuario puede hacer uso del sistema rápidamente sin la necesidad de realizar un entrenamiento previo al sistema. Por otro lado, se encuentran los *de usuario específico* que, como su nombre lo indica, cada usuario debe entrenar al sistema realizando sus propios gestos y el sistema tendrá un alto grado de exactitud para dicho usuario y por la misma razón, tendrá una exactitud muy baja para otros. Esto dificulta que este tipo de sistemas pueda ser utilizado por varias personas.

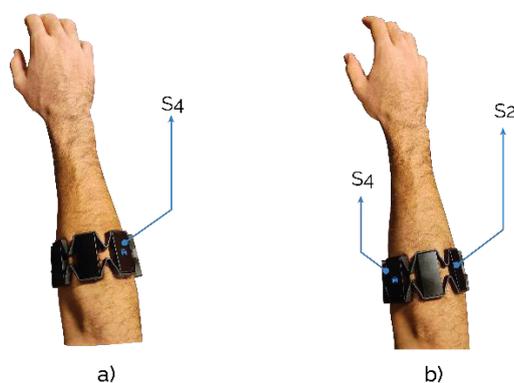
- **Dataset de entrenamiento y testeo**

Para lograr la cualidad de usuario general, el sistema debe ser entrenado con la mayor cantidad de personas posible. Para este fin, el reconocimiento de gestos utilizado en este proyecto se realizó con un dataset de 612 personas de diferentes edades concentradas sobre todo entre 18 y 25 años. Además, se puede mencionar que el 96% de las personas son derechas y 4% zurdas y adicionalmente, el 66% son hombres y 34% mujeres. De este dataset de 612 personas, el 50% (es decir los datos de 306 personas) se utilizaron para entrenamiento del sistema y la otra mitad para testeo. El dataset en cuestión se puede encontrar en el repositorio libre EMG-EPN-612 indicado en [32].

- **Corrección de orientación**

El fabricante del dispositivo Myo Armband recomienda el uso del brazalete con el sensor 4 (S4) ubicado sobre el medio de la superficie posterior del antebrazo, tal como se muestra

en la Figura 1.16 a). Si el sistema es entrenado con una configuración específica, por ejemplo, la indicada por el fabricante, y no se realiza ninguna otra corrección, la exactitud se reduce considerablemente al cambiar de orientación al sensor. Tanto que el sistema se vuelve inutilizable. Debido a este problema, en [33] se propone un método para corrección de orientación basado en la detección de máxima amplitud.



**Figura 1.16.** Colocación del sensor en 2 diferentes orientaciones: a) orientación recomendada por el fabricante, b) orientación arbitraria.

El algoritmo de detección de máxima amplitud o MAD (por sus siglas en inglés) consiste en identificar el sensor (desde S1 hasta S8) que presente la mayor actividad durante el tiempo que el usuario realiza el gesto de sincronización, que corresponde a Wave Out. En base a esa identificación se reordenan las señales de los sensores mediante software, de modo que se interpretan como si se tuviera la orientación recomendada. Posteriormente se realiza el sistema de reconocimiento tradicional en base a la nueva orientación [33].

#### *1.3.4.4.2 Preprocesamiento*

Luego de realizar la adquisición de datos y el ordenamiento correspondiente, se tiene la etapa de preprocesamiento. En esta se utilizan ventanas de muestreo deslizantes que agrupan las señales obtenidas en partes más pequeñas para analizarlas. Esta etapa sirve para determinar si las señales encontradas en cada ventana deben o no pasar a la siguiente etapa. Para esto se tiene un umbral de energía del 17% [31] que deben superar para continuar, de este modo se evita que las señales del estado de reposo ingresen innecesariamente al sistema de clasificación ya que eso generaría falsas detecciones.

#### *1.3.4.4.3 Extracción de características*

En el sistema de HGR utilizado se utilizan 5 funciones para extracción de características, las cuales se aplican sobre cada ventana únicamente cuando se supera el umbral de energía explicado anteriormente. Las funciones utilizadas son las que se mencionan a

continuación y las siglas entre paréntesis indican su nombre en inglés. Se puede encontrar una explicación con mayor detalle sobre estas funciones en la referencia [31].

- **Desviación estándar (SD):** esta característica indica el grado de dispersión de la señal EMG, es decir qué tan separadas se encuentran las muestras con relación al promedio.
- **Envolvente absoluta (AE):** se basa en la obtención de atributos instantáneos de una serie de datos en el tiempo como su amplitud y frecuencia.
- **Media del valor absoluto (MAV):** como su nombre lo indica, es el promedio del valor absoluto de la amplitud de la señal EMG.
- **Energía (E):** representa la distribución de energía de la señal obtenida.
- **Raíz cuadrática media (RMS):** se relaciona con la fuerza y contracción muscular generadas durante la obtención de la señal EMG.

#### 1.3.4.4.4 Clasificación

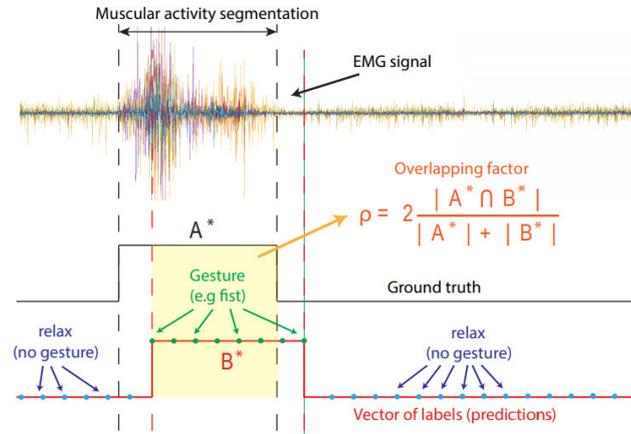
El sistema utilizado emplea una máquina de vectores de soporte o SVM (por sus siglas en inglés) para la clasificación de los gestos realizados. Las SVMs son modelos de aprendizaje supervisado que se utilizan para encontrar el hiperplano óptimo en un problema de clasificación de datos. En la aplicación de HGR, este método resulta de mucho interés debido a que entrega una exactitud remarcable al tiempo que requiere una menor capacidad computacional [34].

De manera resumida, el concepto de hiperplano es una región en el espacio ( $n$ -dimensional, dependiendo de las  $n$  clases que se tengan. En este caso  $n=5$  que corresponde a los 5 gestos que reconoce el sistema) que sirve como un límite para separar a las diferentes clases. El caso más simple es un hiperplano lineal que para 2 dimensiones corresponde a una línea y en 3 es un plano. En este caso, en lugar de un hiperplano se utiliza una denominada *función kernel* polinomial (no lineal) de grado 3 como límite de separación entre clases. Adicionalmente se utiliza una estrategia “uno contra uno” que se basa en separar el problema multiclase en varios clasificadores binarios para luego combinar todos los resultados. Los detalles de esta implementación se pueden encontrar en [31].

#### 1.3.4.4.5 Post-procesamiento

La etapa de post-procesamiento se encarga de eliminar las posibles predicciones espurias que puedan presentarse. Para esto, se compara el intervalo temporal en el que se detecta actividad muscular (segmentado manualmente) con el intervalo en el que se detecta un

gesto en específico. La Figura 1.17 muestra un esquema del método realizado para post-procesamiento. Esto permite eliminar las predicciones falsas que pueden surgir cuando se mantiene el estado de reposo.



**Figura 1.17.** La etapa de post-procesamiento compara el valor de *verdad fundamental* (ground truth) con el vector de predicciones. Gráfico tomado de [31].

## 2. METODOLOGÍA

El capítulo presentado a continuación expone la metodología aplicada para el desarrollo del presente trabajo de titulación. En este se describen las técnicas aplicadas para la elaboración del proyecto empezando desde la recopilación de información, para lo cual se han empleado principalmente fuentes primarias como libros, revistas y artículos técnicos y también fuentes secundarias como artículos de páginas web, trabajos de titulación, etc. Adicionalmente, dado que el proyecto es de implementación, la principal técnica para obtención de información ha sido la experimentación y registro de resultados.

La experimentación consiste principalmente en la elaboración de un banco de pruebas para comprobar funciones como el seguimiento de referencias y cumplir con el objetivo de integración con el sistema de reconocimiento de gestos. La obtención de datos se realiza a través de mediciones del sistema y de las posiciones angulares en los tres grados de libertad de la plataforma.

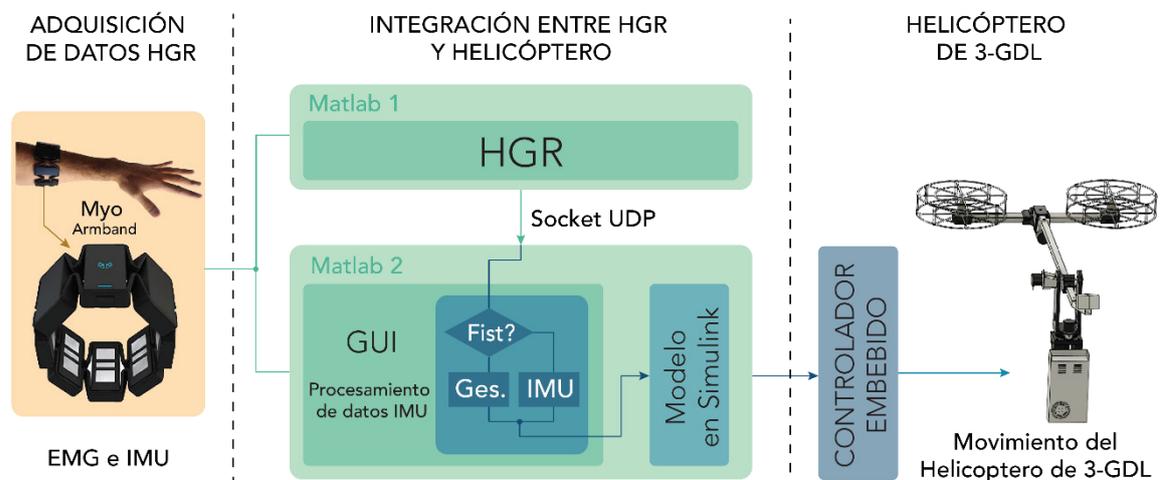
### 2.1. ARQUITECTURA DEL SISTEMA

El presente trabajo de titulación tiene como fin demostrar la funcionalidad del sistema de reconocimiento de gestos como una interfaz para interacción entre humano y máquina

aplicada al manejo de sistemas aéreos. Para este propósito la arquitectura completa del sistema consta de 3 componentes principales que son:

1. Adquisición de datos para el sistema de reconocimiento de gestos de la mano (HGR)
2. Integración entre HGR y helicóptero de 3-GDL
3. Helicóptero de 3-GDL (banco de pruebas)

Como representación de la plataforma aérea se ha implementado un helicóptero de 3-GDL que sirve además como un banco de pruebas para la implementación y estudio de sistemas de control. En la Figura 2.1 se muestra un esquema de la arquitectura propuesta para este proyecto. Cada una de las partes mostradas se explicarán con mayor detalle en las siguientes secciones del presente capítulo.



**Figura 2.1.** Arquitectura del sistema.

De manera resumida, el sistema de reconocimiento de gestos sirve para generar un valor de referencia que se ingresa al sistema de control para que el helicóptero de 3-GDL realice un movimiento correspondiente. El sistema propuesto permite la utilización de dos modos para control del movimiento de la plataforma. El primero es mediante el reconocimiento de gestos, en el cual cada uno de los 5 gestos, mencionados en el capítulo anterior, sirven para generar valores de setpoint específicos que la plataforma debe seguir. El segundo modo hace uso de la información inercial del sensor Myo, para esto se utilizan los datos entregados por la IMU del dispositivo. En este modo los movimientos horizontal y vertical del antebrazo del usuario son relacionados con el eje horizontal (travel) y vertical (elevation) del helicóptero. Esta funcionalidad recibe el nombre de *control multimodo* debido a los dos modos de control que permite; en la sección de integración se describe con mayor detalle las características de esta funcionalidad. Además de los métodos mencionados, el

helicóptero también puede seguir valores de referencia específicos indicados mediante la interfaz gráfica del sistema sin la necesidad de utilizar el sistema de reconocimiento de gestos o movimientos del brazo.

En la siguiente sección se describen los componentes e implementación del prototipo y posteriormente se indica el funcionamiento del sistema de control, con el cual se integra el reconocimiento de gestos que se describió previamente en la Sección 1.3.

## 2.2. IMPLEMENTACIÓN DEL HELICÓPTERO DE 3-GDL

La implementación del prototipo se presentará en 2 partes que son el hardware de control electrónico, que corresponde a los elementos eléctricos y electrónicos para mover el prototipo e interactuar con el mismo; y por otro lado, se encuentra el prototipo mecánico, que corresponde a la estructura y los elementos necesarios para permitir el movimiento en los 3 ejes pitch, elevation y travel.

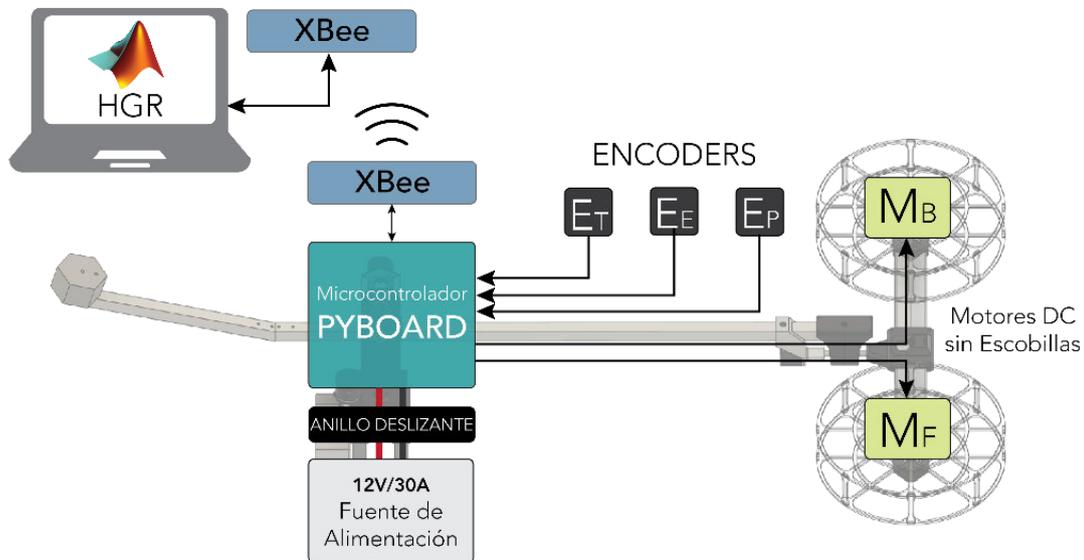
### 2.2.1. HARDWARE DE CONTROL ELECTRÓNICO

El hardware necesario para el correcto funcionamiento del helicóptero de 3-GDL se escogió considerando las tres funciones principales que son: i) realimentación y control, ii) comunicación y iii) alimentación. Los elementos necesarios para cada función se indican en la Tabla 2.1.

**Tabla 2.1.** Clasificación de los componentes del hardware para control de acuerdo con su función.

<b>Funciones</b>	<b>Descripción</b>	<b>Elementos</b>
Realimentación y control	Medición de las variables travel, elevation, pitch, procesamiento de las mismas y generar movimiento en la plataforma	<ul style="list-style-type: none"> <li>• Encoders</li> <li>• Microcontrolador</li> <li>• Motores</li> <li>• Drivers ESC</li> </ul>
Comunicación	Interacción entre la plataforma y la interfaz de usuario en la PC.	<ul style="list-style-type: none"> <li>• Módulos de comunicación inalámbrica</li> </ul>
Alimentación	Entregar la energía para alimentar los componentes del sistema	<ul style="list-style-type: none"> <li>• Fuente</li> <li>• Anillo deslizante</li> <li>• Relés</li> <li>• Pulsadores</li> </ul>

El esquema general del hardware necesario para el control electrónico se muestra en la Figura 2.2, aquí se pueden observar los diferentes componentes utilizados en el sistema, los mismos que se describirán en los siguientes párrafos.



**Figura 2.2.** Esquema de los componentes eléctricos y electrónicos necesarios para el sistema.

### 2.2.1.1. Elementos de realimentación y control

Estos elementos son los necesarios para realizar la regulación de las 3 posiciones angulares de la plataforma que son: travel, elevation y pitch. Para la medición de ángulos se han utilizado encoders incrementales. Las señales entregadas por estos elementos son procesadas por medio de un microcontrolador en una placa Pyboard donde las leyes de control son calculadas y enviadas a los actuadores, que en este caso son motores DC sin escobillas que comandan todo el movimiento necesario en la plataforma.

#### 2.2.1.1.1. Pyboard

Se ha escogido esta placa de desarrollo en base a las siguientes necesidades:

- Procesamiento de señales de encoders: dado que se utilizan 3 encoders (uno para cada grado de libertad) para la realimentación de las posiciones angulares, el sistema debe poder procesar 6 señales diferentes (2 por cada encoder). En los siguientes párrafos se explica el funcionamiento de las señales de los encoders.
- Periféricos para comunicación serial: esto permitirá establecer una interfaz entre el computador y el sistema para poder interactuar con el usuario.

- Cálculo de la ley de control: el microcontrolador debe tener la capacidad y velocidad de procesamiento necesarias para poder calcular las leyes de control de forma rápida y efectiva.

La placa Pyboard posee las especificaciones mostradas en la Tabla 2.2.

**Tabla 2.2.** Características de la placa Pyboard.

	<b>Especificaciones</b>	<b>Valor</b>
<b>Microcontrolador</b>	MCU	STM32F405RGT6 (32 bits)
	CPU	Cortex-M4F
	Flash interna	1024 kB
	RAM	192 kB
	Frecuencia máxima	168 MHz
<b>Características de la placa</b>	Conexión USB	Sí
	SD externa	Sí
	Acelerómetro	Sí
	Reloj en tiempo real	Cristal 32 kHz
	Pulsadores	USR, RST
	LEDs	R, G, B, Y
<b>Alimentación</b>	Voltaje de alimentación	3.6 – 16V
	Corriente máxima de salida @3.3V, regulado	250 mA
<b>Capacidades E/S</b>	Pines E/S	30
	Pines PWM	20
	Pines convertidores A/D	16 (4 blindadas)
	Pines convertidores D/A	2
<b>Periféricos</b>	Timers independientes	14
	UART	5
	I2C	2
	SPI	2
	CAN	2

De manera específica, las funcionalidades que ofrece esta placa para la implementación del prototipo necesario para el presente proyecto son las siguientes:

### **a) Microcontrolador**

La placa Pyboard viene equipada con un microcontrolador STM32F405RGT6 de la marca ST Microelectronics [35]. Este procesador incluye las prestaciones que se muestran en la Tabla 2.2. Como se puede observar, tiene una capacidad de procesamiento de 32 bits, lo cual permite realizar operaciones matemáticas rápidamente, además de que la velocidad de reloj es de 168MHz. Estas dos cualidades combinadas permiten calcular y generar las acciones de control de forma rápida y efectiva.

### **b) Lenguaje MicroPython**

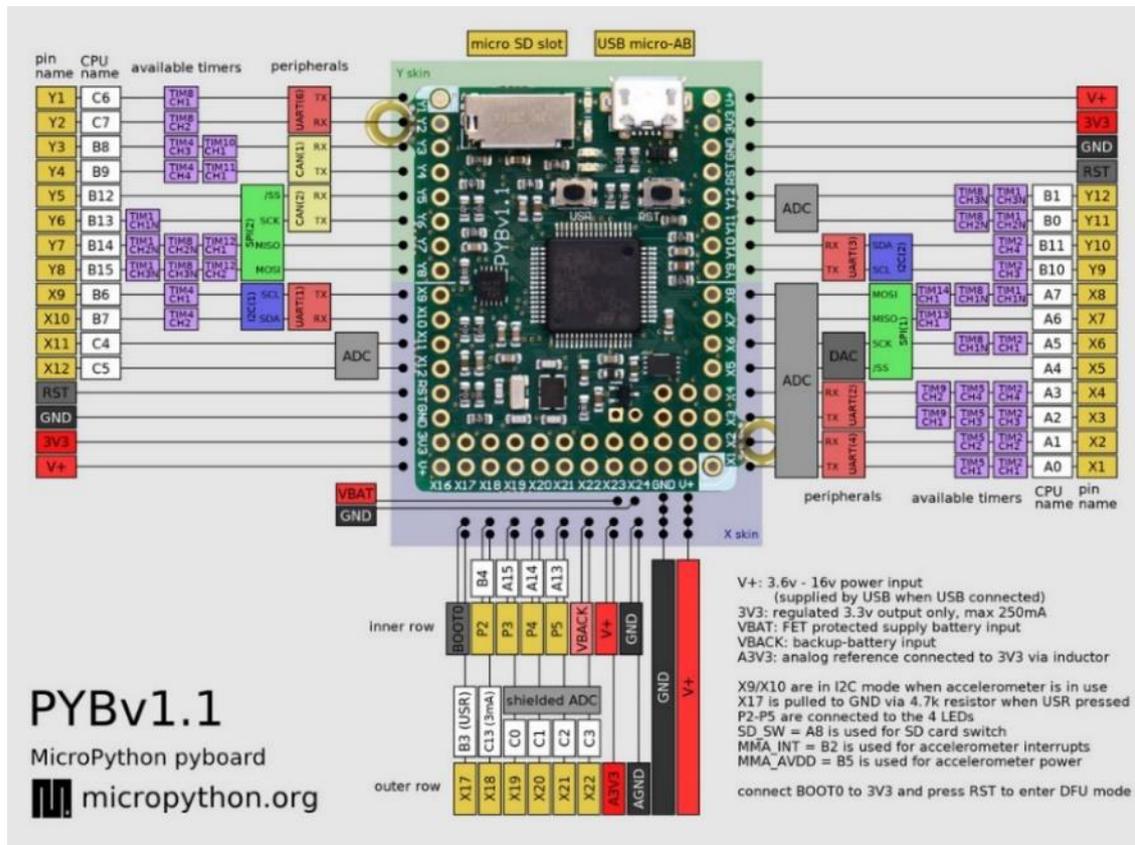
MicroPython es una reimplementación del lenguaje Python 3 optimizada para funcionar en microcontroladores y sistemas embebidos. Existen algunas diferencias entre Python completo y MicroPython, sin embargo, la principal cualidad es que MicroPython está diseñado para funcionar bajo condiciones extremadamente limitadas (como por ejemplo 192 kB de RAM). Este corre directamente sobre el hardware sin necesidad de un sistema operativo subyacente como Windows o Linux. Todas las operaciones y servicios que normalmente son provistos por el sistema operativo se manejan directamente por MicroPython. En este caso el lenguaje está en control directo y completo sobre el hardware [36].

El uso del lenguaje MicroPython radica en la facilidad que representa el acceso a los periféricos del microcontrolador de la placa y la sintaxis en general del programa para control de la plataforma. En la memoria interna de la placa Pyboard se incluye un módulo de Python denominado pyb, a través de la cual se pueden acceder a diferentes clases de objetos para controlar los temporizadores, LEDs, puertos UART, I2C, etc.

### **c) Temporizadores / contadores**

La placa Pyboard permite acceder a 14 temporizadores (o Timers) de forma independiente que poseen diferentes características. Adicionalmente, el uso del lenguaje MicroPython permite configurar dichas entradas y salidas con gran facilidad ya que se realiza mediante constructores definidos. Para el caso de los Timers, el constructor es `pyb.Timer(parámetros)` con el cual se configura el Timer seleccionado (del 1 al 15 exceptuando el 13), la frecuencia y preescalador para el conteo (con respecto a los 168 MHz), etc. Dentro del contexto del presente proyecto, los Timers son utilizados para el conteo de las señales de los encoders (para realimentación de las posiciones angulares) y la generación de señales de modulación de ancho de pulso (PWM) para el control de los

motores. Una imagen de la placa Pyboard junto con los periféricos disponibles para cada uno de sus pines se presenta en la Figura 2.3.



**Figura 2.3.** Placa Pyboard con la especificación de cada uno de sus pines. Imagen tomada de [38].

A continuación, se muestra un ejemplo de la sintaxis para el uso de los Timers en el Pyboard con MicroPython:

- **Lectura de encoders:**

```
# Se escoge el Timer 2 de 30 bits y cuenta hasta un valor máximo de
10^9 que es la máxima potencia de 10 que se puede contar con 30 bits:
timEnc1 = Timer(2, prescaler=0, period=100000000)
# Configurar los canales 1 y 2 para modo encoder:
timEnc1.channel(1, pyb.Timer.ENC_AB)
# Se coloca el contador en el valor medio para que pueda contar
valores negativos sin que se desborde el valor de la cuenta:
timEnc1.counter(50000000)
```

Como se puede observar, el módulo pyb permite acceder al parámetro `Timer.ENC_AB`, con lo cual se configura a nivel de hardware directamente al periférico para que permita el conteo de las señales de los encoders. Esto es una funcionalidad que ofrece la familia de microcontroladores STM32, sin embargo, se hace fácilmente accesible y configurable a través de MicroPython.

#### **Configuración de salidas PWM para motores:**

```
# Se escoge el Timer 12 a una frecuencia de 50Hz para trabajar con  
Los drivers de los motores:  
timESC = Timer(12, freq=50)  
# Creación del objeto para controlar el motor B en el canal 1 del  
timer en el pin Y7:  
ESC_B = timESC.channel(1, Timer.PWM, pin=Pin('Y7'))  
# Creación del objeto para controlar el motor F en el canal 2 del  
timer en el pin Y8:  
ESC_F = timESC.channel(2, Timer.PWM, pin=Pin('Y8'))
```

Luego de la configuración de los Timers en modo PWM se puede controlar el ancho de pulso necesario mediante el método `pulse_width_percent()` con esto se puede colocar directamente un valor en porcentaje, lo cual facilita la realización de cálculos.

#### **d) Periféricos para comunicación**

Al igual que en los ejemplos anteriores, el Pyboard tiene una gran cantidad de periféricos para comunicación serial que se pueden configurar fácilmente de manera programática. Para el presente proyecto se ha utilizado el puerto UART debido a que los módulos de comunicación empleados hacen uso de esta interfaz para comunicación serial. La configuración en MicroPython se realiza mediante el siguiente código:

```
# Se selecciona el puerto UART 2, velocidad de transmisión 115200  
baud, 8 bits de datos, sin paridad y 1 bit de parada:  
xbee = UART(2, 115200, bits=8, parity=None, stop=1)
```

En este caso se puede observar que la configuración de puertos de comunicación se facilita en gran medida gracias al uso de la placa Pyboard y MicroPython. Adicionalmente, otra prestación que ofrece la placa es su tamaño reducido ya que ocupa una superficie de 33 x 42.5 mm.

### 2.2.1.1.2. Encoders

Para la realimentación de las variables se utilizan encoders incrementales de tipo ES38 de la marca CALT [39], tal como el indicado en la Figura 2.4. Las características de este componente se presentan en la Tabla 2.3.



**Figura 2.4.** Encoders utilizados para la realimentación de las posiciones angulares.  
Imagen tomada de [39].

**Tabla 2.3.** Especificaciones del encoder ES38-06G1000BSC824

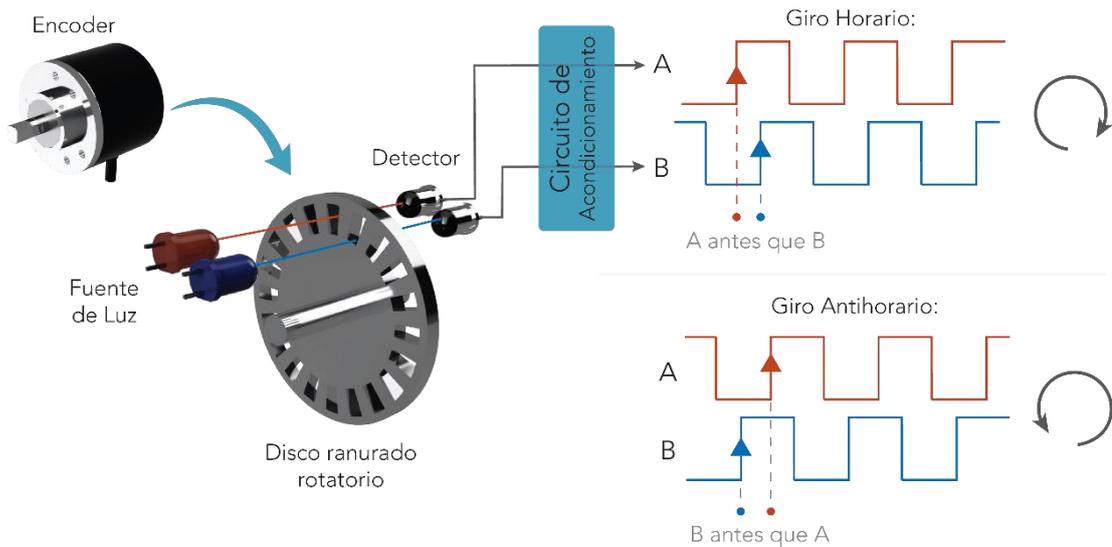
Especificaciones	Valor o tipo	Unidad
Resolución	1000	PPR (pulsos por revolución)
Voltaje de alimentación	8 – 24	V
Tipo de salida	NPN	-
Velocidad máxima	3000	RPM

El encoder CALT ES38-06G1000BSC824 es de 1000 pulsos por revolución (PPR), con lo cual es posible obtener una resolución mínima de  $0.36^\circ$  de acuerdo con lo indicado en la Ecuación 2.1.

$$resolución = 360 \frac{^\circ}{rev} \times \frac{1}{1000} \frac{rev}{pulso} = 0.36 \frac{^\circ}{pulso} \quad (2.1)$$

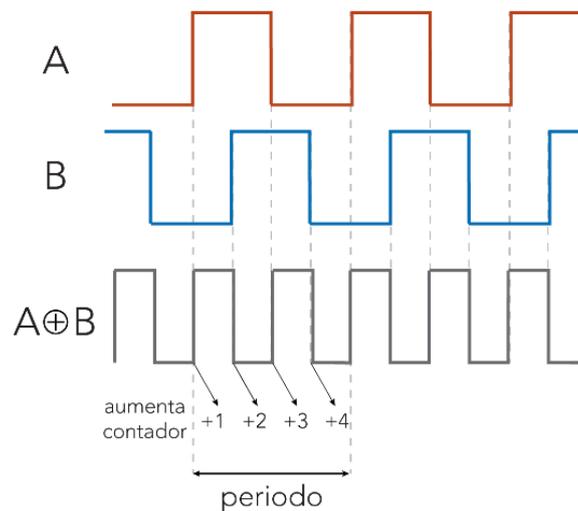
El funcionamiento de los encoders incrementales se basa en la generación de 2 señales cuadradas mediante la obstrucción y el paso de la luz a través de un disco dentado en su interior. Estas señales, comúnmente denominadas A y B se encuentran en cuadratura, lo que quiere decir que están desfasadas  $90^\circ$  entre sí. Esto permite determinar el sentido de giro del eje del encoder dependiendo del orden de los pulsos obtenidos de las dos señales. De manera convencional, cuando A presenta un flanco positivo antes que B se considera como dirección de giro horaria, mientras que el caso contrario es para giro antihorario. En

la Figura 2.5. se muestra un gráfico que ilustra el funcionamiento interno de los encoders y la convención del sentido de giro dependiendo del comportamiento de las señales.



**Figura 2.5.** Funcionamiento interno de los encoders incrementales y señales obtenidas.

Para aumentar la resolución efectiva de los encoders se puede realizar una operación XOR entre las señales A y B lo cual genera una señal del doble de la frecuencia. Con esta señal se pueden contar 4 valores en cada periodo de las señales originales lo cual aumenta la resolución en un factor de 4, este procedimiento se indica en la Figura 2.6.



**Figura 2.6.** La operación XOR entre las señales A y B del encoder aumenta la resolución efectiva en un factor de 4.

En el sistema implementado se realiza este método mediante el uso del parámetro `Timer.ENC_AB`. Como se mencionó en los párrafos anteriores, se pueden contar hasta

4000 pulsos por cada revolución, por esto la conversión entre grados y pulsos se da por la siguiente relación:

$$360 \frac{^\circ}{rev} \times \frac{1}{4000} \frac{rev}{pulso} = 0.09 \frac{^\circ}{pulso} \quad (2.2)$$

Este valor de conversión de 0.09 se utiliza para relacionar el número de pulsos entregados por el encoder con un valor de posición angular. Si, por ejemplo, en un momento dado el contador de pulsos tiene un valor 314 esto corresponde a un ángulo de 28.26°, esto en base a lo expresado en la Ecuación 2.2. Cabe recalcar que este método permite la medición de ángulos negativos ya que internamente en el microcontrolador se realiza el procesamiento para aumentar o reducir el conteo de pulsos dependiendo del sentido de giro del encoder, de acuerdo con lo explicado en la Figura 2.5.

#### 2.2.1.1.3. Motores brushless

Para generar el movimiento en la plataforma se han utilizado motores DC sin escobillas (también denominados brushless o BLDC por su nombre en inglés). Estos motores presentan una mayor eficiencia que los motores DC tradicionales pudiendo llegar hasta un valor de 90%. Otras ventajas que se pueden mencionar son: mayor relación torque/peso, mayor confiabilidad, menor ruido, mayor tiempo de vida útil ya que se elimina la erosión en las escobillas, etc [40].

Para escoger el motor se toma en cuenta principalmente tres factores que son: velocidad, tamaño y fuerza de empuje. Para el caso del helicóptero de 3-GDL, estos factores se resumen a continuación:

- **Tamaño:** dado que se trata de un banco de pruebas para representar el control de vuelo en drones multi-rotor se ha escogido un tamaño de motor reducido con diámetro de entre 25 y 35 mm, que son tamaños estándar. Los más comunes en este rango son los de 27-28 mm por lo que su costo también es reducido.
- **Fuerza de empuje:** como se explicará en las siguientes secciones, el cuerpo del helicóptero tiene una masa neta de alrededor de 900g, lo que quiere decir que las dos hélices en conjunto deben poder ejercer un empuje de aproximadamente 9N para poder sostener el peso del helicóptero.
- **Velocidad:** la velocidad de giro necesaria depende de la hélice que se escoja para acoplar al motor. En el presente proyecto se han escogido las hélices de tipo 1045 que a 10.000RPM pueden generar alrededor de 800g de empuje [41] con lo cual se puede levantar el peso necesario ya que se tienen 2 motores.

Para esta especificación, adicionalmente, se considera el factor KV de los motores brushless que indica la cantidad de revoluciones por minuto (RPM) por cada voltio aplicado. Por ejemplo, un motor de 1000KV generará 10.000RPM cuando funciona a 10V. Con base en estas especificaciones se puede escoger un motor de entre 900 y 1500KV ya que a 12V se pueden generar entre 10.800 y 18.000RPM lo cual produce más de 800g de empuje con las hélices mencionadas. Con esto, una posibilidad para esta aplicación es el motor A2212 de 1000KV que se puede conseguir fácilmente en el mercado, gracias a lo cual, su costo es muy accesible. En la Figura 2.7 se muestra una imagen del motor seleccionado para el prototipo del presente proyecto junto con la montura y el acople para la hélice 1045.



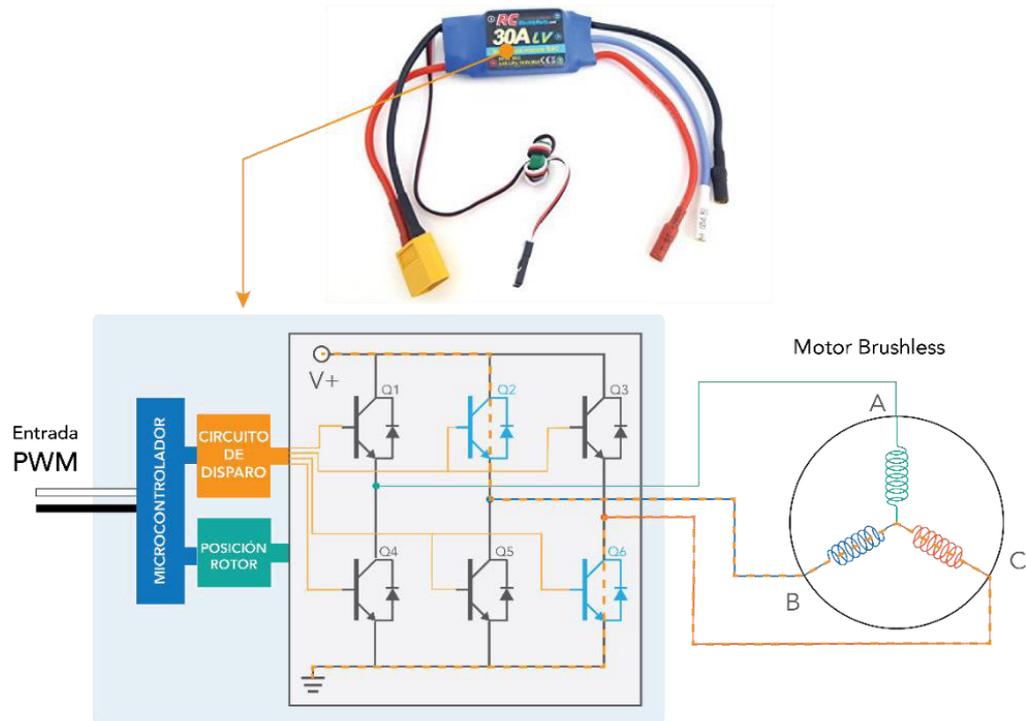
**Figura 2.7.** Motores brushless A2212 de 1000KV con los acoples para montura y hélices 1045. Imagen obtenida de [41].

#### 2.2.1.1.4. Controladores de velocidad ESC

Internamente, el motor sin escobillas o brushless posee una construcción muy parecida a la de un motor sincrónico de imanes permanentes ya que, en ambos casos, hay un estator bobinado y un rotor con imanes permanentes que sigue al campo rotativo del estator. La diferencia radica en la forma y la generación del voltaje en el estator.

Para conseguir el campo magnético rotatorio en el motor brushless, las bobinas del estator son alimentadas de forma cíclica lo cual se logra por medio de un pequeño inversor que transforma la corriente DC en AC. El inversor mencionado forma parte de un dispositivo llamado Controlador Electrónico de Velocidad o ESC por sus siglas en inglés. Existen una gran variedad de ESCs de diferentes marcas, tipos y especificaciones. Sin embargo, el parámetro más importante que se debe considerar es la corriente necesaria. Para el tipo de motor mencionado (A2212, 1000KV), el controlador recomendado por los fabricantes es

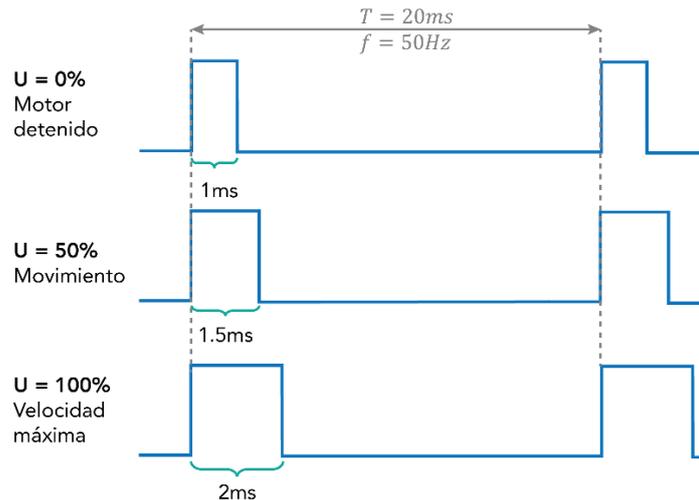
de 30A, que es el utilizado en el presente proyecto. En la Figura 2.8 se muestra un diagrama del funcionamiento de los ESCs junto con el modelo utilizado de la marca RC Electric Parts [42].



**Figura 2.8.** ESC de la marca RC Electric Parts junto con un diagrama simplificado del circuito de funcionamiento del controlador. Como se puede observar, la entrada del sistema es una señal PWM y a la salida se tiene un voltaje trifásico que se consigue mediante un circuito inversor en el interior del ESC.

Para modificar la velocidad de giro del motor brushless se controla el ESC mediante una señal PWM que posee ciertas características específicas, que se detallan a continuación:

- **Frecuencia:** la frecuencia a la cual trabajan los ESC es un valor estandarizado de 50Hz. Este valor se configura fácilmente en Micropython con el parámetro `freq` en la configuración de los objetos de tipo `Timer()`.
- **Ancho de pulso:** el ancho de pulso es el parámetro de control de velocidad. También posee un valor estándar que varía desde 1ms (0% de la señal de control) hasta 2ms (100% de la señal de control), tal como se indica en la Figura 2.9



**Figura 2.9.** Características de la señal PWM para control de velocidad mediante ESCs. La letra U se utiliza para representar la señal de control en porcentaje.

- **Implementación de la señal PWM:**

En el microcontrolador se escoge un Timer y se configura la frecuencia a 50Hz, en este caso se usa el Timer 12. Posteriormente se conectan los dos canales de este temporizador a las salidas correspondientes y se les asigna a un objeto, como se observa en el siguiente código, que en este caso llevan el nombre de ESC\_B y ESC\_F para los motores posterior y frontal, respectivamente.

```
# Configuración del Timer 12 a 50Hz:
timESC = Timer(12, freq=50)
# Configuración de los pines Y7 y Y8 como salidas del timer para
PWM. Esto permite tratar ambas salidas como una variable
independiente:
ESC_B = timESC.channel(1, Timer.PWM, pin=Pin('Y7'))
ESC_F = timESC.channel(2, Timer.PWM, pin=Pin('Y8'))
# Cálculo de las acciones de control...
```

Para generar el ancho de pulso de 1 a 2 ms se utiliza la función `pulse_width_percent(duty)` donde `duty` es el valor correspondiente en porcentaje. Dado que se tiene una PWM de 20ms (50Hz) el porcentaje debe variar entre el 5% y el 10% ya que corresponde a pulsos de 1 y 2 ms, respectivamente. Para esto se hace la conversión como se muestra en el siguiente código, donde `MotorB` y `MotorF` son variables que varían de 0 a 100%.

```

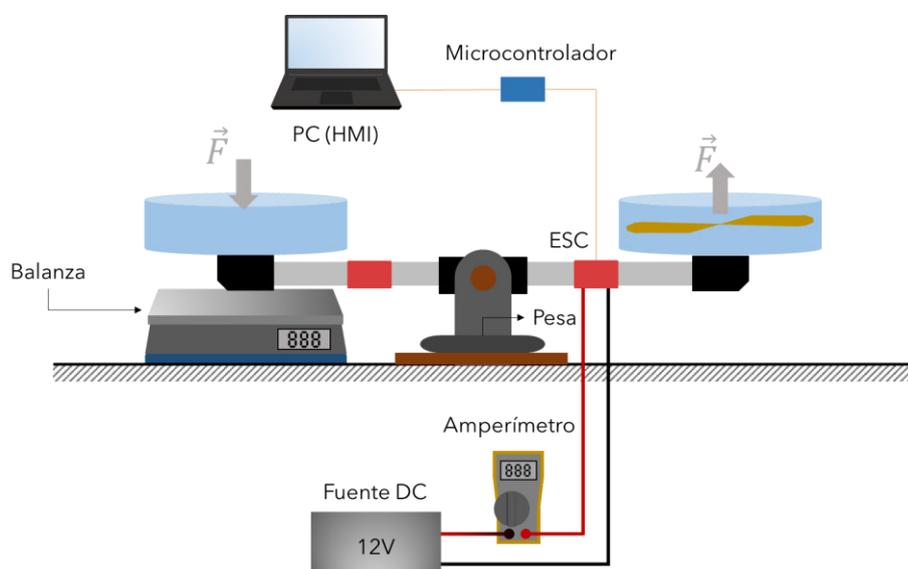
dutyB = MotorB*0.05 + 5 # Motor posterior
dutyF = MotorF*0.05 + 5 # Motor frontal
# Se mandan Los anchos de pulso convertidos:
ESC_B.pulse_width_percent(dutyB)
ESC_F.pulse_width_percent(dutyF)

```

Como se puede observar, las variables MotorB y MotorF se multiplican por 0.05, con lo cual se asegura que están en un rango de [0; 5] que al sumar 5 se convierte a [5; 10] que son exactamente los límites necesarios para generar los pulsos entre 1 y 2ms de ancho.

#### 2.2.1.1.5. Relación entre velocidad y fuerza de empuje

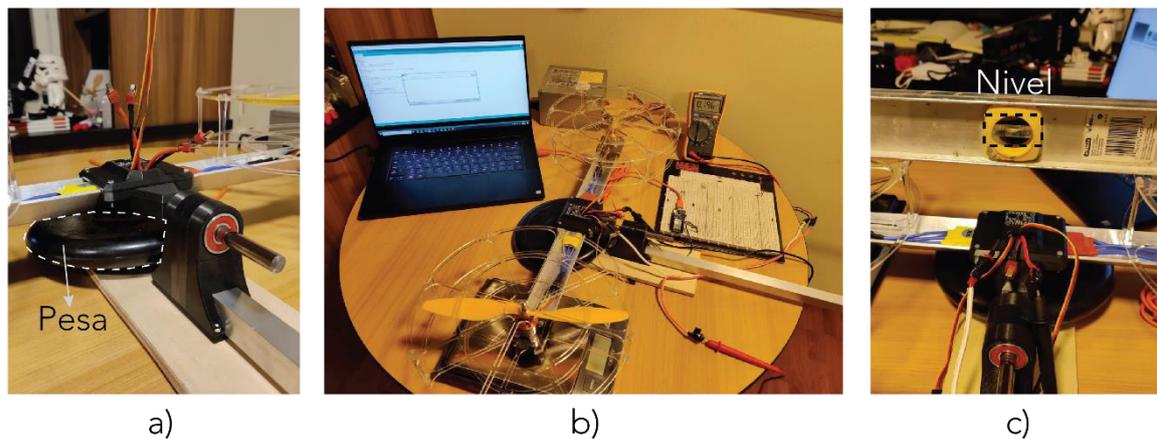
La fuerza de empuje generada por las hélices y la velocidad de giro de las mismas no poseen una relación lineal, por lo cual es necesario encontrar qué característica presentan para poder implementar las acciones de control. Esto se debe a que las fuerzas de los motores son las entradas del sistema, como se explicará en la sección del modelo matemático. Para encontrar dicha relación es necesario medir la fuerza generada por la hélice a la vez que se aumenta progresivamente la señal de control, la cual es proporcional a la velocidad de rotación. Por lo que, para realizar esta prueba se ha utilizado el arreglo que se muestra en la Figura 2.10, el cual se basa en el trabajo realizado en [43]. Dado que la fuerza de empuje que genera la hélice es perpendicular a su plano de giro y las dos hélices del helicóptero se encuentran paralelas entre sí, la fuerza generada por una hélice hacia arriba se podrá medir en la otra hélice con dirección contraria; para lo cual se utiliza una balanza electrónica.



**Figura 2.10.** Diagrama del arreglo elaborado para la medición del coeficiente de empuje.

El arreglo para el eje de giro consta del soporte del eje pitch atornillado a una tabla con una pesa de 2.5kg, para evitar el despegue de la hélice. La balanza se encuentra elevada a una altura de aproximadamente 5cm para asegurar que ambas hélices se encuentren correctamente niveladas y así poder medir adecuadamente la fuerza total de empuje.

Con el arreglo explicado en el párrafo anterior se procede a aumentar progresivamente la señal de control, que se asume proporcional a la velocidad angular del motor, y se toman las mediciones de “masa” indicadas por la balanza. En la Figura 2.11 se muestra el sistema elaborado.



**Figura 2.11.** Arreglo realizado para medir el coeficiente de empuje. a) Pesa para evitar el despegue de la hélice, b) sistema completo, c) el eje de las hélices se encuentra nivelado de manera horizontal.

Cabe mencionar que internamente la balanza relaciona la fuerza aplicada sobre su superficie con la masa del cuerpo de acuerdo con la relación  $m = P/g$ , donde  $m$  es la masa en gramos y  $g$  es la aceleración de la gravedad que tiene un valor de  $9.81 \text{ m/s}^2$ , de modo que, para despejar la fuerza de empuje, que en este caso corresponde al peso,  $P$  solo basta tomar la lectura de masa de la balanza y multiplicarla por  $g$ . Adicionalmente se ha medido la corriente que circula a través del ESC para los diferentes valores de la señal de control, esto con el propósito de verificar los valores máximos de corriente que puede manejar el sistema. Los resultados obtenidos se muestran en la Tabla 2.4.

Como se puede observar en la Tabla 2.4, solo se han tomado datos hasta el 65% del valor de la PWM (un ancho de pulso de 1.65ms) debido a que en este valor ya se obtuvo una corriente de casi 9A y el amperímetro utilizado está limitado a 10A; es decir, básicamente como medida de precaución.

**Tabla 2.4.** Datos obtenidos de la fuerza de empuje para diferentes valores de la señal de control PWM en porcentaje.

PWM [%]	Masa [g]	Fuerza de empuje [N]	Corriente [A]
0	0	0	0.045
8	30	0.2943	0.233
10	44	0.43164	0.303
15	78	0.76518	0.491
20	126	1.23606	0.753
25	184	1.80504	1.17
30	250	2.4525	1.724
35	318	3.11958	2.356
40	390	3.8259	3.097
45	452	4.43412	3.963
50	530	5.1993	4.91
55	616	6.04296	6.025
60	670	6.5727	7.33
65	764	7.49484	8.66

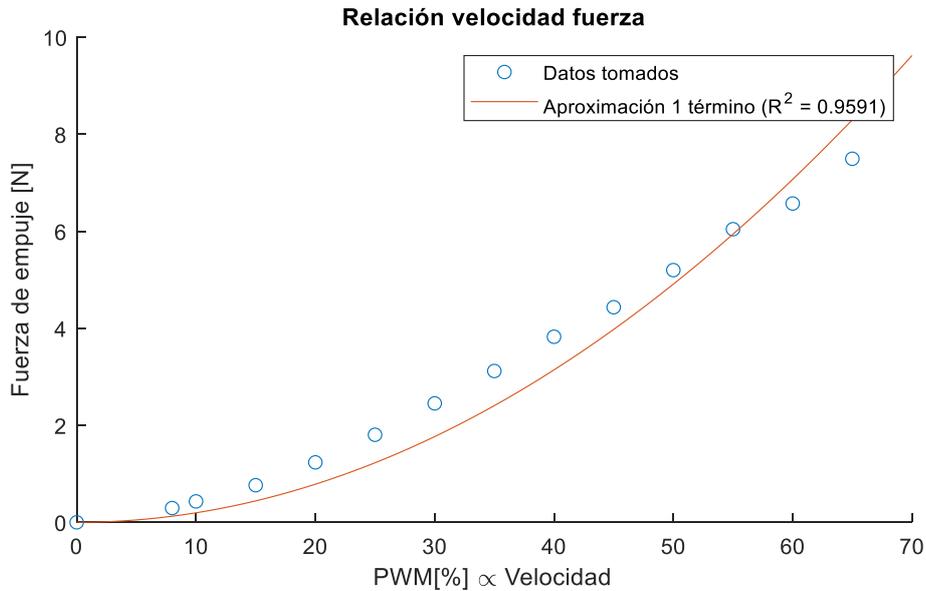
A partir de los datos obtenidos con las pruebas realizadas se utiliza la herramienta de ajuste de curva (Curve Fitting) de Matlab para encontrar una aproximación del coeficiente de empuje considerando que la relación entre la fuerza de empuje y la velocidad de rotación es una curva de segundo orden. Es decir, una relación como la de la ecuación (2.3).

$$F(U) = aU^2 + bU + c \quad (2.3)$$

Donde  $F$  es la fuerza en Newtons generada por las hélices y  $U$  es la señal de control en porcentaje enviada a los ESCs. Los coeficientes  $a$ ,  $b$  y  $c$  determinan la relación entre  $F$  y  $U$ .

Para este resultado se ha encontrado una curva con solamente el primer coeficiente de la ecuación cuadrática. Esto se hace debido a que este término es el de mayor influencia y con esto se reduce la complejidad de implementación en el microcontrolador. La curva en cuestión se muestra en la Figura 2.12 junto con los puntos de los datos obtenidos. En esta gráfica se muestra el valor del parámetro  $R^2$  denominado *coeficiente de determinación* e indica la relación que existe entre dos variables, que para este caso son los datos tomados

y la curva ajustada. Se considera un mejor ajuste cuando el valor de  $R^2$  se aproxima a 1.0 [44].



**Figura 2.12.** Datos obtenidos para el coeficiente de empuje junto con su aproximación cuadrática.

Mediante el uso de la herramienta Curve Fitting de Matlab se ha encontrado la curva cuadrática que se indica en la ecuación (2.4) de acuerdo con lo explicado en los párrafos anteriores. La gráfica de esta parábola se presenta en la Figura 2.12 en color rojo.

$$F(U) = 0.001964U^2 \text{ [N]} \quad (2.4)$$

Donde  $U$  es la señal de control del ESC en porcentaje y  $F$  es la fuerza de empuje en newtons. Esta expresión permite encontrar el valor de la señal enviada a los motores en porcentaje a partir de la acción de control calculada en Newtons tal como se indica en la ecuación (2.5).

$$U = \sqrt{\frac{1}{0.001964} F} = \sqrt{509.164 F} \text{ [%]} \quad (2.5)$$

### 2.2.1.2. Comunicación

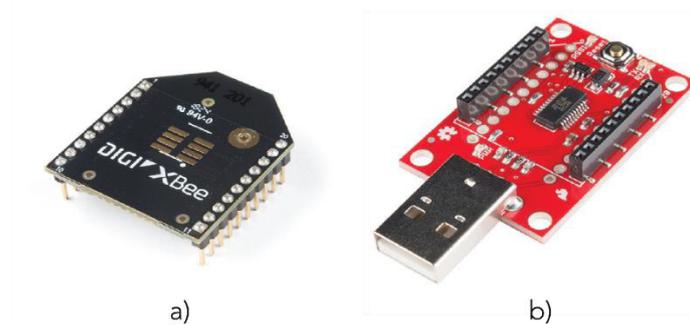
Para interactuar con la plataforma desde la PC es importante considerar que la variable travel tiene un rango de movimiento muy amplio, por lo que utilizar una interfaz de comunicación cableada no resulta factible. Para esto se utilizan módulos de comunicación

inalámbrica, con los cuales se puede transmitir señales desde de la plataforma a la computadora y viceversa, sin necesidad de cables que se puedan enredar.

#### 2.2.1.2.1. Módulos XBee

Para este propósito se han utilizado los módulos XBee 3 de la marca DIGI [45] como el que se muestra en la Figura 2.13. Estos módulos utilizan un protocolo de comunicación inalámbrica llamado Zigbee [46] y presentan algunas ventajas con respecto a módulos de bluetooth como las siguientes:

- Configuración sencilla: los módulos bluetooth (como por ejemplo HC-05 y 06) se configuran por medio de comandos AT en una interfaz de línea de comandos. En el caso de XBee, se tiene un programa gratuito propio del fabricante que permite acceder de forma sencilla a los parámetros de configuración para cada módulo.
- Rango de alcance: el rango aproximado para bluetooth es de alrededor de 10m, mientras que para XBee (Zigbee) es más de 100m. Esto asegura una mayor robustez para comunicación de corta distancia.
- Consumo de energía: comparado con los módulos de bluetooth tradicional, los XBee consumen menos de 1/3 de la potencia para su funcionamiento. Es decir, alrededor de 30mW mientras que para bluetooth es 100mW [47].

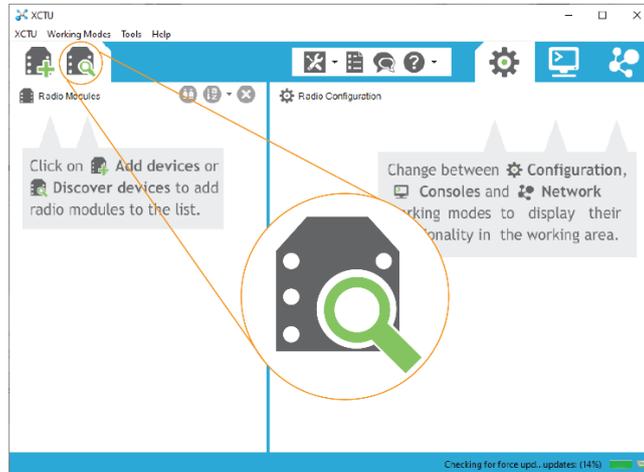


**Figura 2.13.** Componentes para comunicación inalámbrica. a) Módulo XBee 3 con antena PCB, b) "Dongle" USB para conectar el módulo XBee a la computadora.

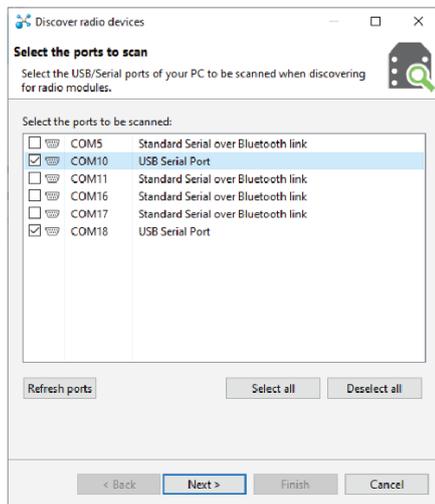
#### 2.2.1.2.2. Configuración de módulos XBee

Para establecer comunicación entre dos o más módulos XBee se debe configurar cada dispositivo mediante el software XCTU que ofrece el fabricante DIGI de forma gratuita. Para esto se conecta el módulo mediante USB a la computadora utilizando un adaptador como el que se muestra en la Figura 2.13. b).

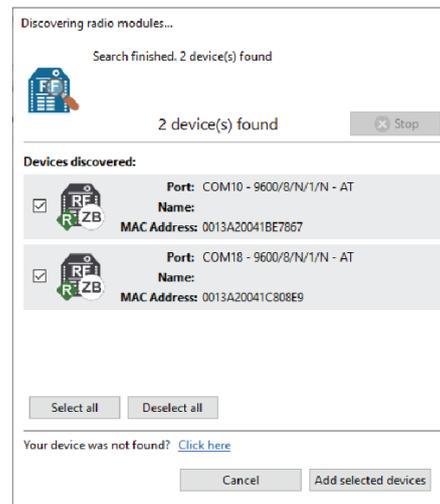
Luego de descargar e instalar el programa DIGI XCTU, se abre la aplicación, donde se muestra la pantalla de la Figura 2.14.a). Aquí se pueden agregar los dispositivos de forma manual o con el buscador dando clic en el ícono que se indica. Posteriormente se seleccionan los puertos USB donde se han conectado los nuevos dispositivos (Figura 2.14.b) y finalmente se escoge el dispositivo que se quiere configurar (Figura 2.14.c).



a)



b)



c)

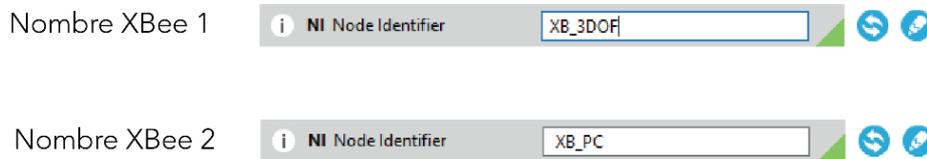
**Figura 2.14.** Procedimiento para agregar dispositivos para la configuración en XCTU.

a) Pantalla de inicio e ícono de buscar dispositivos. b) Pantalla de selección de puertos.

c) Pantalla de dispositivos encontrados.

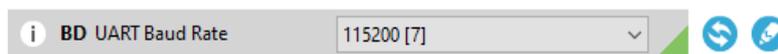
Luego de encontrar y seleccionar el dispositivo que se debe configurar, se tiene acceso a una gran cantidad de parámetros para la configuración, sin embargo, para establecer la comunicación entre únicamente 2 módulos (que es el caso para esta aplicación: PC y microcontrolador), los parámetros que se deben configurar son los que se indican a continuación.

- **Nombre:** el nombre del dispositivo se modifica en el parámetro NI (node identifier). En la Figura 2.15 se muestra la configuración de los dos dispositivos: XB\_3DOF (para el módulo de la plataforma) y XB\_PC (para el módulo conectado a la computadora).



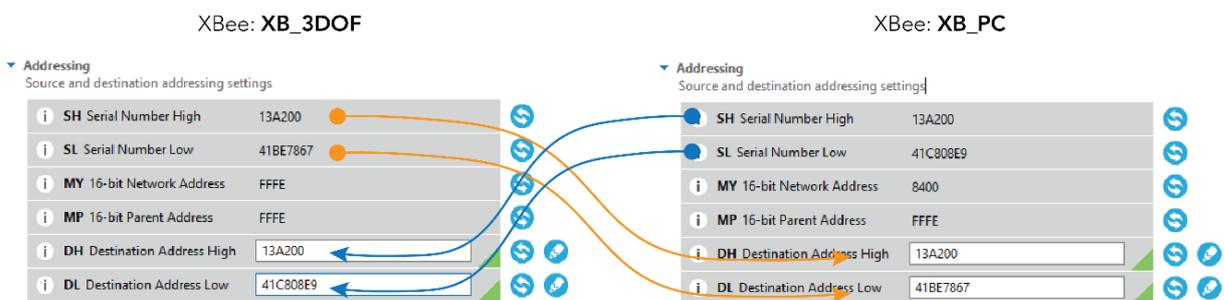
**Figura 2.15.** Configuración de los nombres para ambos módulos. La esquina verde en la parte inferior derecha indica que es un parámetro no guardado.

- **Baudios (baud rate):** este es el parámetro mediante el cual se interactúa con el módulo mediante comunicación UART. Cabe recalcar que este valor no se refiere a la comunicación inalámbrica entre los módulos, sino entre el módulo y el microcontrolador o la PC. Se modifica mediante el parámetro BD (UART Baud Rate). Para este ejemplo se configuran ambos módulos a un valor de 115200 como se muestra en la Figura 2.16.



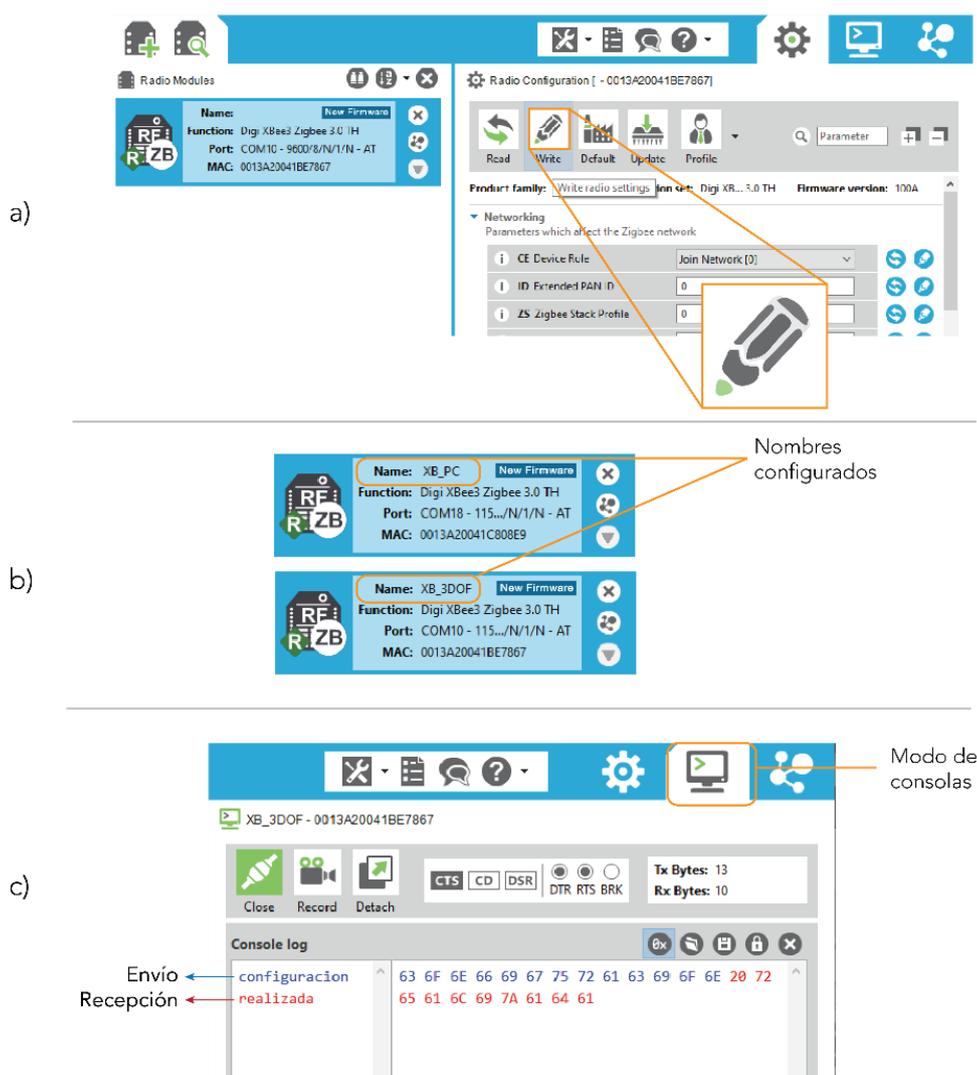
**Figura 2.16.** Configuración del parámetro BD (Baud Rate), se utiliza un valor de 115200 baudios.

- **Dirección:** finalmente, los parámetros que establecen la comunicación entre los dos módulos son los valores de SH, SL (serial number high y low) y DH, DL (destination address high y low). Para poder enlazar dos dispositivos se deben copiar el valor de SH del primer dispositivo en el DH del segundo y viceversa. Este procedimiento se ilustra en la Figura 2.17:



**Figura 2.17.** Para establecer la comunicación entre dos dispositivos XBee se copian las direcciones de SH y SL del primer módulo en DH y DL del segundo, respectivamente.

Finalmente, para guardar las configuraciones realizadas se debe dar clic en el botón Write, como se indica en la Figura 2.18.a). Luego de esto se podrá ver que los ajustes realizados tuvieron efecto ya que el nombre de los dispositivos se cambiará al valor determinado, como se muestra en la Figura 2.18.b). Por último, para comprobar que la configuración fue exitosa, se puede abrir el modo consola que incluye el programa XCTU, abrir el puerto de comunicación y comprobar que los caracteres que se escriben se envían al otro dispositivo. Esto se indica en la Figura 2.18.c) donde los caracteres enviados se muestran en color azul, mientras que los recibidos en rojo, comprobando así el enlace de comunicación correcto entre ambos módulos.



**Figura 2.18.** Configuración de módulos XBee. a) Para escribir los parámetros modificados se da clic en el botón Write. b) Luego de escribir los nuevos parámetros, los nombres de los dispositivos cambian a los valores especificados. c) Se puede comprobar el estado de la comunicación mediante la consola que incluye XCTU.

### 2.2.1.3. Alimentación

Para la alimentación del sistema se ha considerado un voltaje de 12V ya que es el valor al cual trabajan los motores brushless. Este voltaje también permite alimentar a los encoders (que acepta un rango de 8-24V) y a la placa Pyboard (que permite hasta 16V). Además, la corriente máxima que puede soportar el sistema es de 30A ya que, de igual manera, es la especificación máxima del ESC. Cabe recalcar que esta es una especificación a la cual se puede llegar durante breves periodos, es decir, el consumo continuo de corriente en realidad será mucho menor. Por esta razón se ha tomado la corriente de 30A como el valor máximo para todo el sistema.

#### 2.2.1.3.1. Fuente de alimentación

Dicho esto, el elemento más importante para entregar la energía que necesita el sistema es la fuente. Para esto se ha escogido una fuente switching regulada de 12V/30A como la que se indica en la Figura 2.19. Una característica adicional que presenta este tipo de fuentes es su tamaño reducido ya que tienen un grosor de apenas 5cm.



**Figura 2.19.** Fuente switching de 12V/30A y sus dimensiones.

#### 2.2.1.3.2. Anillo deslizante

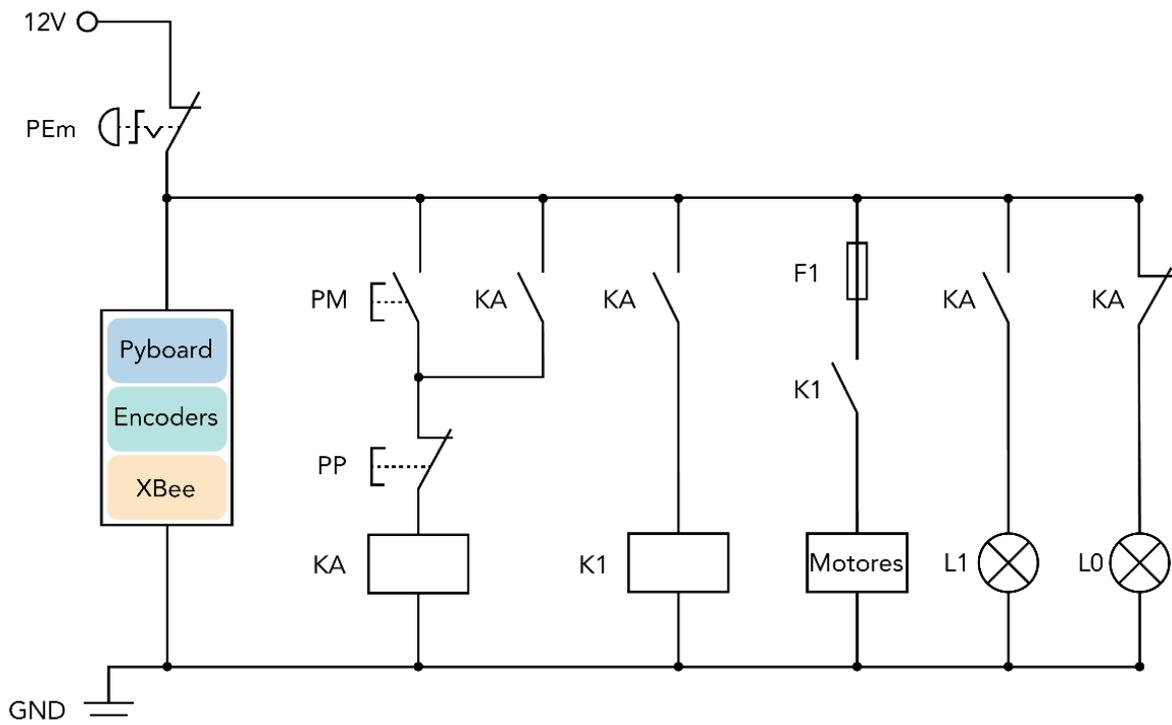
Como se ha mencionado en las secciones anteriores, el rango de movimiento del eje travel de la plataforma es muy amplio, por lo que usar una conexión directa mediante cables normales no resulta factible ya que se enredarían. Por esta razón se utiliza un dispositivo denominado anillo deslizante, el cual permite transmitir una señal eléctrica a lo largo de un eje rotacional. Para la presente aplicación se ha utilizado uno de 2 vías (para 12V y GND de la fuente) de 30<sup>a</sup>, ya que es la corriente que debe soportar para alimentar el sistema. El dispositivo escogido es de la marca Taidacent [48] y se indica en la Figura 2.20.



**Figura 2.20.** Anillo deslizante Taidacent de 2 vias, 30A. Imagen obtenida de [48].

### 2.2.1.3.3. Módulo de encendido

Dado que los motores son los elementos que más potencia consumen (y cuyo manejo inadecuado puede causar, potencialmente, mayor daño) se ha separado el encendido de los motores del resto del sistema electrónico (microcontrolador, módulos de comunicación y encoders). Esto permite verificar el estado de los sensores, comunicación y microcontrolador sin la necesidad de mantener los motores en funcionamiento. Adicionalmente, el uso de un sistema de encendido independiente permite asegurar que, en caso de existir un corte de energía inesperado, los motores no arrancarán de inmediato al volver la energía, lo cual representa una medida de seguridad para el sistema y el usuario.



**Figura 2.21.** Diagrama circuital del módulo de encendido.

El sistema de encendido consta de un botón de paro de emergencia (denotado como PEm en la Figura 2.21). Por otro lado, módulo de encendido de los motores se compone de un sistema de marcha-paro utilizando dos relés: uno para el mando memorizado y otro para conectar la fuente con los motores. Además, se han utilizado LEDs que funcionan como indicadores luminosos del encendido del sistema. El diagrama del sistema de encendido se muestra en la Figura 2.21.

El pulsador PEm es el botón de emergencia (normalmente cerrado), cuando se encuentra desactivado (cerrado) sirve para alimentar todo el sistema electrónico y el módulo de encendido de los motores. El botón PM es el pulsador de encendido o marcha (normalmente abierto), este permite la energización de la bobina auxiliar KA que se mantiene encendida debido al mando memorizado, y el pulsador PP sirve para apagar los motores. A su vez, la bobina KA enciende la bobina K1, cuyos contactos deben ser capaces de soportar los 30A del sistema por lo que se ha utilizado un relé automotriz de tipo G8HN ya que permite manejar hasta corrientes de 45A. Como protección se tiene el fusible F1 que corta la alimentación de los motores en caso de existir un cortocircuito. Además, se han incluido los indicadores luminosos (LED) que indican el estado del encendido:

- L0 (LED amarillo): alimentación conectada (PEm desactivado) sin alimentación en los motores
- L1 (LED verde): alimentación y motores conectados.

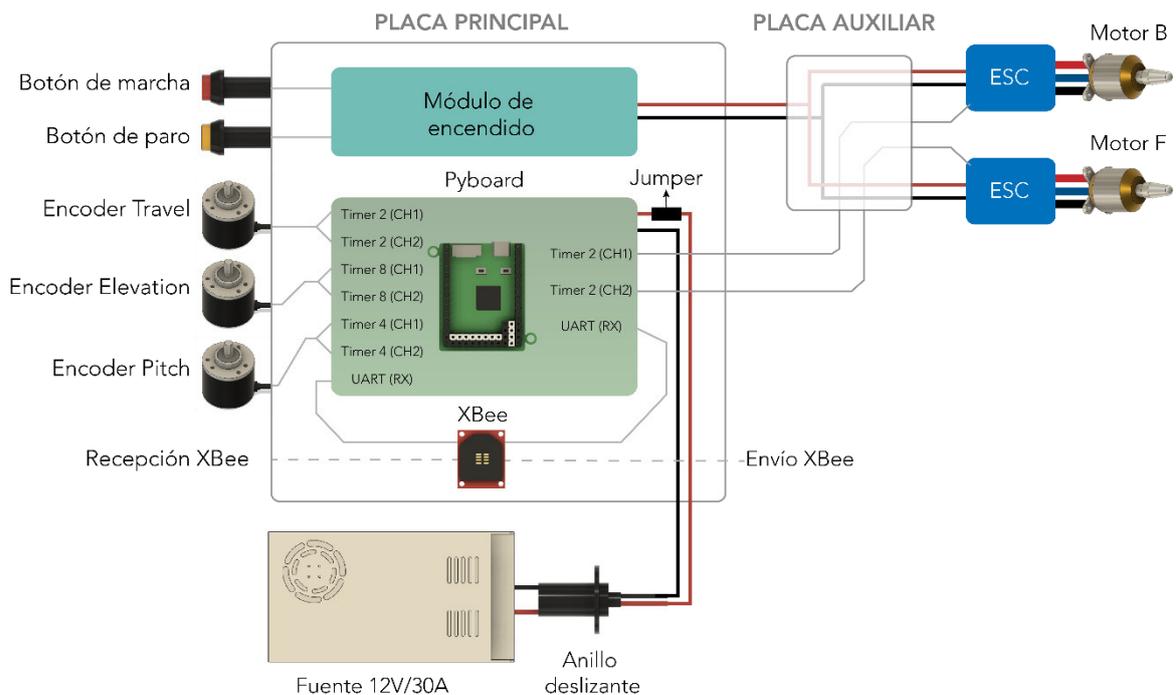
#### **2.2.1.4. Diseño de tarjetas electrónicas**

La integración de todos los componentes se ha realizado por medio de placas de circuito impreso (PCBs), ya que permiten interconectar todos los elementos descritos anteriormente de manera efectiva y confiable sin depender de cables o soldaduras que puedan desconectarse fácilmente y comprometer el funcionamiento del sistema. En la Figura 2.22 se muestra un diagrama de la conexión de todos los elementos descritos en los párrafos anteriores, incluyendo el módulo de encendido.

Para el diseño de las placas de circuito impreso se ha utilizado el software Eagle de Autodesk. Se ha empleado este programa debido a que presenta algunas ventajas considerables:

- Librerías: existen muchos repositorios y recursos en línea desde los cuales se pueden descargar librerías de forma gratuita. Algunos ejemplos son librerías de fabricantes como Sparkfun o Adafruit. Para este proyecto, es de primordial importancia la librería para la placa Pyboard, cuya versión oficial se puede conseguir en Github [49].

- Compatibilidad con Fusion 360: dado que Eagle pertenece a la marca Autodesk se puede relacionar con el software de diseño 3D Fusion 360, gracias a lo cual el diseño de la PCB puede ser fácilmente exportado a un modelo 3D con lo cual se pueden elaborar carcasas y otros elementos.
- Costo: mediante la dirección de correo electrónico terminada en “.edu” se puede acceder de forma gratuita a la versión estudiantil de este software, por lo que resulta de gran conveniencia ya que se puede hacer uso de todas las funcionalidades que ofrece.



**Figura 2.22.** Diagrama de entradas y salidas de la placa principal, a la derecha se representa el funcionamiento de la placa auxiliar. En el anexo A.1 se encuentran los diagramas circuitales y de placa de las PCBs diseñadas.

#### 2.2.1.4.1. Placa principal

Considerando todos los elementos mencionados en los párrafos anteriores, se ha diseñado la placa principal para el control de la plataforma de 3GDL. Una lista de los componentes utilizados se indica en la Tabla 2.5, junto con su descripción y cantidad.

Para los encoders se han utilizado conectores de 4 pines: A, B, 12V y GND. De igual manera, en el caso del módulo XBee se empleó un conector de 4 pines que corresponde al adaptador para UART: Tx, Rx, 3.3V y GND. Para la conexión de los motores se utilizó un conector RJ-45 utilizando 4 pines: MotorB, MotorF y dos cables para GND, esto permite

diferenciar entre los conectores de los motores y los de los encoders. Por último, para la alimentación de la fuente y los motores se han escogido conectores XT60 debido a la corriente que pueden soportar. Todos los diagramas circuitales elaborados en Eagle se muestran en el Anexo A con los elementos descritos.

**Tabla 2.5.** Lista de elementos de la placa principal.

Elemento	Cantidad	Características y descripción
Pyboard	1	Espadines hembra para conectar la placa de control
Pulsador NO	1	Conexión con cables. Pulsante de marcha
Pulsador NC	1	Conexión con cables. Pulsante de paro
Conector 4 pines	3	Conector MOLEX-1X4 (KK). Conexión para encoders
Conector XBee	1	Espadines hembra para adaptador de 4 pines
Conector RJ45	1	No blindado. Conexión de señales PWM para motores
LED	3	LEDs verde, amarillo y rojo. Indicadores de alimentación
Espadín 3 pines	1	Cambiar de alimentación de fuente o USB para Pyboard
Conectores alimentación	4	“Footprint” de conector XT60. Conexión para la fuente, motores, fusible y contacto del relé.
Relé de mando memorizado	1	Relé tipo 4052 de 5A para el mando memorizado del módulo de encendido
Relé de alimentación	1	Relé automotriz tipo G8HN de 30A para alimentación de los motores
Resistencia 1K $\Omega$	3	Resistencias imitadoras para indicadores LED

Como se puede observar en la Tabla 2.5, se incluye un espadín que permite la conexión y desconexión de la placa Pyboard a la fuente de 12V mediante un *jumper*. Esta funcionalidad se ha incluido debido a que se puede conectar la placa mediante un cable USB a la computadora para programar el microcontrolador, en cuyo caso la fuente debe estar desconectada ya que la alimentación se toma desde el USB. Para poder conectar y desconectar la fuente fácilmente se ha utilizado un espadín de 3 pines, de los cuales 2 son para la conexión de la fuente y el tercero es para un indicador luminoso de color rojo que muestra si la placa se encuentra conectada o no a la fuente de 12V.

A partir del diagrama circuitual se rutean las conexiones para los dispositivos empleados. Todas las rutas poseen un grosor de 20mil (0.508mm) a excepción de las conexiones entre los motores y la fuente que son de 100mil (2.54mm) para poder soportar la corriente que necesitan los motores. Adicionalmente, se ha añadido un plano de tierra que sirve como

vía de retorno de la corriente de los componentes y además permite realizar las conexiones a GND con mayor facilidad. Estas consideraciones de diseño se pueden observar en las capturas de las placas diseñadas que se presentan en el Anexo A.

#### 2.2.1.4.2. Placa auxiliar

Esta placa sirve para transmitir la alimentación de 12V/30A las señales de control a los motores ubicados sobre el cuerpo del helicóptero. Esta PCB es de mucho menor complejidad ya que su propósito solo es unir a los motores con los cables que pasan a través del brazo de soporte del prototipo. Los elementos utilizados en esta placa se indican en la Tabla 2.6.

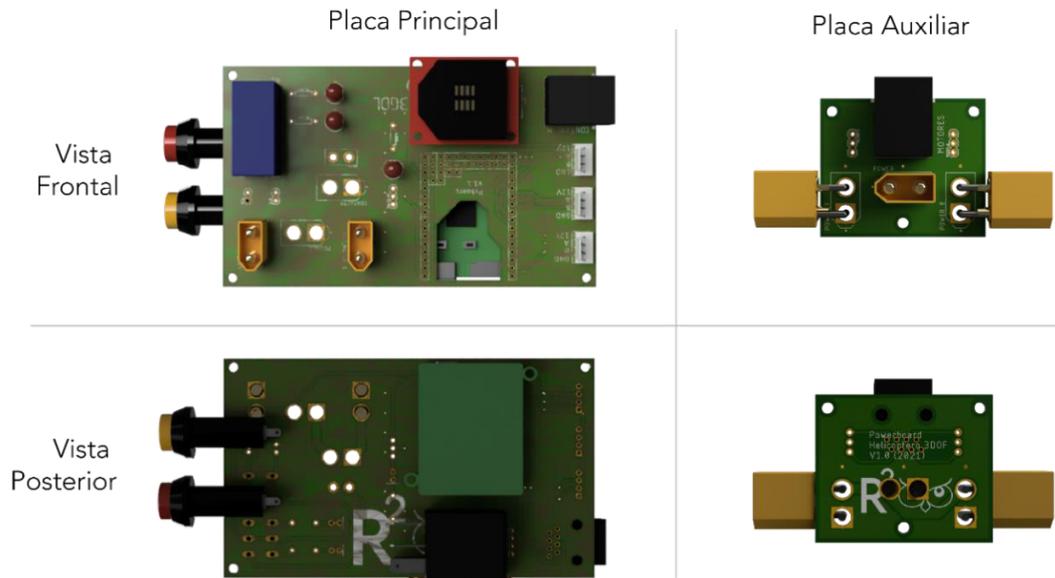
**Tabla 2.6.** Lista de elementos de la placa auxiliar.

Elemento	Cantidad	Características y descripción
Conectores alimentación	3	Conectores XT60. Conexión para la fuente, y los dos ESCs
Conector RJ45	1	No blindado. Conexión de señales PWM para motores
Espadines de 3 pines	2	Espadines para la conexión de las señales de control a los ESCs

Los únicos elementos utilizados en esta placa son un conector RJ-45 para las señales PWM, 3 conectores XT60 para direccionar la alimentación de la fuente a los motores y 2 jumpers para la conexión de las señales de control a los ESCs. El diagrama circuital y el diseño de esta PCB se indica en el Anexo A.2.

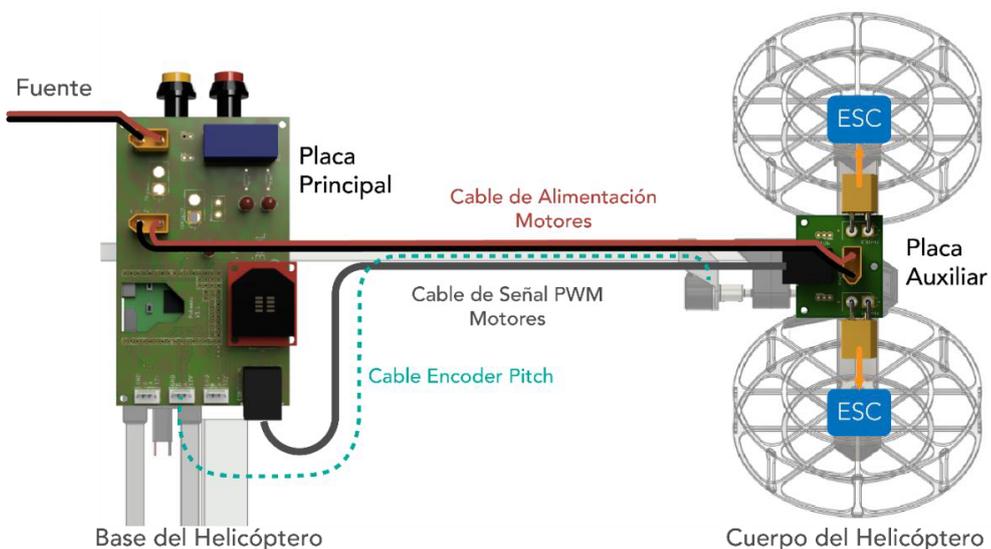
Como se mencionó anteriormente, desde Eagle se puede exportar un modelo en 3D para ser usado en Fusion 360. Esto permite visualizar las dimensiones de la placa y ciertas características como la ubicación de los componentes, el espacio que ocupan, etc. Para las placas diseñadas el modelo 3D se muestra en la Figura 2.23. Cabe mencionar que los modelos 3D de los componentes fueron elaborados individualmente en Fusion 360 ya que no se tenían dichos modelos en las librerías descargadas. En esta Figura, se presentan la vista frontal y posterior de ambas placas, con lo cual se pueden observar todos los elementos que las conforman. Cabe recalcar que, para el caso de la placa principal tanto el relé de 30A (que se encuentra en la vista posterior de color negro) como los pulsadores de marcha y paro se conectan a la placa mediante cables soldados desde los terminales de los elementos a la placa. Esto se realiza debido al espacio que ocupan, lo cual debe ser considerado para la elaboración de la carcasa protectora, cuyo modelo 3D se presenta en

la sección de la construcción del prototipo. La obtención de los modelos de las placas sirve para tener una referencia de la posición en la que se encuentran los elementos electrónicos y el espacio que ocupan.



**Figura 2.23.** Modelos 3D de ambas placas y vistas frontal y posterior de cada una.

La utilidad de la placa auxiliar radica en la necesidad de conectar los cables que pasan desde la base del helicóptero hasta la parte móvil del cuerpo del mismo. Las conexiones se han organizado de esta manera debido a que existe una distancia relativamente larga (~75cm) entre la ubicación de la placa principal y los motores, por lo cual la conexión se asegura mediante la placa auxiliar. En la Figura 2.24 se observa un esquema de la ubicación de las placas y su interconexión.



**Figura 2.24.** Diagrama de conexión de las placas y su ubicación en el helicóptero.

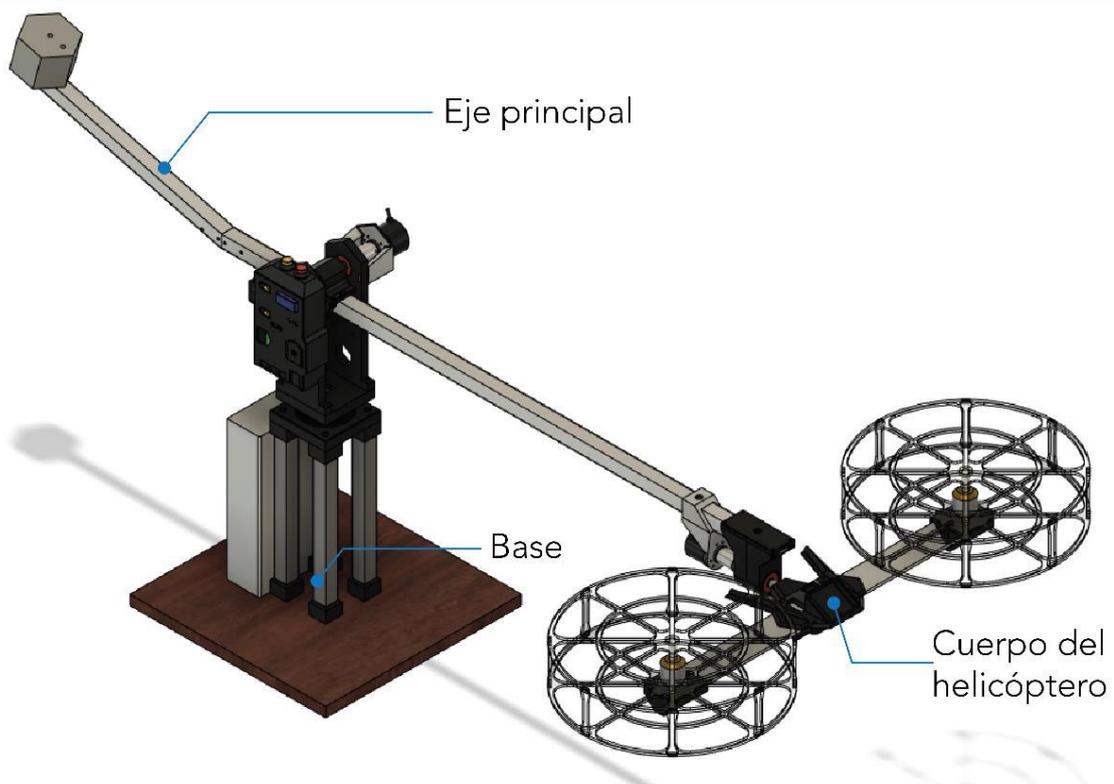
## 2.2.2. CONSTRUCCIÓN DEL PROTOTIPO MECÁNICO

Para la construcción del prototipo mecánico se han considerado diferentes materiales, principalmente debido a la facilidad de su obtención. Básicamente, estos materiales son: plástico PLA para piezas impresas en 3D, acrílico para los protectores de las hélices y tubos y perfiles de aluminio para brindar soporte a la estructura.

Todo el prototipo se conforma de 3 grandes partes principales que son:

- Cuerpo del helicóptero: consiste básicamente en los dos motores con sus protectores y un perfil de aluminio que los une. Además, une al helicóptero con el eje principal permitiendo el movimiento del ángulo pitch.
- Eje principal: se trata de la unión del brazo de soporte y el contrapeso. Es el componente que sujeta el cuerpo del helicóptero sobre el eje elevation.
- Base: es el soporte de toda la estructura. Sobre este se asienta el eje principal y permite el movimiento a lo largo del eje travel.

Al juntar las 3 partes mencionadas se obtiene el prototipo completo del helicóptero de 3-GDL elaborado. El modelo en 3D de este prototipo se indica en la Figura 2.25 indicando sus 3 partes principales.



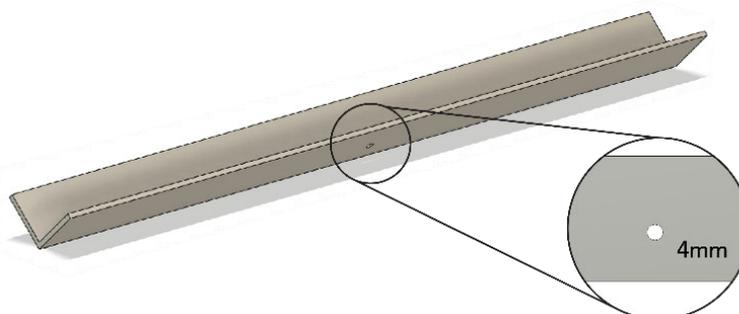
**Figura 2.25.** Modelo 3D del helicóptero de 3-GDL completo. El componente rectangular detrás de los apoyos de la base es la fuente de 12V.

En los siguientes párrafos se explicará con mayor detalle los componentes que conforman cada una de estas partes. Además de las descripciones que se presentan a continuación, en el Anexo B se muestran los planos de las piezas elaboradas con sus respectivas dimensiones

### 2.2.2.1. Cuerpo del helicóptero

#### 2.2.2.1.1. Barra central del helicóptero

La barra central es la pieza que sirve de soporte para todo el cuerpo del helicóptero. Consiste en un perfil de aluminio en forma de "L" de 31.7mm (1 ¼ ") de ancho y 3mm de grosor. La longitud de toda la barra es de 450mm. Se ha realizado un agujero de 4mm en la mitad de la barra como se indica en la Figura 2.26. Esto sirve para unir con el soporte central, el cual sostiene el eje de rotación del ángulo pitch.

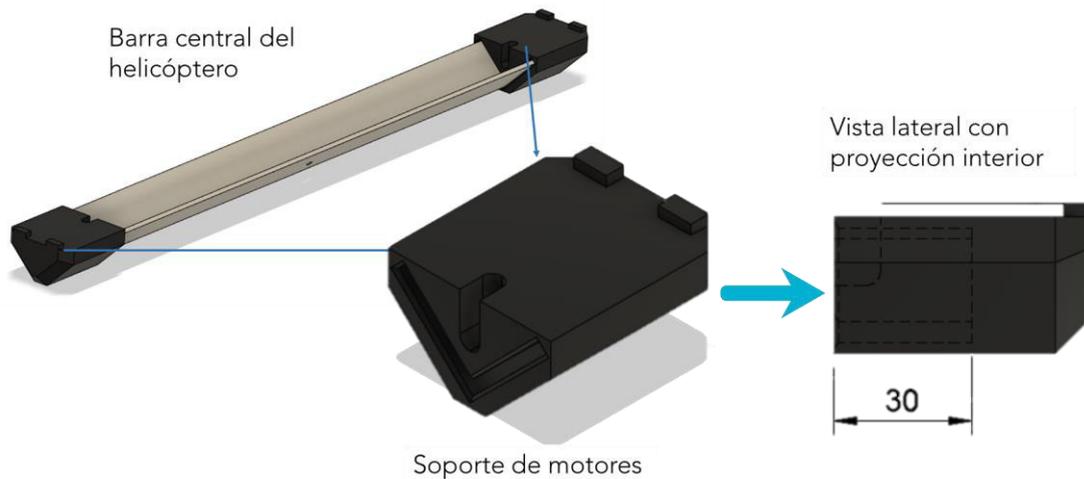


**Figura 2.26.** Vista 3D de la barra central del cuerpo del helicóptero y agujero para soporte central. Las dimensiones se muestran en el anexo B.

#### 2.2.2.1.2. Soporte de motores

Los soportes de motores son piezas impresas en filamento 3D (PLA) cuyo propósito es proporcionar soporte a los motores y unirlos a la barra central en los extremos. Se pueden usar diferentes tipos de plástico como PET y ABS, los cuales ofrecen mayor resistencia mecánica y térmica, sin embargo, el PLA soporta los pesos necesarios y es una alternativa de menor costo.

Esta pieza posee una ranura con forma de "L" para unirse con la barra central como se puede observar en la Figura 2.27. Dado que el perfil de aluminio posee dimensiones determinadas, para poder elaborar el modelo 3D de esta pieza se deben medir las partes ya existentes, para lo cual se ha utilizado un calibrador.

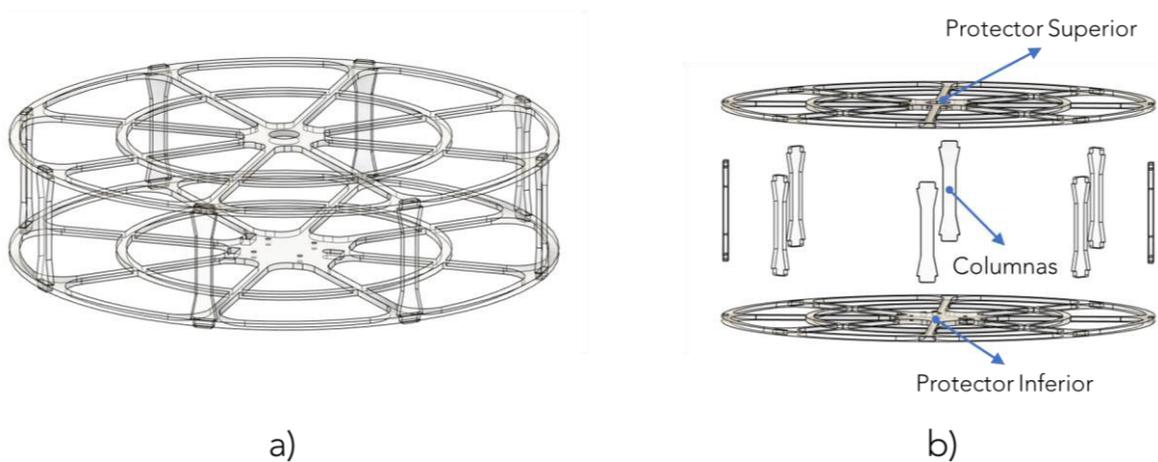


**Figura 2.27.** Vista 3D del soporte de los motores y su colocación con la barra central. Las dimensiones se muestran en el anexo B.

El perfil de aluminio de la barra central se introduce 30mm en el soporte de motor como se ilustra en la Figura 2.27, lo cual le proporciona la sujeción necesaria para mantenerse acoplado al cuerpo del helicóptero.

### 2.2.2.1.3. Protectores

Los protectores son estructuras que se colocan encima de los soportes de motor para proporcionar protección a la hélice (y al usuario) ya que puede impactarse contra objetos externos. El modelo en 3D de esta pieza se muestran en la Figura 2.28.



**Figura 2.28.** Protector de hélices. a) Vista general. b) Vista de despiece de las partes que lo conforman.

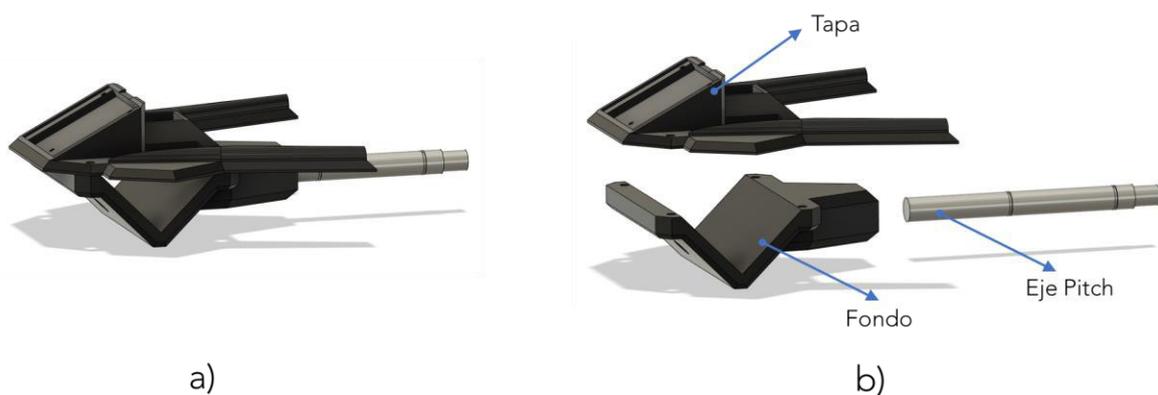
Los protectores están conformados por 3 piezas diferentes que son la parte inferior, la superior y las columnas que son pequeñas varillas de 63mm que unen ambas partes, tal

como se presenta en el Anexo B. Las partes inferior y superior se han elaborado con acrílico debido a que poseen un tamaño considerable (27cm de diámetro) y la mayoría de las impresoras 3D poseen una superficie de impresión limitada para estas dimensiones. Adicionalmente, dado que son piezas planas que no tienen formas complejas, se pueden obtener de una lámina de acrílico (que en este caso posee un grosor de 3mm) mediante corte láser.

La unión del soporte, protector y motores se denomina *conjunto motor - protector*. Esta denominación se utilizará en las siguientes secciones para identificar a este grupo de piezas.

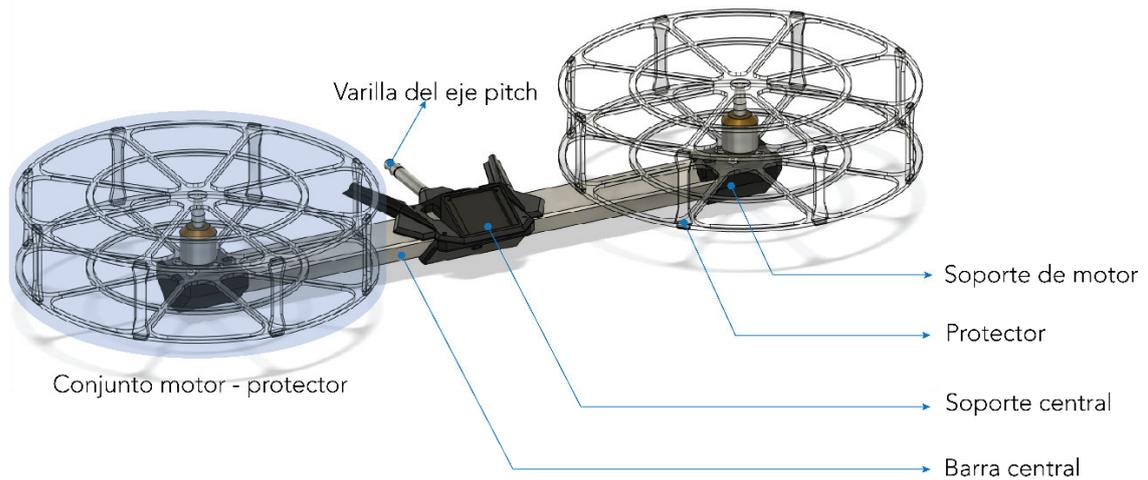
#### 2.2.2.1.4. Soporte central

El soporte central es la pieza que une la barra central con el eje principal de la plataforma y permite el movimiento alrededor del eje pitch. Esta pieza se compone de 3 partes que son la tapa, el fondo y el eje pitch. La tapa y el fondo son impresas en 3D, mientras que el eje es una varilla de acero plata de 10mm cuyas características (como ranuras y cejas) se realizan mediante trabajo en torno, los cuales le permiten acoplarse con el encoder de este eje. En la Figura 2.29 se muestra una vista general de esta pieza y las partes que la conforman. En el Anexo B se muestran diferentes vistas de esta pieza, y sus dimensiones, así como de cada una de las partes mencionadas.



**Figura 2.29.** Soporte central. a) Vista 3D general. b) Vista de despiece de las partes que lo conforman.

Al juntar todas las piezas descritas hasta ahora se conforma todo el cuerpo del helicóptero tal como se muestra en la Figura 2.30. La varilla del eje pitch es la estructura que une esta estructura con el eje principal y a su vez con el resto de la plataforma.

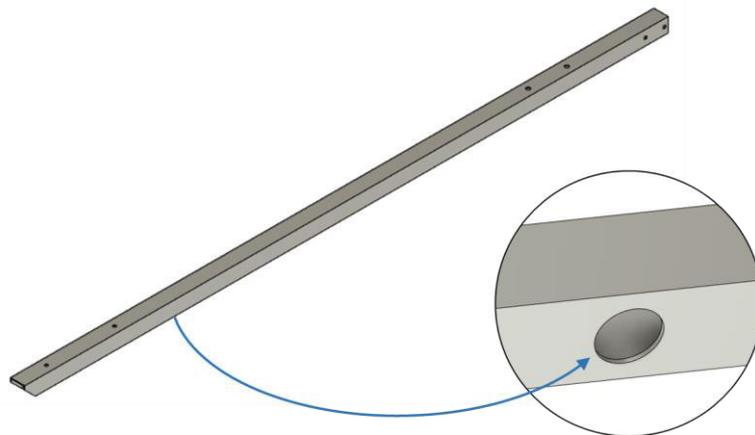


**Figura 2.30.** Cuerpo del helicóptero completo con sus partes señaladas.

## 2.2.2.2. Eje principal

### 2.2.2.2.1. Brazo de soporte

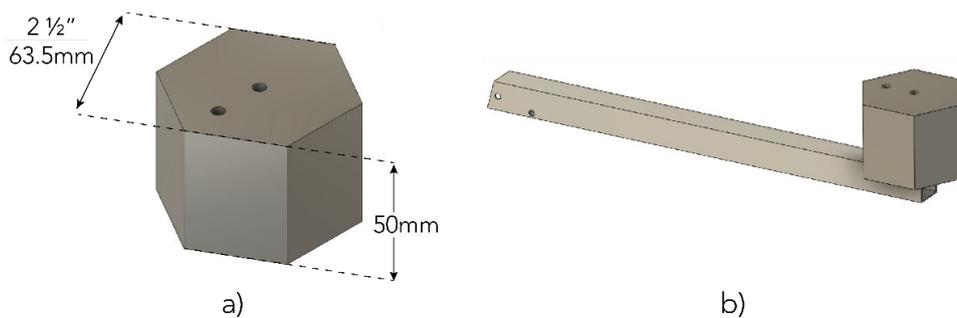
El brazo de soporte es la pieza de mayor longitud de la plataforma y se encarga de sostener el cuerpo del helicóptero y hacerlo rotar alrededor de los ejes travel y elevation. Consiste en un tubo cuadrado de aluminio de 19mm de ancho y sus paredes son de 1mm de espesor. Como se puede observar en la Figura 2.31, el brazo de soporte posee diferentes agujeros y ranuras mediante las cuales se acoplan las demás piezas. Además, en la parte inferior se encuentra un orificio de 17mm que sirve para pasar los cables desde la base hasta el cuerpo del helicóptero a través del interior del brazo.



**Figura 2.31.** Vista general en 3D del brazo de soporte y uno de los orificios para pasar cables en su parte inferior.

### 2.2.2.2.2. Contrapeso

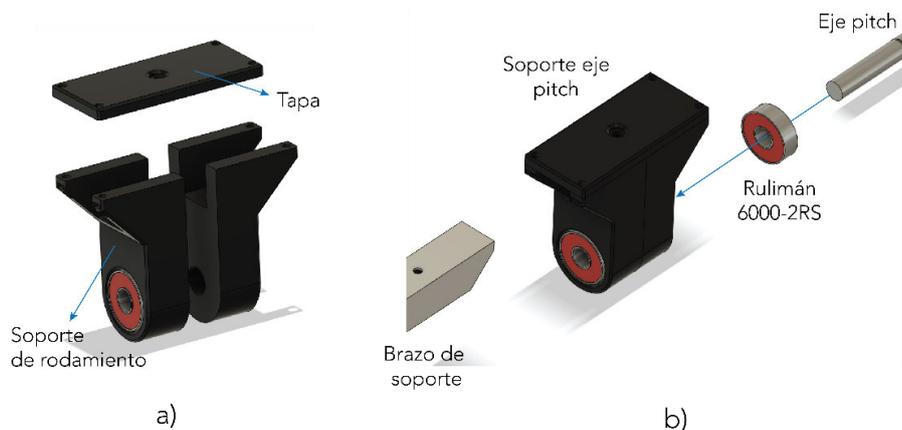
Para el contrapeso se ha utilizado una transmisión hexagonal de acero de tamaño estándar de 2 ½" y 5cm de altura, como se indica en la Figura 2.32. Con estas dimensiones se consigue una masa de aproximadamente 1400g. El contrapeso completo se conforma de la masa y un brazo de 31cm utilizando el mismo tubo de aluminio de 19mm de grosor que se utiliza en el brazo de soporte. Los agujeros que se observan en el contrapeso sirven para acoplarlo al brazo y al resto del eje principal mediante tornillos M5.



**Figura 2.32.** Contrapeso completo. a) Vista 3D y dimensiones del contrapeso. b) El contrapeso junto con el tubo para acoplar al brazo de soporte.

### 2.2.2.2.3. Soporte del eje pitch

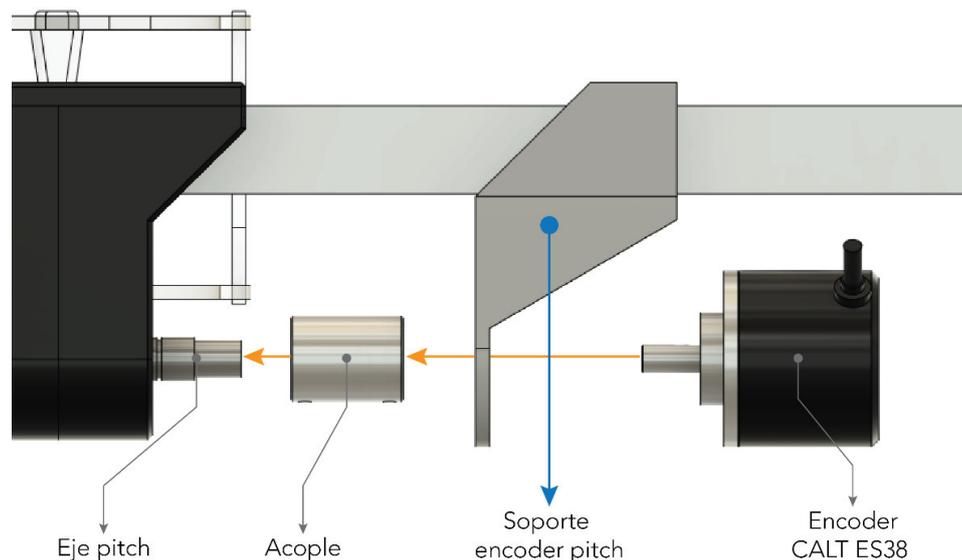
El soporte del eje pitch está conformado de 2 piezas que sostienen rodamientos de tipo 6000-2RS los cuales permiten el giro del cuerpo del helicóptero alrededor del eje pitch. Estas piezas se unen por medio de una tapa que a su vez se acopla con el brazo de soporte tal como se indica en la Figura 2.33 b). Las dimensiones y otras vistas de esta pieza se muestran a detalle en el Anexo B.



**Figura 2.33.** Soporte del eje pitch. a) Vista de despiece de la pieza y sus partes. b) Ubicación del brazo de soporte, rulimanes y varilla del eje pitch.

#### 2.2.2.2.4. Soporte para encoder del ángulo pitch

Esta pieza sirve para proporcionar un apoyo mecánico para el encoder de la variable pitch. Sin esta, el encoder de pitch estaría suspendido en el aire y el único sostén sería el proporcionado por el acople para el eje, lo cual daría lugar a mediciones poco robustas para esta posición angular. Adicionalmente, esta pieza permite mantener al sensor a una posición exacta del eje pitch de modo que el acople pueda calzar adecuadamente, tal como se muestra en la Figura 2.34.

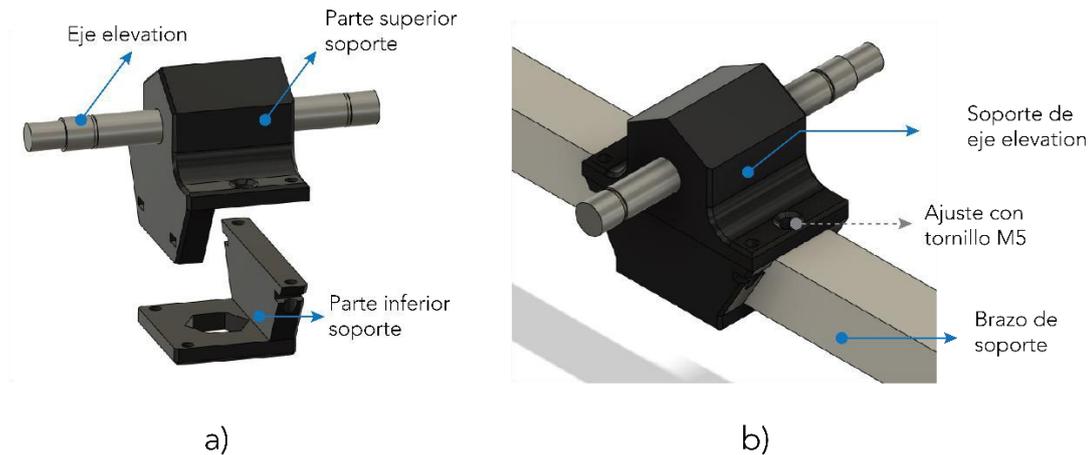


**Figura 2.34.** Ubicación del soporte para encoder del eje pitch con respecto a las demás partes. Revisar el anexo B para un detalle de las dimensiones y vistas de esta pieza.

La elaboración de esta pieza requiere conocer con exactitud las dimensiones del encoder CALT ES38 utilizado en este proyecto. Como se indica en los planos del Anexo B, los agujeros para los tornillos se ubican de forma correspondiente con las dimensiones del encoder.

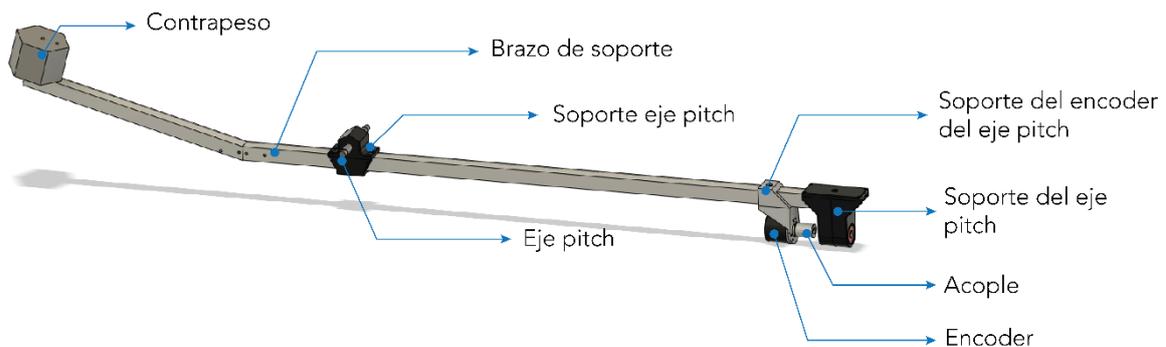
#### 2.2.2.2.5. Soporte del eje elevation

Esta es la pieza que se acopla en la parte central del brazo de soporte, se encarga de sostener todo el brazo de soporte y el contrapeso para apoyarse sobre la base. Consta de dos partes en forma de "L" para sostener todo el peso del soporte y el cuerpo del helicóptero y adicionalmente se acopla mediante tornillos M5 al brazo como se muestra en la Figura 2.35. Además, en esta pieza se acopla el eje elevation, que se elabora de la misma varilla de acero plata de 10mm que el eje pitch.



**Figura 2.35.** Soporte del eje elevation. a) Vista de despiece de la parte inferior y superior en forma de “L” b) Colocación en el brazo de soporte.

Tomando en cuenta todas las piezas de este grupo, se forma el segundo componente importante del helicóptero de 3-GDL que lleva el nombre de eje principal y se indica en la Figura 2.36. Tanto el eje principal como el cuerpo del helicóptero se colocan sobre la base, que es el tercer componente del prototipo y se describirá en los siguientes párrafos.

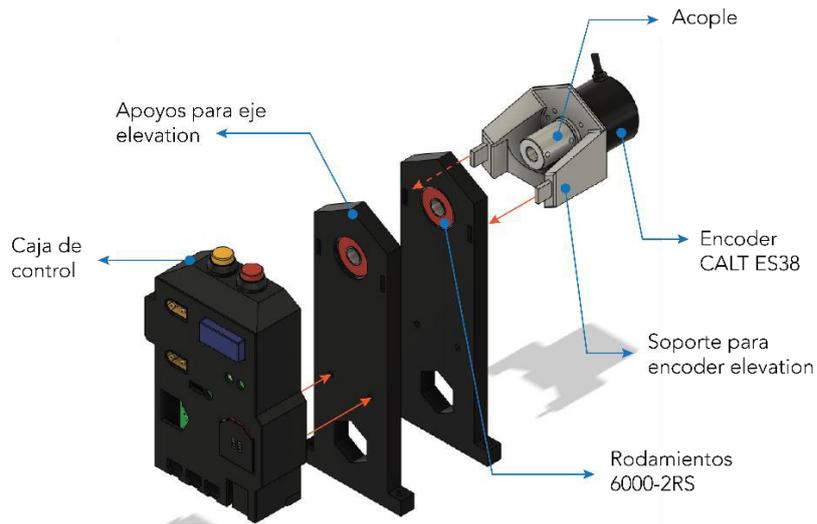


**Figura 2.36.** El eje principal completo con cada una de sus partes constituyentes.

### 2.2.2.3. Base

#### 2.2.2.3.1. Base elevation

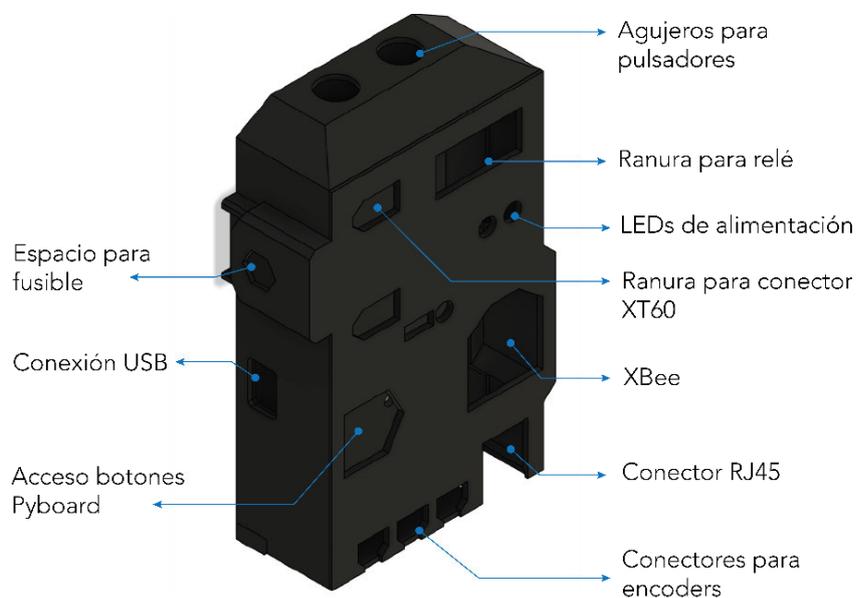
La estructura de base del eje elevation consta de dos piezas grandes que sirven de apoyo para todo el eje principal y el cuerpo del helicóptero. Debido al peso que deben resistir, estas piezas tienen un grosor de 10mm. Adicionalmente son las piezas que sirven para sostener la caja de control y el soporte para el encoder de elevation, tal como se indica en la Figura 2.37. Estas dos estructuras poseen un rodamiento de tipo 6000-2RS cada una, los cuales sostienen el eje elevation y permiten el giro en esta dirección.



**Figura 2.37.** Piezas de la base de elevation.

#### 2.2.2.3.2. Caja de control

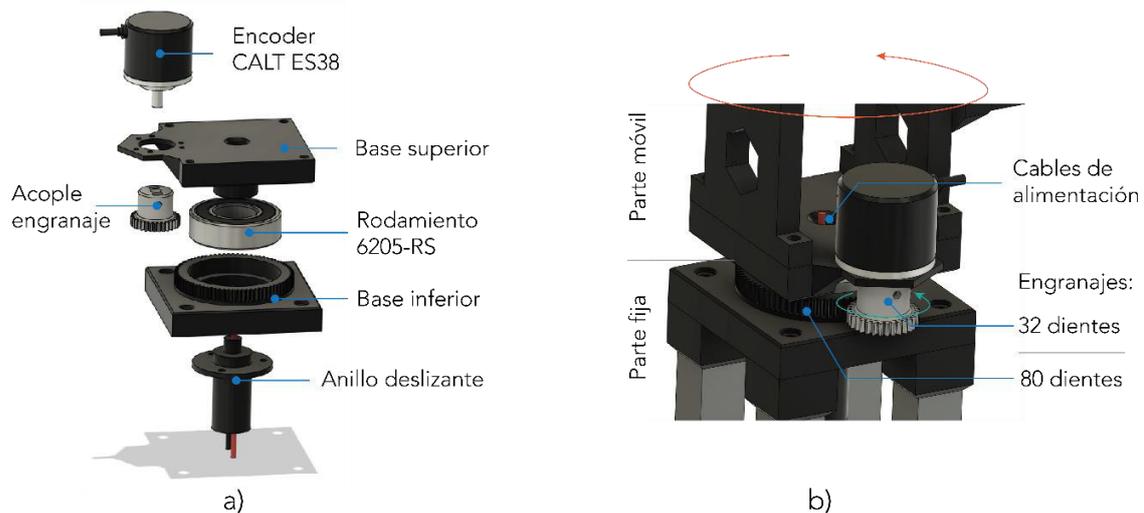
La caja de control es una estructura elaborada en impresión 3D que sirve como protección para la PCB principal y está constituida de 2 partes que son el fondo y la tapa. El modelo 3D fue desarrollado a partir de la PCB diseñada junto con los elementos que incluye, sus dimensiones y ubicaciones. Un diagrama explicativo de la caja se muestra en la Figura 2.38. Como se puede observar, la parte frontal posee numerosos orificios y ranuras que cumplen diferentes funciones, como por ejemplo permiten el espacio para algunos elementos y cables y también permiten visualizar los indicadores luminosos. La caja posee un agujero encima de la placa Pyboard, el cual permite acceder al botón de reset, lo cual es necesario para programar el microcontrolador y reiniciar el sistema.



**Figura 2.38.** Modelo 3D de la caja de control y descripción de las aberturas que posee.

### 2.2.2.3.3. Base travel

Los apoyos de la base de elevation, el eje principal y el cuerpo del helicóptero se asientan sobre la denominada base travel, la cual posee dos partes, una inferior y una superior que se acoplan mediante un rodamiento de tipo 6205-RS. Los planos de estas piezas se detallan en el Anexo B. Esta pieza también es la encargada de sostener el anillo deslizante que transfiere la energía desde la base hasta la parte móvil como se muestra en la Figura 2.39 a).



**Figura 2.39.** Base travel. a) Partes que constituyen. b) Se utiliza un engranaje para transmitir el movimiento sin interrumpir los cables del anillo deslizante.

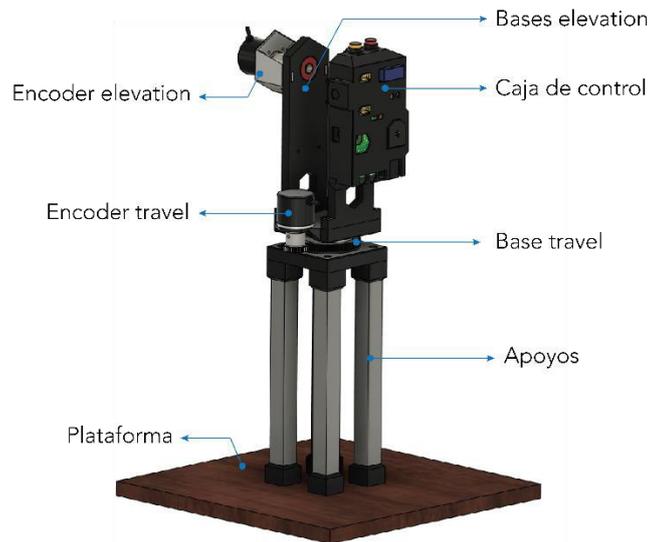
Dado que la base travel es la que permite el movimiento alrededor de este eje, se debe acoplar al encoder de forma que se pueda hacer la medición de la posición angular. Esto representa una dificultad ya que, como se mencionó antes, los cables del anillo deslizante deben pasar por el centro de la base. Para solucionar este inconveniente se ha utilizado un engranaje, mediante el cual se puede transmitir el movimiento hacia el encoder como se indica en la Figura 2.39 b).

El diseño de engranajes es una competencia de la ingeniería mecánica, sin embargo, el programa Fusion 360 incluye una gran cantidad de complementos, entre los cuales se encuentra la creación de engranajes rectos. Mediante esta herramienta se elaboró el engranaje de la base inferior (80 dientes) y el acople para el encoder (32 dientes), los cuales poseen una relación de 2:5. Es de suma importancia considerar que esta relación afecta la resolución de la medición ya que ahora se producirán 2.5 veces más pulsos por cada revolución del eje travel, por lo cual la ecuación corregida para la resolución en este eje es:

$$0.09 \frac{\circ}{\text{pulso}} \times \frac{2 \text{ revoluciones travel}}{5 \text{ revoluciones encoder}} = 0.036 \frac{\circ}{\text{pulso}} \quad (2.6)$$

#### 2.2.2.3.4. Apoyos y plataforma

Finalmente, toda la estructura se asienta sobre 4 apoyos elaborados del mismo tubo de aluminio de 20mm utilizado para el eje principal. Todos los apoyos se colocan sobre una plataforma de madera de 30x30cm. Al juntar todos los elementos mencionados para la base se obtiene la estructura indicada en la Figura 2.40.



**Figura 2.40.** Base completa con cada una de sus partes.

## 2.3. SISTEMA DE CONTROL

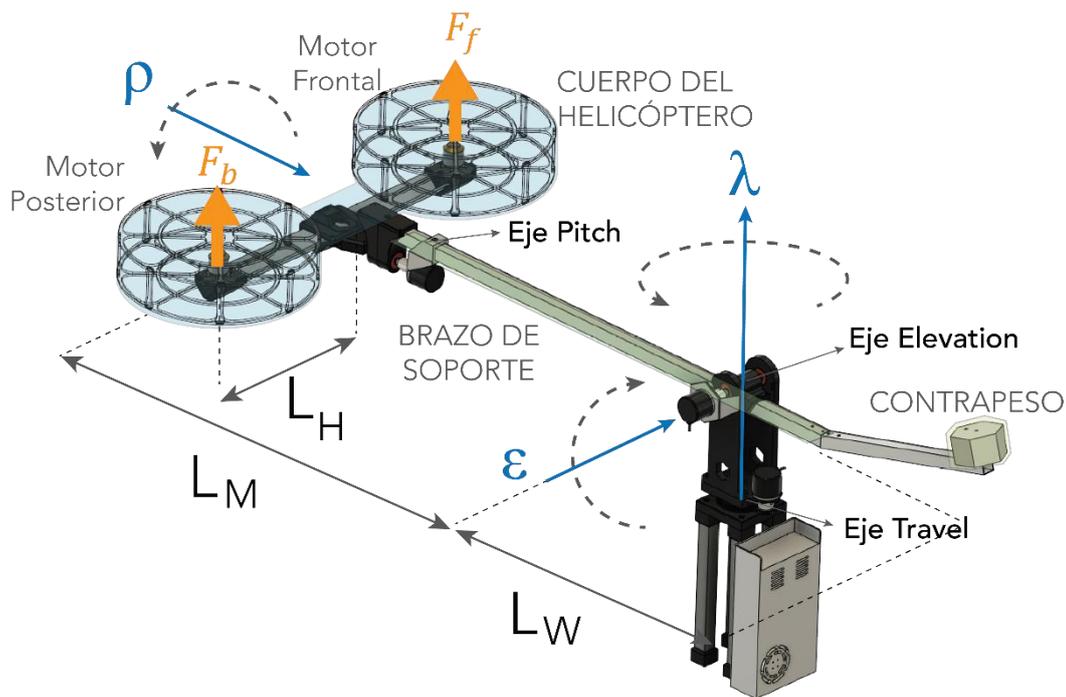
El helicóptero de 3-GDL es una plataforma experimental simplificada para validar la efectividad de los controladores de vuelo para UAVs multi-rotor. El control de esta plataforma representa una tarea de cierta complejidad debido a las características inherentes del sistema:

- Sistema sub-actuado: en esencia esto quiere decir que el número de actuadores (2) es menor que el número de grados de libertad (3).
- No lineal: las variables se relacionan de forma senoidal y cosenoidal entre ellas.
- Acoplamiento: como se ha mencionado en el anterior capítulo y se mostrará en los siguientes párrafos, los movimientos de travel y pitch se encuentran acoplados, lo que implica que no se pueden mover de forma independiente.

## 2.3.1. MODELO MATEMÁTICO DEL HELICÓPTERO DE 3-GDL

### 2.3.1.1. Variables del modelo

La dinámica del helicóptero de 3-GDL viene determinada por sus parámetros tales como longitudes, masas y las dos fuerzas de entrada  $F_b$  y  $F_f$ , que corresponden a las fuerzas de empuje generadas por las hélices posterior frontal (representada por la letra  $b$ ) y frontal (representada por la letra  $f$ ), respectivamente. Estas fuerzas son las encargadas de generar los movimientos alrededor de los tres ejes de rotación de la plataforma que se denominan travel  $\lambda$ , elevation  $\epsilon$  y pitch  $\rho$ . Estas son las salidas del sistema. La ubicación de las hélices y cada uno de los 3 ejes de libertad se especifican en la Figura 2.41.



**Figura 2.41.** Modelo 3D del helicóptero de 3-GDL y sus parámetros principales.  $F_b$  y  $F_f$  representan las fuerzas generadas por los motores posterior y frontal respectivamente

El ángulo de pitch es considerado  $0^\circ$  cuando el cuerpo del helicóptero se encuentra en posición horizontal. De manera similar, el eje elevation se encuentra en  $0^\circ$  cuando el brazo de soporte se encuentra completamente horizontal. Los rangos de movimiento de cada variable se muestran en la Tabla 2.7.

Los límites en el rango de pitch se dan por los topes mecánicos que han sido colocados para evitar vuelcos accidentales que puedan causar lesiones al usuario o daños de la plataforma. Cabe recalcar que dichos topes mecánicos permiten un rango de

aproximadamente  $[-50^\circ; 50^\circ]$ , el cual es suficiente para poder manipular el movimiento de la plataforma de forma efectiva. En el caso del ángulo de elevación,  $\epsilon$ , el límite inferior corresponde al ángulo al cual el helicóptero hace contacto con el piso.

**Tabla 2.7.** Variables del sistema y sus rangos de operación.

Variable	Símbolo	Unidad	Rango
Travel	$\lambda$	$^\circ$	$[-\infty; \infty]$
Elevation	$\epsilon$	$^\circ$	$[-30; 30]$
Pitch	$\rho$	$^\circ$	$[-50; 50]$
Fuerza del motor posterior	$F_b$	N	$[0; 5]$
Fuerza del motor frontal	$F_f$	N	$[0; 5]$

Como se indicó, el anillo deslizante permite transferir la energía a la plataforma incluso cuando se encuentra girando, lo cual elimina la necesidad de utilizar cables que se puedan enredar y permite la rotación libre alrededor del eje travel. Por esto, el límite de movimiento de esta variable está dado por los temporizadores/contadores internos del microcontrolador, los cuales, como se explicó en las subsecciones anteriores, pueden contar hasta más de mil millones. Considerando la relación de transformación del engranaje de acople del eje travel (2:5) y la resolución del encoder (1000PPR en cuadratura), este número corresponde a 50.000 vueltas en ambos sentidos. Por esta razón, en la Tabla 2.7 se muestra el rango de movimiento como  $[-\infty; \infty]$  para fines prácticos, y la posición de origen  $0^\circ$  se puede escoger de forma arbitraria.

Todo el movimiento de la plataforma está determinado por las fuerzas de los motores posterior y frontal. Como se puede observar en la Figura 2.41, cuando las dos fuerzas  $F_b$  y  $F_f$  son iguales entre sí y diferentes de cero, aparece una fuerza de empuje total, denominada  $F_{sum}$  en la dirección perpendicular al cuerpo del helicóptero (y al plano de giro de ambas hélices). El valor de esta variable está dado por la ecuación (2.7).

$$F_{sum} = F_f + F_b \quad (2.7)$$

Esta fuerza se encarga de generar una aceleración en la dirección de  $\epsilon$  (cuando  $\rho = 0$ ). Por otro lado, si las dos fuerzas son diferentes entre sí, la diferencia  $F_{dif}$ , dada por la ecuación (2.8), producirá un torque en la dirección del eje pitch, haciendo rotar al cuerpo del helicóptero en la dirección de  $\rho$ .

$$F_{dif} = F_f - F_b \quad (2.8)$$

Justamente, los motores frontal y posterior se nombran en función de la dirección en la cual se produce un ángulo pitch positivo satisfaciendo la condición  $F_f > F_b$ . Un ángulo pitch en sentido antihorario se considera positivo si se observa desde la posición del contrapeso. A partir de esta explicación se puede entender de mejor manera el modelo del helicóptero de 3-GDL en función de  $F_{sum}$  y  $F_{dif}$ , por lo tanto, estas son las variables que se considerarán como entradas al sistema para la derivación del modelo matemático.

Puesto que las variables que se modifican directamente son las fuerzas de cada motor, es decir,  $F_b$  y  $F_f$  resulta necesaria una relación entre estas fuerzas con  $F_{sum}$  y  $F_{dif}$  mediante la cual se puedan transformar entre sí. De las definiciones de ambas fuerzas, se puede observar que al sumar las Ecuaciones (2.7) y (2.8) se obtiene el siguiente resultado:

$$F_{sum} + F_{dif} = 2 F_f$$

De donde se encuentra que:

$$F_f = \frac{F_{sum} + F_{dif}}{2} \quad (2.9)$$

De manera similar, al restar la expresión (2.8) de (2.7) se encuentra que:

$$F_{sum} - F_{dif} = 2 F_b$$

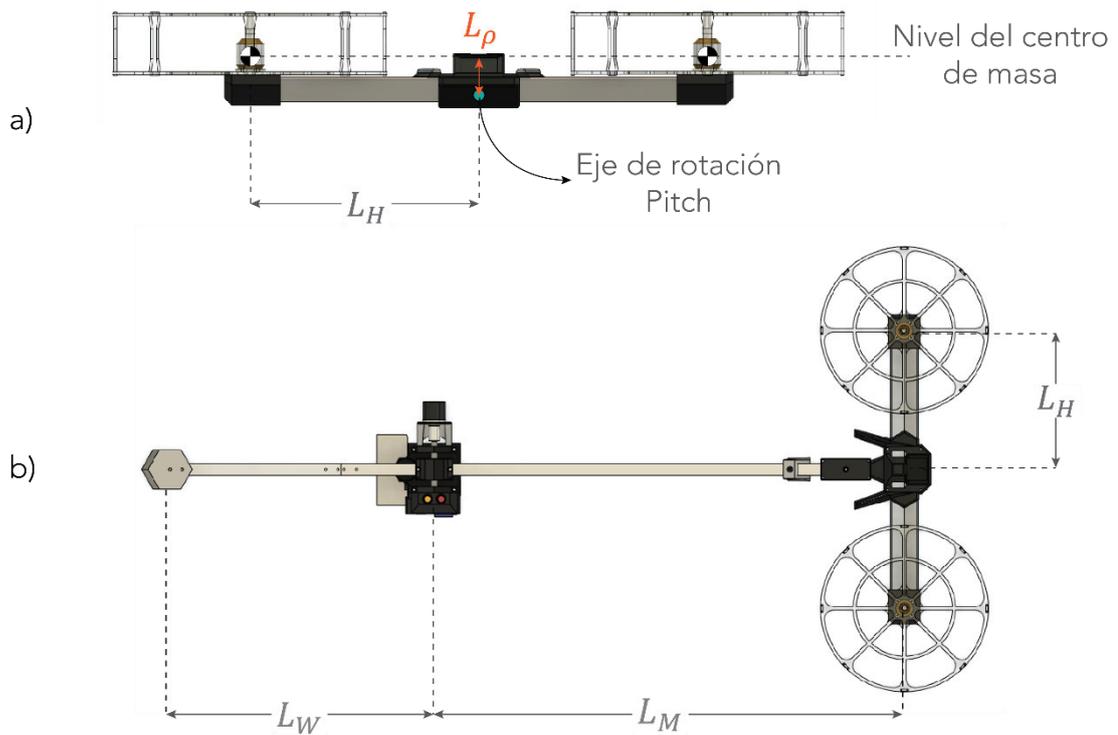
De donde:

$$F_b = \frac{F_{sum} - F_{dif}}{2} \quad (2.10)$$

Las expresiones (2.9) y (2.10) permiten expresar las acciones de control  $F_f$  y  $F_b$  en términos de  $F_{dif}$  y  $F_{sum}$ , lo cual resulta beneficioso debido a la relación directa que existe entre éstas y las salidas  $\rho$  y  $\epsilon$ . Por lo que, determinando los valores de  $F_{dif}$  y  $F_{sum}$  se pueden calcular los valores de  $F_b$  y  $F_f$  mediante las expresiones encontradas, las cuales son las acciones de control aplicadas directamente sobre cada motor.

### 2.3.1.2. Parámetros del sistema

Los parámetros que determinan toda la estructura y se relacionan con la dinámica del sistema son las longitudes, masas y momentos de inercia. Las longitudes principales son las que se muestran en la Figura 2.42.



**Figura 2.42.** Longitudes de la plataforma. a) Vista aislada del cuerpo del helicóptero. b) Vista superior de toda la plataforma.

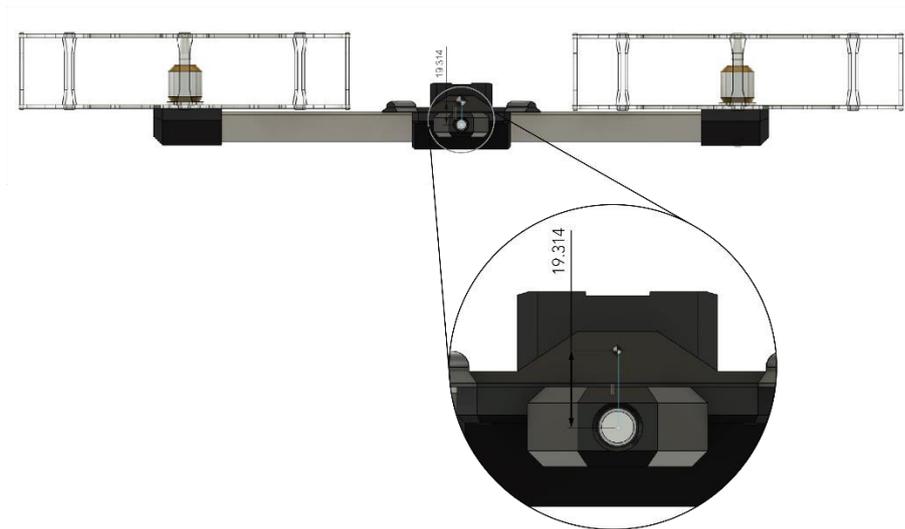
### 2.3.1.2.1. Longitudes

- $L_H$ : es la longitud desde el punto de rotación del eje pitch hasta el centro de masa del conjunto motor-protector a lo largo del cuerpo del helicóptero
- $L_M$ : es la longitud desde el eje de rotación de travel hasta el centro del cuerpo del helicóptero.
- $L_W$ : es la longitud desde el eje de rotación de travel hasta el contrapeso
- $L_\rho$ : longitud desde el eje de rotación de pitch hasta el nivel de los centros de masa de los conjuntos motor-protector tal como se muestra en la Figura 2.42 a).

Como se puede observar en la Figura 2.43, la longitud  $L_\rho$  no es una distancia física medible, sino que resulta de la determinación del centro de masa a partir de las distribuciones de masa de todos los componentes del sistema. Para determinar esta longitud se ha empleado la herramienta de centro de masa del programa Fusion 360 [50]. La exactitud de este resultado dependerá de la selección de materiales adecuados en la elaboración de los modelos 3D en el software. Para este caso, se ha considerado acrílico para los protectores,

aluminio para el perfil del cuerpo del helicóptero, plástico para los soportes y acero para los motores, lo cual corresponde de manera muy aproximada con el modelo real.

Al emplear la herramienta de análisis de centro de masa se ha encontrado que se sitúa a 19.314mm del centro del eje de rotación pitch. La determinación de esta longitud  $L_\rho$  es importante debido a que representa una característica fundamental de la dinámica de la variable pitch, lo cual, como se indica más adelante, se representa como un polo en el semiplano derecho en la función de transferencia obtenida en base al modelo de la plataforma.



**Figura 2.43.** El centro de masa del cuerpo del helicóptero se determina en Fusion 360. Para el cuerpo del helicóptero, el centro de masa se encuentra a  $L_\rho = 19.314\text{mm}$  del eje pitch.

#### 2.3.1.2.2. Masas

Las masas que se consideran en el helicóptero de 3-GDL son:

- $m_h$ : masa del conjunto motor-protector. Considerando ambos motores, esta masa representa casi el 50% del total de la masa del cuerpo del helicóptero. Se asume que ambos conjuntos motor-protector tienen el mismo peso.
- $m_H$ : masa total del cuerpo del helicóptero. Esta masa está principalmente conformada por los dos conjuntos motor-protector, el perfil de aluminio de soporte para el cuerpo, y el soporte central. No se debe confundir con  $m_h$  (h minúscula) que es solamente la masa del conjunto motor-protector.
- $m_W$ : masa del contrapeso. El contrapeso sirve para reducir el esfuerzo generado por los motores y, por lo tanto, para modificar la dinámica del sistema.

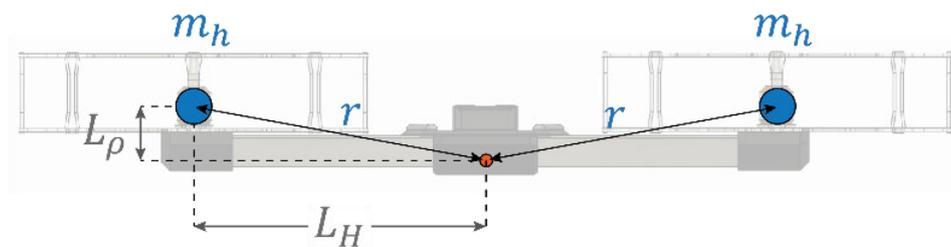
### 2.3.1.2.3. Momentos de inercia

Para simplificar la consideración de los momentos de inercia se considerarán los componentes del sistema como masas puntuales. Esto permite reducir la complejidad de las ecuaciones sin omitir las características dinámicas más importantes del sistema. Considerando esto, cabe recordar que la determinación del momento de inercia para un sistema de masas puntuales se realiza mediante la ecuación (2.11).

$$I_x = \sum_i m_i r_i^2 \quad (2.11)$$

Donde  $m_i$  son todas las masas que conforma el sistema y  $r_i$  son las distancias perpendiculares desde la ubicación de la masa  $m_i$  hasta el eje de rotación  $x$ .

- $I_\rho$ : momento de inercia alrededor del eje de rotación pitch. Para el cálculo se considera que los conjuntos motor-protector son masas puntuales  $m_h$  que se encuentran a una distancia vertical  $L_\rho$  y horizontal  $L_H$  del centro de rotación del eje pitch, tal como se muestra en la Figura 2.44.

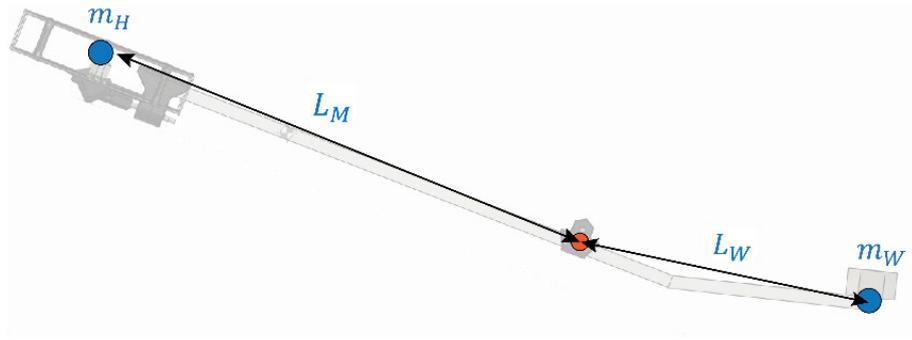


**Figura 2.44.** Distancia de las masas  $m_h$  con respecto al eje de rotación  $\rho$ .

Considerando los parámetros del esquema de la Figura 2.44 y la ecuación (2.11), el momento de inercia en este eje es:

$$I_\rho = 2m_h(L_H^2 + L_\rho^2) \quad (2.12)$$

- $I_\epsilon$ : momento de inercia alrededor del eje de rotación elevation. Para este caso se considera que toda la distribución de masa del cuerpo del helicóptero se reduce en una sola masa puntual  $m_H$  y el contrapeso es otra masa puntual  $m_W$ . Tal como se indica en la Figura 2.45.

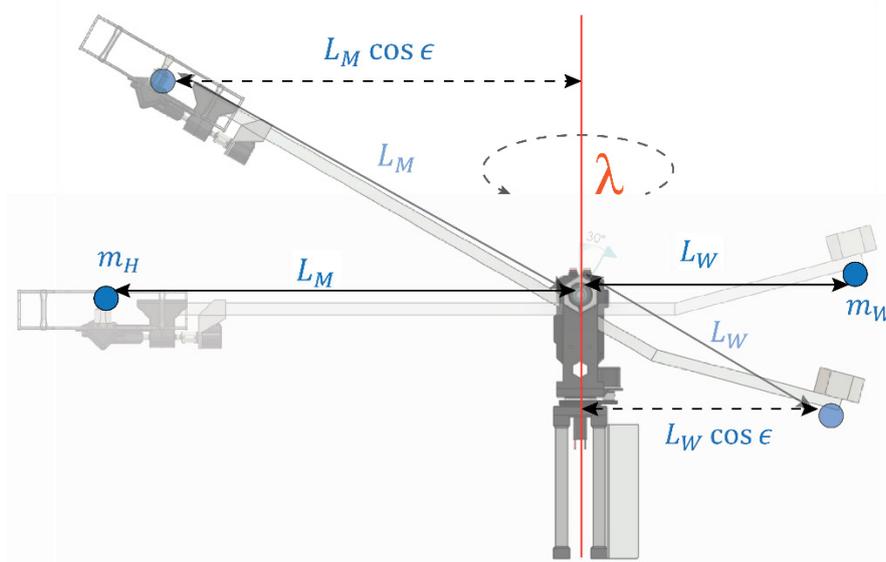


**Figura 2.45.** Distancia de las masas  $m_H$  y  $m_W$  respecto al eje de rotación  $\epsilon$ .

Entonces, el momento de inercia alrededor del eje  $\epsilon$  está dado por:

$$I_{\epsilon} = m_H L_M^2 + m_W L_W^2 \quad (2.13)$$

- $I_{\lambda}$ : momento de inercia alrededor del eje de rotación travel. Al igual que en el caso anterior se consideran únicamente las masas puntuales  $m_H$  y  $m_W$  sin embargo, la posición en elevation genera un decaimiento de la distancia con respecto al eje de rotación ya que es proporcional al  $\cos(\epsilon)$ , tal como se indica en la Figura 2.46.



**Figura 2.46.** La distancia de las masas  $m_H$  y  $m_W$  hacia el eje de rotación de  $\lambda$  varía de manera proporcional al  $\cos(\epsilon)$ .

Tomando esto en cuenta, el momento de inercia alrededor del eje travel se expresa mediante la Ecuación (2.14):

$$I_{\lambda}(\epsilon) = m_H(L_H \cos(\epsilon))^2 + m_W(L_W \cos(\epsilon))^2$$

$$I_{\lambda}(\epsilon) = (m_H L_M^2 + m_W L_W^2) \cos^2(\epsilon) \quad (2.14)$$

Todos los parámetros se han medido utilizando una balanza para las masas o un flexómetro para las longitudes. Los momentos de inercia se han determinado para el punto de operación mediante las ecuaciones mostradas anteriormente y el resumen de todos los parámetros del helicóptero de 3-GDL se muestra en la Tabla 2.8.

**Tabla 2.8.** Parámetros del helicóptero de 3-GDL elaborado.

Parámetro	Valor	Unidad	Descripción
$I_{\lambda}$	0.702	kg m <sup>2</sup>	Momento de inercia alrededor del eje travel en el punto de operación.
$I_{\epsilon}$	0.702	kg m <sup>2</sup>	Momento de inercia alrededor del eje elevation en el punto de operación.
$I_{\rho}$	0.0201	kg m <sup>2</sup>	Momento de inercia alrededor del eje pitch en el punto de operación.
$m_h$	0.189	kg	Masa del conjunto motor-protector.
$m_H$	0.899	kg	Masa total del cuerpo del helicóptero.
$m_W$	1.477	kg	Masa del contrapeso.
$L_{\rho}$	0.0193	m	Longitud desde el eje pitch hasta el centro de masa del cuerpo del helicóptero.
$L_H$	0.23	m	Longitud desde el motor hasta el eje pitch.
$L_M$	0.72	m	Longitud desde el cuerpo del helicóptero hasta el eje travel.
$L_W$	0.4	m	Longitud desde el contrapeso hasta el eje travel.
$g$	9.81	m/s <sup>2</sup>	Constante de aceleración de la gravedad.

### 2.3.1.3. Modelo matemático

Para la derivación de las ecuaciones matemáticas del sistema se ha utilizado como base el procedimiento descrito en [51]. La obtención de un modelo detallado del helicóptero de 3-GDL representa una dificultad considerable debido a su naturaleza no lineal y variante en el tiempo. Sin embargo, un modelo de complejidad reducida que conserve el comportamiento dinámico más relevante del sistema puede ser derivado utilizando las ecuaciones de Euler-Lagrange con relativa facilidad.

### 2.3.1.3.1. Ecuación de Euler-Lagrange.

Las ecuaciones de Lagrange son una formulación de la mecánica equivalente a las ecuaciones de Newton. Esta representación ofrece una gran cantidad de ventajas como por ejemplo que tiene la misma forma independientemente del sistema de coordenadas utilizado. Además, utiliza cantidades escalares como energías cinéticas y potenciales del sistema en lugar de cantidades vectoriales como en el caso de las ecuaciones de Newton [52].

Para hacer uso de esta representación se emplea la *función Lagrangiana*,  $\mathcal{L}$ , (también llamada simplemente *Lagrangiano*) que recibe su nombre por su formulador Joseph-Louis Lagrange. Este es un operador escalar que resulta de la diferencia entre la energía cinética  $K$  y la potencial  $U$  como se muestra en la Ecuación (2.15)

$$\mathcal{L} = K - U \quad (2.15)$$

Con base en esta definición se establece la relación mostrada en la ecuación (2.16) conocida como ecuación de Euler-Lagrange:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}_j} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}_j} = 0 \quad (2.16)$$

En esta ecuación  $\mathbf{q}_j$  se refiere a las coordenadas generalizadas del sistema (que para el modelo en cuestión son los ángulos de libertad:  $\rho, \epsilon, \lambda$ ) y  $\dot{\mathbf{q}}_j$  representa a todas sus derivadas temporales (las velocidades  $\dot{\rho}, \dot{\epsilon}, \dot{\lambda}$ ). Se puede añadir un término adicional en la Ecuación (2.16) para representar la excitación por una fuerza externa  $\mathbf{Q}_j$  de la siguiente forma:

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}_j} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}_j} = \mathbf{Q}_j \quad (2.17)$$

Para el caso del helicóptero de 3-GDL, la fuerza  $\mathbf{Q}_j$  mostrada en la Ecuación (2.17) corresponde a los torques generados por  $F_{sum}$  y  $F_{dif}$  sobre cada uno de los ejes de rotación de la plataforma.

Considerando todo lo dicho, para entender de mejor manera el movimiento de la plataforma se puede separar en 3 partes diferentes que esencialmente corresponden a cada uno de los grados de libertad del helicóptero. Esto se hace posible gracias a que, como se mencionó en los párrafos anteriores, se pueden relacionar las entradas  $F_{dif}$  y  $F_{sum}$  con las

salidas  $\rho$  y  $\epsilon$  respectivamente, lo cual permite diferenciar 3 “submodelos” distintos que se mencionan a continuación.

### 2.3.1.3.2. Movimiento del eje pitch $\rho$ .

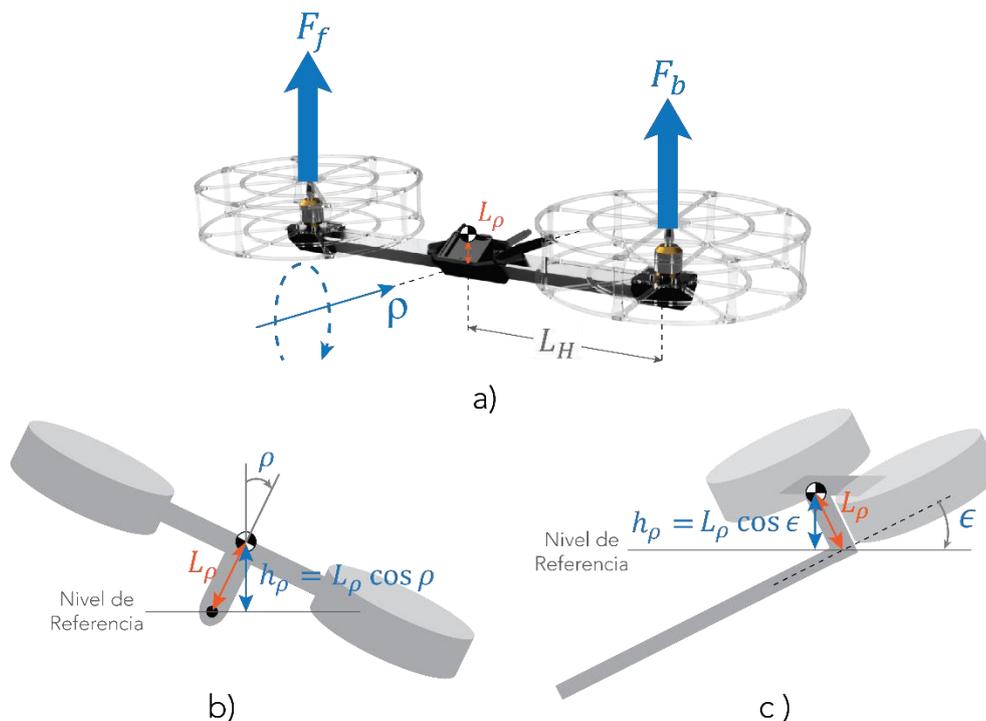
Para el modelo de la dinámica del ángulo pitch, se calcula el Lagrangiano, para lo cual se determinan las energías del sistema, partiendo por la cinética  $K$  de la siguiente manera:

$$K = \frac{1}{2} I_\rho \dot{\rho}^2 \quad (2.18)$$

Donde  $I_\rho$  es el momento de inercia alrededor del eje pitch y  $\dot{\rho}$  es la velocidad angular. La energía potencial  $U$  toma la siguiente forma:

$$U = m_H g h_\rho \quad (2.19)$$

Donde  $m_H$  es la masa del helicóptero,  $g$  es el valor de la aceleración de la gravedad y  $h_\rho$  es la altura desde el centro de masa del cuerpo del helicóptero hasta el nivel de referencia, el cual se encuentra siempre horizontal en el punto de rotación del eje pitch tal como se muestra en la Figura 2.47.



**Figura 2.47.** Esquema del cuerpo del helicóptero y las variables para el eje pitch. a) Vista general del cuerpo del helicóptero. b) Vista frontal: la altura del centro de masa depende del ángulo pitch. c) Vista lateral: la altura también depende de elevation.

Como se puede observar en esta imagen, la altura  $h_\rho$  es función de los valores de los ángulos  $\epsilon$  y  $\rho$  relacionados de acuerdo con lo que se muestra en la ecuación (2.20)

$$h_\rho = L_\rho \cos(\epsilon) \cos(\rho) \quad (2.20)$$

Por lo que,  $h_\rho$  simplemente es la posición del centro de masa con respecto al eje pitch (nivel de referencia), y la razón por la cual depende del valor de  $\epsilon$  es porque el nivel de referencia se mantiene siempre horizontal. Por ejemplo, si tomara un valor de  $\epsilon = 90^\circ$  la altura  $h_\rho$  estuviera exactamente en el nivel de referencia, por lo que no existiría energía potencial y no habría ninguna aceleración sobre el eje pitch causada por este parámetro.

Tomando en cuenta las Ecuaciones (2.18), (2.19) y (2.20), el Lagrangiano para el eje pitch estaría definido por:

$$\mathcal{L} = \frac{1}{2} I_\rho \dot{\rho}^2 - m_H g L_\rho \cos(\epsilon) \cos(\rho) \quad (2.21)$$

Obteniendo las derivadas:

$$\frac{d\mathcal{L}}{d\rho} = m_H g L_\rho \cos(\epsilon) \sin(\rho) \quad (2.22)$$

$$\frac{d\mathcal{L}}{d\dot{\rho}} = I_\rho \dot{\rho} \rightarrow \frac{d}{dt} \left( \frac{d\mathcal{L}}{d\dot{\rho}} \right) = I_\rho \ddot{\rho} \quad (2.23)$$

Por último, la fuerza general  $Q$  para el movimiento de este eje está dado por la diferencia de los torques generados por los empujes de los motores sobre este eje. Asumiendo que los empujes de los motores son completamente perpendiculares al cuerpo del helicóptero, los torques resultan del producto entre el brazo de palanca  $L_H$  y la fuerza de cada motor, tal como se indica en la ecuación (2.24):

$$\begin{aligned} Q_\rho &= L_H F_f - L_H F_b \\ &= L_H (F_f - F_b) = L_H F_{dif} \end{aligned} \quad (2.24)$$

Reemplazando las expresiones (2.22), (2.23), y (2.24) en la ecuación de Euler-Lagrange, se obtiene el modelo para el eje pitch mostrado en la ecuación (2.25), en el cual se tiene una única entrada  $F_{dif}$  y una única salida  $\rho$ . No obstante, se puede observar que el valor del ángulo  $\epsilon$  afecta el comportamiento de esta variable, sin embargo, es un factor que multiplica a la constante  $L_\rho$ , la cual tiene un valor muy pequeño, por lo que el efecto que el ángulo  $\epsilon$  produce sobre la dinámica del eje pitch es prácticamente despreciable.

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\rho}} \right) - \frac{\partial \mathcal{L}}{\partial \rho} = Q_{\rho}$$

$$I_{\rho} \ddot{\rho} - m_H g L_{\rho} \cos(\epsilon) \sin(\rho) = L_H F_{dif}$$

$$I_{\rho} \ddot{\rho} = L_H F_{dif} + m_H g L_{\rho} \cos(\epsilon) \sin(\rho) \quad (2.25)$$

### 2.3.1.3.3. Movimiento del eje elevation $\epsilon$

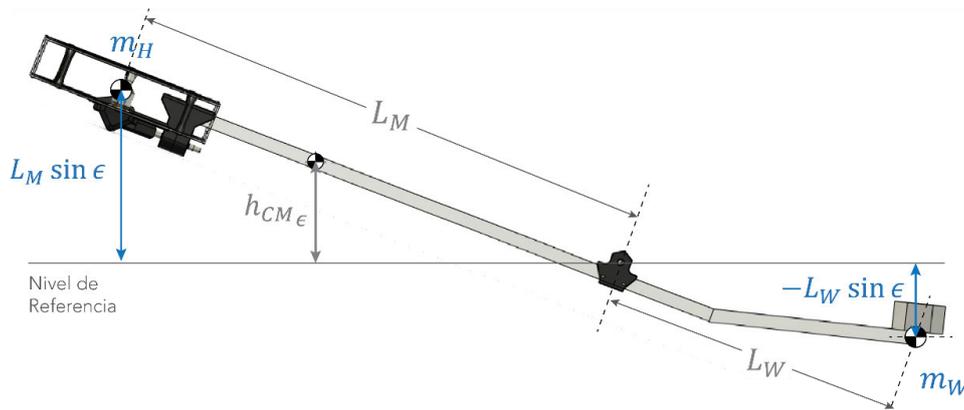
Al igual que en el caso anterior se empieza por el cálculo del Lagrangiano, para lo cual se determinan las energías:

$$K = \frac{1}{2} I_{\epsilon} \dot{\epsilon}^2 \quad (2.26)$$

La energía potencial  $U$  viene dada por:

$$U = (m_H + m_W) g h_{CM\epsilon} \quad (2.27)$$

Donde  $h_{CM\epsilon}$  es la altura del centro de masa del sistema con respecto al nivel de referencia fijado sobre el eje de elevation. Para encontrar la ubicación del centro de masa se considerará que el movimiento sobre este eje está dado únicamente por dos masas puntuales que corresponden al cuerpo del helicóptero  $m_H$  y el contrapeso  $m_W$  tal como se muestra en la Figura 2.48.



**Figura 2.48.** Ubicación de las masas del cuerpo del helicóptero, el contrapeso y el centro de masa del sistema sobre el nivel de referencia del eje elevation. Como se puede observar, el nivel de  $m_W$  es negativo ya que se encuentra al lado contrario de  $h_{CM\epsilon}$ .

Dicho esto, el centro de masa (sobre el eje vertical, ya que es una altura) se encuentra en la posición dada por la ecuación (2.28):

$$h_{CM\epsilon} = \frac{m_H L_M \sin(\epsilon) - m_W L_W \sin(\epsilon)}{m_H + m_W} = \frac{m_H L_M - m_W L_W}{m_H + m_W} \sin(\epsilon) \quad (2.28)$$

De modo que la energía potencial en este caso es:  $U = (m_H + m_W)g \frac{m_H L_M - m_W L_W}{m_H + m_W} \sin(\epsilon)$  (2.29)

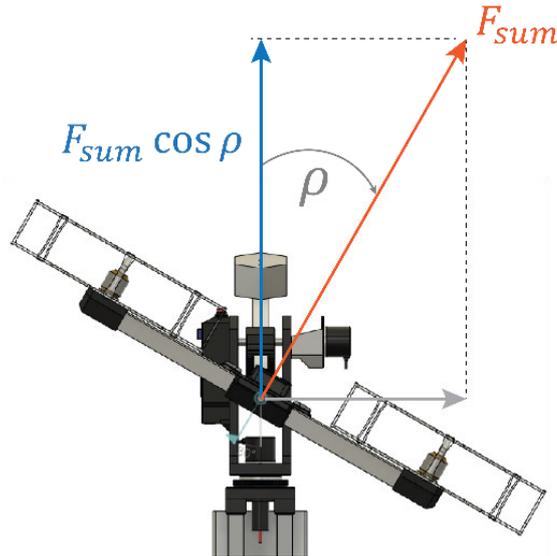
Tomando en cuenta las Ecuaciones (2.26) y (2.29), el Lagrangiano para el eje elevation queda definido por:

$$\mathcal{L} = \frac{1}{2} I_\epsilon \dot{\epsilon}^2 - (m_H L_M - m_W L_W)g \sin(\epsilon) \quad (2.30)$$

Obteniendo las derivadas:

$$\frac{d\mathcal{L}}{d\epsilon} = (m_W L_W - m_H L_M)g \cos(\epsilon) \quad (2.31)$$

$$\frac{d\mathcal{L}}{d\dot{\epsilon}} = I_\epsilon \dot{\epsilon} \rightarrow \frac{d}{dt} \left( \frac{d\mathcal{L}}{d\dot{\epsilon}} \right) = I_\epsilon \ddot{\epsilon} \quad (2.32)$$



**Figura 2.49.** Vista frontal del helicóptero de 3-GDL. Como se puede observar, la componente del empuje total que genera torque sobre el eje de elevation es proporcional al coseno del ángulo pitch.

En este caso, el torque viene dado por la componente de la suma de los empujes de los motores en la dirección perpendicular al eje elevation. Es decir, depende del valor del ángulo pitch ya que si este se encuentra a 90°, el empuje a favor del ángulo elevation será

nulo, tal como se puede observar en la Figura 2.49. Por esta razón, el torque  $Q_\epsilon$  sobre el eje de elevation viene dado por la expresión (2.33):

$$\begin{aligned} Q_\epsilon &= L_M(F_f + F_b) \cos(\rho) \\ &= L_M F_{sum} \cos(\rho) \end{aligned} \quad (2.33)$$

De manera similar al ejemplo anterior, reemplazando las expresiones (2.31), (2.32), y (2.33) en la ecuación de Euler-Lagrange, se obtiene el modelo para el eje elevation, que se muestra en la ecuación (2.34). Como se puede observar en esta ecuación, la entrada es  $F_{sum}$  y la salida es el valor del ángulo  $\epsilon$ , tomando en cuenta que el ángulo  $\rho$  toma valores muy pequeños en los cuales  $\cos(\rho) \approx 1$ .

$$\begin{aligned} \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\epsilon}} \right) - \frac{\partial \mathcal{L}}{\partial \epsilon} &= Q_\epsilon \\ I_\epsilon \ddot{\epsilon} - (m_W L_W - m_H L_M) g \cos(\epsilon) &= L_M F_{sum} \cos(\rho) \\ I_\epsilon \ddot{\epsilon} &= L_M F_{sum} \cos(\rho) - (m_H L_M - m_W L_W) g \cos(\epsilon) \end{aligned} \quad (2.34)$$

#### 2.3.1.3.4. Movimiento del eje travel $\lambda$

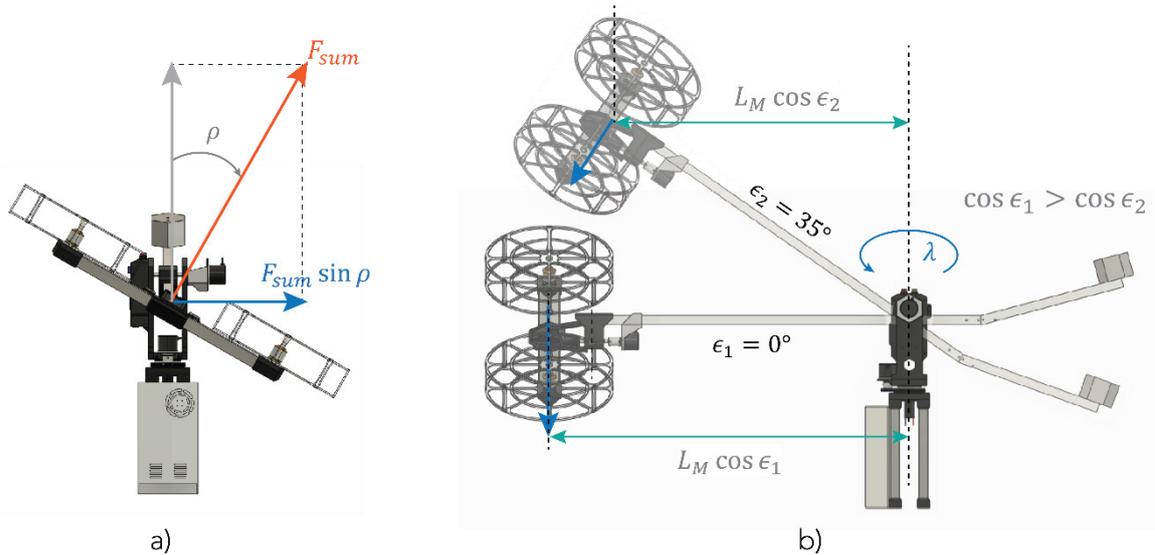
Finalmente, para la obtención del modelo sobre el eje travel se sigue un procedimiento similar. No obstante, cabe mencionar que sobre este eje no existe la influencia de la gravedad (ya que se encuentra sobre un plano perpendicular) por lo cual la energía potencial  $U$  es nula. Por esta razón, el Lagrangiano en el eje travel es igual a la energía cinética  $K$ , de modo que se encuentra la expresión de la ecuación (2.35). Cabe recalcar que, como se mencionó en los párrafos anteriores, el momento de inercia con respecto a este eje de giro es dependiente del ángulo  $\epsilon$  por lo cual se muestra como una función  $I_\lambda(\epsilon)$ , véase la ecuación (2.14).

$$\mathcal{L} = K = \frac{1}{2} I_\lambda(\epsilon) \dot{\lambda}^2 \quad (2.35)$$

Para este caso solo se tiene la derivada con respecto a  $\lambda$ . A su vez, la derivada temporal está dada por:

$$\frac{d\mathcal{L}}{d\dot{\lambda}} = I_\lambda(\epsilon) \dot{\lambda} \rightarrow \frac{d}{dt} \left( \frac{d\mathcal{L}}{d\dot{\lambda}} \right) = I_\lambda(\epsilon) \ddot{\lambda} \quad (2.36)$$

De manera análoga al caso anterior, el torque sobre esta dirección viene dado únicamente por la componente horizontal del empuje neto  $F_{sum}$ , es decir, es proporcional al seno del ángulo pitch, tal como se indica en la Figura 2.50 a).



**Figura 2.50.** Parámetros del movimiento en el eje travel. a) Vista frontal: la componente que genera movimiento en este eje es proporcional al seno del ángulo pitch. b) Vista lateral para 2 valores de elevation. El brazo de palanca es proporcional al  $\cos(\epsilon)$ .

Adicionalmente, es importante considerar que la fuerza de empuje en esta dirección va decayendo de forma proporcional al coseno del ángulo  $\epsilon$ . Esto puede ser comprobado observando que cuando  $\epsilon = 90^\circ$  el torque sobre el eje travel será nulo, mientras que en la posición horizontal  $\epsilon = 0^\circ$  tomará el máximo valor. Esto se debe a que el torque es proporcional al brazo de palanca, medido con respecto al eje de rotación. Como se muestra en la Figura 2.50 b), conforme  $\epsilon$  aumenta, el brazo de palanca disminuye ya que la proyección sobre el plano de giro es  $L_M \cos(\epsilon)$ .

Considerando este análisis, el torque que ingresa al sistema de travel  $Q_\lambda$  está dado por:

$$Q_\lambda = L_M F_{sum} \sin(\rho) \cos(\epsilon) \quad (2.37)$$

Reemplazando las dos expresiones anteriores en la ecuación de Euler-Lagrange se obtiene el modelo matemático para el eje travel como se muestra en la ecuación (2.38). En este caso se puede observar que la aceleración de  $\lambda$  depende de  $\rho$ ,  $\epsilon$  y  $F_{sum}$ . Sin embargo, cabe recalcar que para mantener un valor de  $\epsilon = 0^\circ$  se necesita una fuerza  $F_{sum}$  constante, con lo cual se considera que la entrada de este sistema es el ángulo  $\rho$  y la salida es el valor

de  $\lambda$ . Estas consideraciones se utilizarán más adelante en la obtención del modelo reducido en función de transferencia.

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\epsilon}} \right) = Q_{\epsilon}$$

$$I_{\lambda}(\epsilon) \ddot{\lambda} = L_M \sin(\rho) \cos(\epsilon) F_{sum} \quad (2.38)$$

### 2.3.1.3.5. Modelo completo y linealización

El sistema total está compuesto de 3 salidas que son los ángulos de libertad de la plataforma y las dos entradas  $F_{dif}$  y  $F_{sum}$  que pueden ser convertidas a las fuerzas  $F_f$  y  $F_b$  para su implementación en el sistema de control.

Las ecuaciones que describen el movimiento de la plataforma son:

$$I_{\rho} \ddot{\rho} = L_H F_{dif} + m_H g L_{\rho} \cos(\epsilon) \sin(\rho) \quad (2.39)$$

$$I_{\epsilon} \ddot{\epsilon} = L_M F_{sum} \cos(\rho) - (m_H L_M - m_W L_W) g \cos(\epsilon) \quad (2.40)$$

$$I_{\lambda}(\epsilon) \ddot{\lambda} = L_M \sin(\rho) \cos(\epsilon) F_{sum} \quad (2.41)$$

Como se ha mostrado en los párrafos anteriores, para la obtención de las ecuaciones de este modelo no se ha considerado el efecto de rozamientos ni fuerzas Coriolis ya que su efecto puede ser despreciado [53].

A partir de estas ecuaciones se puede observar que el modelo es no lineal y, además, como se mencionó anteriormente, el momento de inercia sobre el eje travel depende en gran medida de las otras variables. Sin embargo, se puede reducir la complejidad del sistema linealizando alrededor de una condición de operación establecida por:

$$\begin{pmatrix} \rho \\ \epsilon \end{pmatrix} \approx \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2.42)$$

Asumiendo que el sistema opera en las cercanías de esta condición, se considera que los senos y cosenos se aproximan a valores lineales de la siguiente manera:

$$\begin{aligned} \sin(\rho) &\approx \rho, & \sin(\epsilon) &\approx \epsilon \\ \cos(\rho) &\approx 1, & \cos(\epsilon) &\approx 1 \end{aligned} \quad (2.43)$$

Con las aproximaciones consideradas, las ecuaciones del sistema se reducen a:

$$I_{\rho} \ddot{\rho} = L_H F_{dif} + m_H g L_{\rho} \rho \quad (2.44)$$

$$I_\epsilon \ddot{\epsilon} = L_M F_{sum} - K g_\epsilon \quad (2.45)$$

$$I_\lambda \ddot{\lambda} = L_M \rho F_{sum} \quad (2.46)$$

Donde  $K g_\epsilon = (m_H L_M - m_W L_W) g$  es un término que expresa el efecto gravitacional sobre la aceleración del eje elevation. En otras palabras, lo que ese término indica es que, si no se aplica ningún empuje sobre el helicóptero, el valor de elevation se acelera “hacia abajo”, es decir, se cae, por eso lleva el signo negativo. Por otra parte, en cuanto al modelo de travel, como  $\cos(\epsilon) \approx 1$  el momento de inercia alrededor de este eje se considera constante  $I_\lambda$ . A partir de esta formulación linealizada de complejidad reducida se puede obtener un modelo en funciones de transferencia. Obteniendo la transformada de Laplace del modelo linealizado dado en la ecuación (2.44) se encuentra:

$$I_\rho P(s) s^2 = L_H F_{dif}(s) + m_H g L_{rho} P(s)$$

$$P(s) [I_\rho s^2 - m_H g L_\rho] = L_H F_{dif}(s)$$

$$\frac{P(s)}{F_{dif}(s)} = G_\rho(s) = \frac{L_H}{I_\rho s^2 - m_H g L_\rho} \quad (2.47)$$

Donde  $P(s)$  y  $F_{dif}(s)$  son las transformadas de Laplace de  $\rho$  y  $F_{dif}$  respectivamente.

Para la ecuación (2.45) se puede aplicar la transformada de Laplace de igual manera, de donde se obtiene la siguiente expresión:

$$E(s) s^2 = \frac{L_M}{I_\epsilon} F_{sum}(s) - \frac{K g_\epsilon}{I_\epsilon} \quad (2.48)$$

Considerando que la salida del sistema es la variable  $E(s)$  y la entrada es la suma de las fuerzas o  $F_{sum}(s)$ , la función de transferencia en forma de fracción se puede encontrar dividiendo ambas partes de la ecuación para  $F_{sum}(s)$ :

$$\frac{E(s)}{F_{sum}(s)} = \frac{L_M}{s^2 I_\epsilon} - \frac{K g_\epsilon}{s^2 I_\epsilon F_{sum}(s)} \quad (2.49)$$

Como se puede observar en la ecuación (2.49), el término de la derecha incluye  $F_{sum}(s)$ , lo cual imposibilita relacionar directamente esta entrada con la salida  $E(s)$ . Esto se debe a que, para este sistema, la fuerza de la gravedad se representa en el término  $K g_\epsilon$  y dicha fuerza actúa como otra entrada para este sistema, el término negativo indica que es una entrada que hace que el sistema se acelere “hacia abajo”. Dado que en este caso la única fuerza que se puede controlar es la de  $F_{sum}(s)$ , la función de transferencia que se considera

es la que se muestra en la ecuación (2.50). En este caso, la fuerza de la gravedad (representada por el efecto de  $Kg_\epsilon$ ) actúa como una perturbación constante, la cual se corrige mediante el término integral de un controlador PID.

$$\frac{E(s)}{F_{sum}(s)} = G_\epsilon(s) = \frac{L_M}{I_\epsilon s^2} \quad (2.50)$$

Por último, en la ecuación (2.46) de la variable travel se puede notar que todavía existe una no linealidad que resulta del producto de  $\rho$  y  $F_{sum}$ . Para obtener una función de transferencia se considera que  $F_{sum}$  toma un valor constante, que es igual al necesario para mantener al helicóptero en posición horizontal. Esta suposición resulta de la consideración del punto de operación que se indica en la ecuación (2.42) ya que si el ángulo de elevation se mantiene cercano a  $0^\circ$  (horizontal), el valor de  $F_{sum}$  tendrá un valor prácticamente constante.

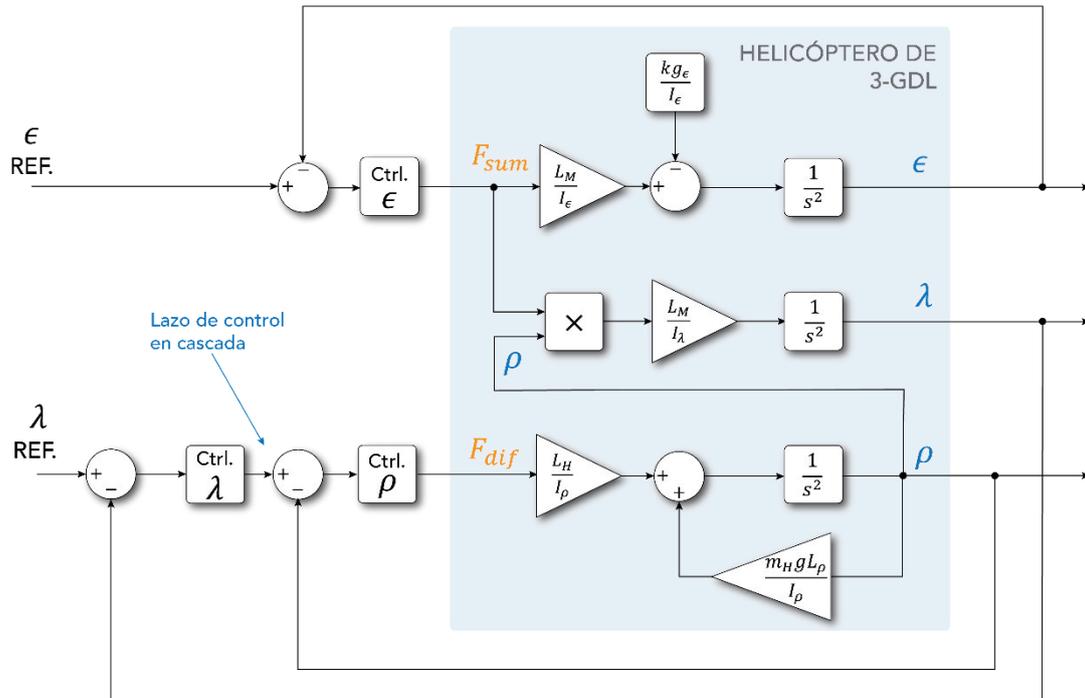
Esta suposición se contrapone a lo expuesto en las Ecuaciones (2.48 a 2.50) en las cuales se mostraba que el valor de  $F_{sum}(s)$  es la variable de entrada del sistema de  $\epsilon$ . Para evitar confusiones cabe recalcar que esto solo se aplica para los casos en los que se modifica el ángulo de elevation. Cuando dicho ángulo se mantiene constante,  $F_{sum}$  también se considera constante. Sin embargo, en los instantes que  $F_{sum}$  y  $\epsilon$  cambian, el resultado es una perturbación en el sistema de travel, el cual se corrige mediante un lazo cerrado de control. El acoplamiento entre las variables se puede visualizar de mejor manera en el diagrama de la Figura 2.51.

Con esta consideración en mente, se puede notar que el movimiento en travel dependerá únicamente del ángulo  $\rho$  ya que como se indicó en la Figura 2.50 a), la componente horizontal de la fuerza es la que genera aceleración en travel.

Tomando esto en cuenta, el modelo reducido para travel es el que se indica en la Ecuación (2.51) cuya salida es la variable  $T(s)$  y la entrada es el ángulo pitch,  $P(s)$ .

$$\frac{T(s)}{P(s)} = G_\lambda(s) = \frac{L_M F_{sum}}{I_\lambda s^2} \quad (2.51)$$

Esta ecuación muestra claramente el acoplamiento existente entre las variables  $\rho$  y  $\lambda$  y además indica que el lazo de control deberá tener una estructura en cascada, ya que la variable pitch (lazo interno) controla a la variable travel (lazo externo) como se muestra en la Figura 2.51.



**Figura 2.51.** Diagrama de bloques del modelo simplificado del helicóptero de 3-GDL con el esquema de control en cascada para  $\lambda$  y  $\rho$ .

### 2.3.2. DISEÑO DE CONTROLADORES

Los submodelos simplificados encontrados anteriormente sirven para determinar la estructura de los controladores para cada variable. A continuación, se mostrará el diseño de los controladores para cada variable considerando que los 3 subsistemas que se tienen son:

- Pitch: entrada  $F_{dif}$ , salida  $\rho$ .
- Elevation: entrada  $F_{sum}$ , salida  $\epsilon$ .
- Travel: entrada  $\rho$ , salida  $\lambda$ .

Para cada variable se ha considerado un esquema de control PID debido a la facilidad de su implementación y sintonización de ganancias.

#### 2.3.2.1. Controlador Pitch

Para el diseño y sintonización del controlador de la variable pitch se considera el modelo simplificado que se muestra en la Ecuación (2.47). Reemplazando con los valores indicados en la Tabla 2.8 se obtiene la función de transferencia con valores numéricos, como se muestra a continuación:

$$G_{\rho}(s) = \frac{L_H}{I_{\rho}s^2 - m_HgL_{\rho}} = \frac{0.72}{0.0201s^2 - 0.1702}$$

$$G_{\rho}(s) = \frac{35.82}{s^2 - 8.4677} \quad (2.52)$$

Dado que el sistema posee un polo doble en el semiplano derecho tiene una característica inestable y su respuesta ante una entrada paso será no acotada. Por este motivo se utiliza la herramienta PID Tool de Matlab, que permite sintonizar un controlador adecuado de forma rápida y efectiva. Tomando esto en cuenta se ha considerado una respuesta en lazo cerrado de 3s de tiempo de establecimiento y 20% de sobreimpulso como máximo. El controlador PID obtenido en Matlab con estas características se indica en la Ecuación (2.52)

$$C_{\rho}(s) = K_p + K_i \frac{1}{s} + K_d s = 2.372 + 2.018 \frac{1}{s} + 1.651s \quad (2.53)$$

Las características de las respuestas del sistema en lazo cerrado al utilizar este y los demás controladores diseñados se muestra en el capítulo de resultados, donde se prueban los lazos de control de cada variable y se muestran las características transitorias.

### 2.3.2.2. Controlador Elevation

De manera similar al caso anterior, se reemplazan los valores numéricos de la Tabla 2.8 en la función de transferencia de la Ecuación (2.50)

$$G_{\epsilon}(s) = \frac{L_M}{I_{\epsilon}s^2} = \frac{1.0256}{s^2} \quad (2.54)$$

También se ha empleado la herramienta PID Tool de Matlab para la sintonización del controlador en esta variable. Cabe recalcar que, como se puede observar, la función de transferencia simplificada es un integrador doble, por lo que el controlador adecuado sería un PD. Sin embargo, el efecto de la gravedad genera una perturbación que se manifiesta como la constante  $Kg_{\epsilon}$  en la Ecuación (2.45). Por tal motivo se escoge la estructura PID para lo cual se obtienen las constantes que se indican en la Ecuación (2.54).

$$C_{\epsilon}(s) = 6.849 + 2.262 \frac{1}{s} + 4.075s \quad (2.55)$$

La efectividad de este controlador se evalúa en el capítulo de resultados, en el cual se muestran las características transitorias del sistema obtenido.

### 2.3.2.3. Controlador Travel

Siguiendo el mismo procedimiento que para las variables anteriores se encuentra la función de transferencia linealizada para este sistema. En este caso se considera que el valor de  $F_{sum}$  permanece constante y es el necesario para mantener al brazo de soporte de manera horizontal. Como se explicó anteriormente, el valor de esta fuerza no es realmente constante, sin embargo, no varía demasiado mientras el ángulo de  $\epsilon$  permanezca igualmente constante y en este caso se cumple dicha condición y se considera  $\epsilon = 0^\circ$ . Para este valor del ángulo elevation se ha determinado de manera experimental el valor que  $F_{sum}$  debe tener, para lo cual se ha colocado una balanza exactamente debajo del cuerpo del helicóptero mientras se mantiene el brazo de soporte a  $0^\circ$ . Con este arreglo se ha tomado la lectura de la balanza que midió 226g que corresponde a una fuerza de aproximadamente 2.21N. Tomando esto en cuenta se pueden reemplazar el resto de los valores en la Ecuación (2.51), de donde se obtiene la función de transferencia para el controlador de travel.

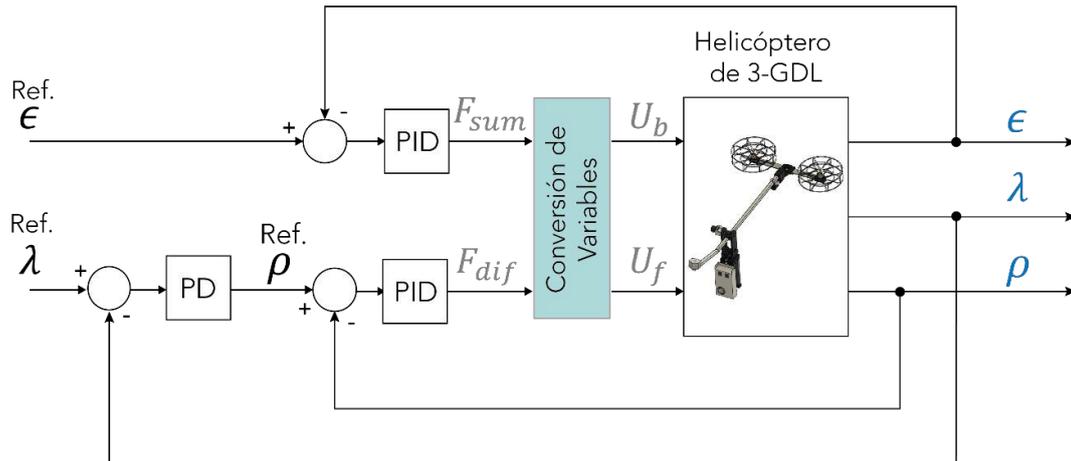
$$G_\lambda(s) = \frac{1.595}{s^2} \quad (2.56)$$

La estructura de esta función de transferencia es un doble integrador, por lo que se considera un controlador de tipo PD. Las consideraciones de diseño para dicho controlador son un tiempo de establecimiento menor a 8s y un máximo sobreimpulso de 20%, con lo cual el control tipo PD obtenido con la herramienta PID tool de Matlab se presenta en la Ecuación (2.57).

$$C_\lambda(s) = 0.321 + 0.5614s \quad (2.57)$$

No obstante, en la Sección 3.3.3 se presentará que este controlador tiene ganancias demasiado altas porque la respuesta es muy oscilatoria. Los cambios realizados para obtener una mejor respuesta se detallan en dicha sección.

Con estos controladores diseñados, el esquema final de control del helicóptero de 3-GDL se indica en la Figura 2.52. En este esquema el bloque de “Conversión de Variables” considera la transformación de  $\begin{bmatrix} F_{sum} \\ F_{dif} \end{bmatrix}$  a  $\begin{bmatrix} F_f \\ F_b \end{bmatrix}$  y posteriormente a las acciones de control en porcentaje  $\begin{bmatrix} U_f \\ U_b \end{bmatrix}$ , las cuales se determinan utilizando la Ecuación (2.5), que relaciona la fuerza  $F$  con la acción de control en porcentaje,  $U$ .



**Figura 2.52.** Esquema de control para el helicóptero de 3-GDL.

### 2.3.2.4. Discretización de Controladores

Para la discretización de los controladores diseñados se ha utilizado la regla de integración en atraso debido a que en el microcontrolador se puede implementar con gran facilidad y no presenta grandes diferencias con los otros equivalentes de integración numérica, esto debido a que el tiempo de muestreo es muy pequeño  $T = 9.5ms$ . Este tiempo es lo que le toma al microcontrolador ejecutar el lazo principal de control. Adicionalmente, la convención es  $1/30$  del tiempo de estabilización [55], que como se mostrará en el siguiente capítulo es alrededor de 6s. Por lo tanto, el tiempo máximo de muestreo que se puede utilizar es 0.2s, por lo que  $9.5ms$  es un valor favorable. Como se conoce, la regla en atraso viene dada por

$$s = \frac{1 - z^{-1}}{T} \quad (2.58)$$

Reemplazando esta expresión en la estructura básica de un PID se encuentra:

$$C(s) = K_p + K_i \frac{1}{s} + K_d s = K_p + K_i \frac{T}{1 - z^{-1}} + K_d \frac{1 - z^{-1}}{T} \quad (2.59)$$

Para encontrar la correspondiente ecuación en diferencias y poder implementar en código del microcontrolador se debe recordar la definición de la Transformada Z que indica que:

$$Z^{-1}\{z^{-n}F(z)\} = f(k - n) \quad (2.60)$$

Donde  $Z^{-1}$  es la transformada Z inversa,  $n$  indica el valor del retardo,  $f$  es una señal en el tiempo  $k$  discreto y  $F$  es su correspondiente Transformada Z. Considerando esto se puede encontrar la expresión para implementar la ley de control en programación.

Desde la Ecuación (2.59) se observa que la parte proporcional se mantiene de la misma forma  $K_p$ . La parte derivativa  $D$  se transforma en:

$$D = K_d \frac{1 - z^{-1}}{T} E(z) \rightarrow \frac{K_d}{T} [e(k) - e(k - 1)] \quad (2.61)$$

Para la parte integral  $I$  se utiliza un acumulador de modo que se vaya sumando el valor actual del error a todas las sumas anteriores, con lo cual se hace:

$$I = K_i T [e(k) + e(k - 1) + e(k - 2) + \dots] \quad (2.62)$$

Lo cual mediante la Transformada Z se puede encontrar que es matemáticamente igual a la expresión (2.63) y es lo que se indica en la definición de la regla de integración en atraso.

$$I = K_i \frac{T}{1 - z^{-1}} E(z) \quad (2.63)$$

A continuación, se muestra un ejemplo de esta implementación para el controlador de la variable elevation, tomando en cuenta las consideraciones que se acaban de mencionar.

```
#Solo integro elevation mientras no hayan Llegado al Limite:
if not((LimiteF and LimiteB) or LimiteWU_E):
    eETotal = eETotal + eE_0    #Para integral elevation

#CONTROL PID elevation:
uE_0 = 6.849*eE_0 + 2.262*Tm*eETotal + 4.075*(1/Tm)*(eE_0 - eE_1)
```

El acumulador `eETotal` sirve para la implementación de la acción integral, y como se puede ver en la última línea, la constante de la parte integral se multiplica al tiempo de muestreo  $T_m$  y en la derivativa se divide. Por último, `eE_0` es el error actual, mientras que `eE_1` es el valor anterior, es decir  $e(k - 1)$ .

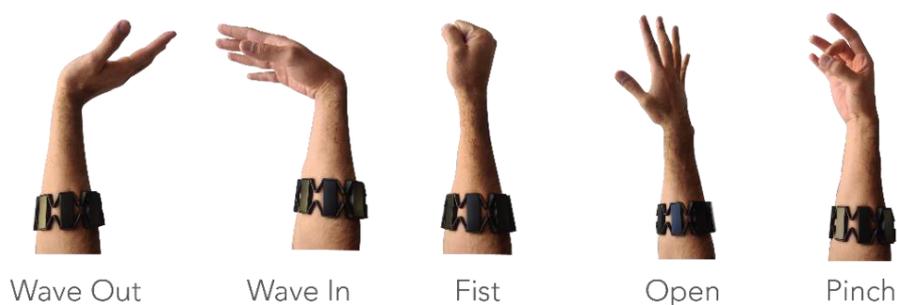
## 2.4. INTEGRACIÓN CON EL SISTEMA DE RECONOCIMIENTO DE GESTOS

Una vez que se ha diseñado el sistema de control, se puede enlazar con el reconocimiento de gestos. El sistema de HGR se encarga de generar un valor de referencia que se ingresa al lazo de control de la Figura 2.52. Dependiendo del gesto o movimiento realizado, se envía un valor de setpoint específico que la plataforma debe alcanzar.

A continuación, se explican los componentes necesarios para enlazar el sistema de reconocimiento con el movimiento de la plataforma.

### 2.4.1. FUNCIONAMIENTO DEL SISTEMA DE HGR

El sistema de reconocimiento de gestos de la mano (HGR) utilizado en el presente proyecto fue desarrollado en Matlab. Tiene la capacidad de reconocer los 5 gestos de la mano que se mencionaron en el capítulo anterior y estos son: Wave Out, Wave In, Fist, Open y Pinch, además de la posición de reposo, indicados en la Figura 2.53.

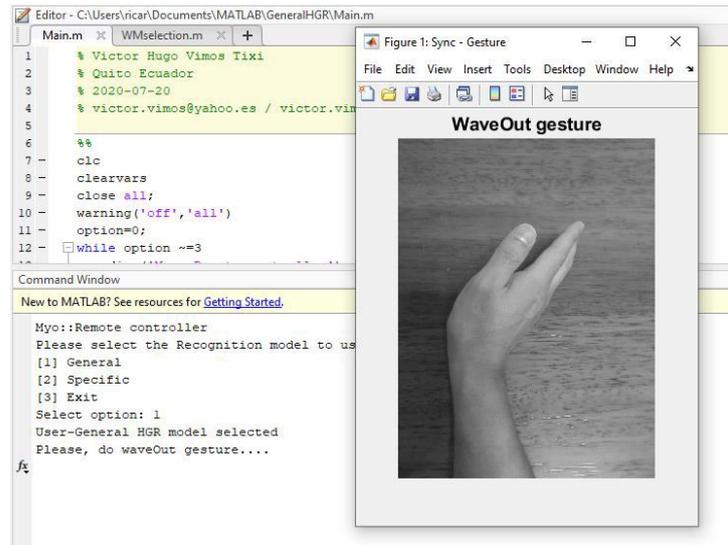


**Figura 2.53.** Gestos de la mano que reconoce el sistema utilizado.

Para utilizar este sistema se emplea un tipo de programa denominado *wrapper* que toma el kit de desarrollo de software del sensor Myo y lo hace compatible con Matlab. Este programa se llama MyoMex, fue desarrollado por Mark Tomaszewski y se encuentra disponible de forma gratuita en su repositorio de Github [56]. Con este software se puede leer la información proveniente del sensor Myo e importarla y procesarla en código de Matlab. Adicionalmente, el sistema de HGR utiliza un compilador de C++ denominado MinGW-w64 con lo cual se puede hacer uso del reconocimiento. El proceso de instalación y uso del sistema se explica a detalle en el Anexo C.

#### 2.4.1.1. Utilización del sistema de reconocimiento

Luego de realizar la instalación de todos los complementos y paquetes de software como se indica en el Anexo C se puede utilizar el sistema de HGR con el sensor Myo. Para empezar, se debe conectar el receptor USB del sensor a la computadora y verificar la conexión mediante el software que incluye el fabricante. Posteriormente, para usar el sistema de reconocimiento se ejecuta el archivo de Matlab llamado "Main.m". Al correr este programa se selecciona la opción 1 para Usuario General y empieza con el gesto de sincronización Wave Out que se debe realizar durante unos segundos para calibrar el sistema, tal como se indica en la Figura 2.54.



**Figura 2.54.** Sistema de reconocimiento de gestos. Al iniciar el programa se realiza el gesto de sincronización.

La calibración que se ejecuta al realizar este gesto es la corrección de orientación que se mencionó en el capítulo anterior. Al realizar el gesto de Wave Out, la mayor cantidad de energía se concentra en la parte posterior del antebrazo, de modo que se hace la corrección para tomar como referencia al detector que se encuentra directamente encima de esta zona. Luego de la calibración el sistema empieza a funcionar inmediatamente y la salida que produce es una etiqueta que corresponde al gesto realizado (que pueden ser “waveOut”, “waveIn”, “fist”, “open”, “pinch”, o “noGesture”, según corresponda).

El programa descrito está diseñado para implementar la aplicación (que en este caso es comandar la plataforma mediante los gestos) en medio de un lazo while en el código del sistema de reconocimiento. Esto, sin embargo, presenta un inconveniente importante que es el retardo generado por el procesamiento del sistema HGR ya que la comunicación con la plataforma debe realizarse de forma rápida para que se pueda tener información del sistema en tiempo real. No obstante, el sistema de reconocimiento tiene un retraso de entre 0.5s hasta 1s. Debido a esto la aplicación de comunicación con el helicóptero no se puede implementar directamente en el mismo lazo while, para lo cual se ha realizado un modelo en Simulink y una aplicación en App Designer de Matlab, esta última también sirve de interfaz gráfica.

## 2.4.2. INTEGRACIÓN DE COMPONENTES

Debido a lo mencionado, la velocidad de ejecución del lazo del sistema de HGR es lenta comparada con la comunicación que debe existir entre la PC y la plataforma. Por esta razón se han elaborado diferentes componentes que funcionan de manera independiente y es

necesario comunicarlos entre sí para que el sistema se integre en uno solo. Los programas que se deben interconectar son:

- Instancia 1 de Matlab: sistema de HGR
- Instancia 2 de Matlab:
  - Modelo en Simulink: interfaz de comunicación inalámbrica con la plataforma (mediante módulos XBee) para enviar comandos y valores de referencia.
  - Interfaz gráfica desarrollada en App Designer

### 2.4.2.1. Instancia 1 de Matlab

La ventana 1 de Matlab es la que se encarga de realizar el reconocimiento de gestos de la mano. En esta ventana se corre el programa “Main.m” en el cual se captura la información entregada por el sensor Myo y se realiza el algoritmo de reconocimiento de gestos de la mano, mediante el cual se generan variables que indican el gesto que se ha realizado. Posteriormente este dato es enviado a los otros componentes del sistema, en los cuales se genera un valor de setpoint para controlar el movimiento del helicóptero de 3-GDL.

En el programa desarrollado existen dos variables que representan el gesto realizado, la primera se llama `newGesture` es de tipo string y muestra el nombre del gesto en forma de una cadena de caracteres. Por otro lado, se encuentra `ges`, la cual indica el gesto en forma de un carácter numérico del 1 al 6 de acuerdo con lo que se muestra en la Figura 2.60

Gesto						
Variable						
<b>newGesture</b>	waveOut	waveIn	fist	open	pinch	noGesture
<b>ges</b>	1	2	3	4	5	6

**Figura 2.55.** Las variables que indican el gesto realizado son `ges` y `newGesture`.

En este proyecto se ha utilizado la variable `ges` ya que es un solo byte que contiene toda la información necesaria. Este número se envía mediante un socket UDP a la interfaz gráfica del sistema, donde se transforma en un valor de setpoint para la plataforma. El cual, a su vez, se transmite al helicóptero de 3-GDL mediante un módulo de comunicación

desarrollado en Simulink. Estos dos componentes se ejecutan en una segunda instancia de Matlab que se describe a continuación.

#### **2.4.2.2. Instancia 2 de Matlab**

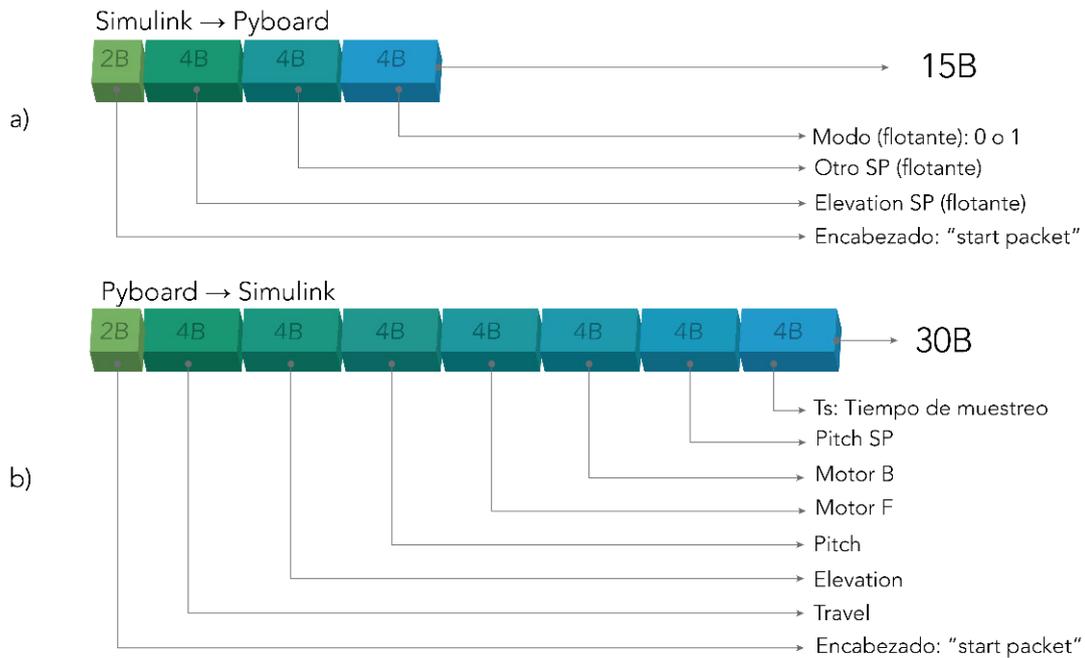
La primera ventana de Matlab incluye únicamente el sistema de reconocimiento de gestos de la mano, para lo cual se ejecuta el programa “Main.m” como se ha explicado. Esto puede funcionar de manera independiente, sin embargo, lo único que se puede hacer es visualizar el gesto realizado en la ventana de comandos de Matlab. Para enlazarlo con el helicóptero es necesario abrir una nueva ventana de Matlab que contiene los componentes de comunicación con la plataforma y la interfaz gráfica.

##### *2.4.2.2.1. Modelo en Simulink de conexión Inalámbrica*

A nivel físico, la comunicación inalámbrica se realiza mediante los módulos XBee, los cuales se conectan luego de realizar la configuración detallada en las secciones anteriores. Sin embargo, para comunicar con la aplicación realizada en Matlab se ha utilizado un modelo en Simulink, ya que, internamente en esta herramienta los bloques se encuentran programados en C, lo cual permite que el programa se ejecute a una mayor velocidad, comparado con el código en Matlab.

En la comunicación desde la PC hacia el microcontrolador se envía un paquete de datos conformado por 15 bytes que incluyen la información que se muestra en la Figura 2.56.a) Los primeros dos bytes corresponden a los caracteres “sp” que indican el inicio del paquete de datos y simbolizan *start packet*. Estos caracteres de inicio se utilizan para controlar el flujo de bytes hacia el Pyboard. Internamente, en los bloques de envío y recepción serial en Simulink se configura el parámetro “Header” con los caracteres “sp”.

A continuación, se envían 4 bytes que corresponden al setpoint de la variable Elevation en un número flotante de 32 bits. Posteriormente se manda el valor *Otro SP* que puede corresponder al setpoint de la variable Travel o Pitch dependiendo del modo de operación, lo cual se explicará más adelante. Finalmente se envía el identificador de este modo de operación que corresponde a “0” para *modo gestos* (setpoint de travel) y “1” para el *modo IMU* (setpoint de pitch). Esta distinción se hace para cambiar el comando para el movimiento de la plataforma, lo cual le da la cualidad de *multimodo*.



**Figura 2.56.** Paquetes de datos transferidos en la comunicación entre Simulink y el microcontrolador. a) Datos enviados de la PC (Simulink) a la placa Pyboard. b) Datos enviados desde la placa a la PC.

Para la comunicación desde el microcontrolador hacia la PC se manda el mismo encabezado de "sp" seguido de 7 números flotantes de 32 bits, lo cual da lugar a un paquete de 30 bytes y las variables que se envían se muestran en la Figura 2.56.b). Pitch, Elevation y Travel son los valores de los ángulos medidos por los encoders en grados; Motor B y Motor F son las señales de control enviadas a cada motor en porcentaje; PitchSP es el setpoint del controlador de la variable pitch. Este dato se envía para conocer la señal de control del eje elevation (ya que es un control en cascada); y finalmente el tiempo de muestreo  $T_s$  se determina midiendo el tiempo de ejecución del lazo principal del programa del microcontrolador. El valor de  $T_s$  es aproximadamente 9.5ms y se utiliza para el cálculo de las leyes de control de acuerdo con las ecuaciones mostradas en la sección de diseño de controladores. El dato del tiempo de muestreo se ha incluido entre los valores de comunicación inalámbrica ya que permite determinar fallas en la ejecución del lazo y la recepción de información. Esto se debe a que ciertas funciones y algoritmos en el microcontrolador pueden generar retardos en el lazo principal, con lo cual la visualización del tiempo de ejecución permite realizar programas más adecuados para el sistema.

#### 2.4.2.2.1. Interfaz gráfica

El otro componente que se ejecuta en la segunda ventana de Matlab es la interfaz gráfica. Esta es una aplicación desarrollada en el entorno de App Designer que se encarga de

presentar de forma visual las variables relacionadas con el movimiento del helicóptero y los gestos detectados por el sistema de HGR. Para el presente proyecto se han elaborado 3 ventanas en la interfaz que son la de inicio, información y aplicación.

- **Ventana de Inicio**

Esta ventana contiene el título del proyecto y 3 botones que sirven para acceder a la ventana de información, a la de aplicación y para cerrarla, como se indica en la Figura 2.57. La aplicación es el panel de control del sistema, en el cual se muestra la información relevante para el control de la plataforma mediante los gestos de la mano, por lo cual el botón de “arrancar aplicación” se mantiene deshabilitado hasta que se encuentren abiertas y corriendo las ventanas de reconocimiento de gestos (HGR) y el modelo en Simulink de comunicación inalámbrica. Como se observa en la Figura 2.57, el estado de dichos componentes se visualiza mediante indicadores luminosos.



**Figura 2.57.** Ventana de inicio con sus elementos.

- **Ventana de información**

La ventana de información contiene básicamente el manual de usuario del uso de la interfaz y de la plataforma. En esta ventana se muestra el manual que se presenta en el Anexo D, para lo cual se ha incluido un objeto HTML en una ventana de App Designer de Matlab. Esto se ha realizado para poder navegar por el manual de usuario de forma interactiva y que se presente de manera sencilla para el usuario.

Como se puede observar en la Figura 2.58, esta ventana tiene una barra de navegación independiente, mediante la cual se puede acceder a las diferentes secciones del manual de usuario de forma rápida. Adicionalmente se tiene un ícono para regresar a la primera

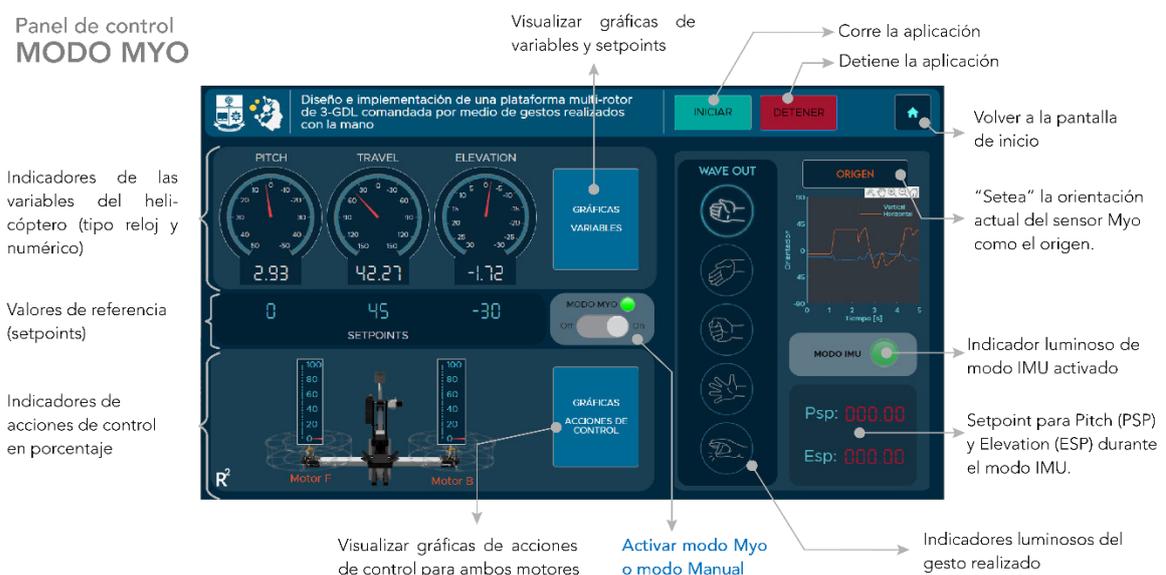
página del manual y el botón para volver a la ventana de inicio que se presentó anteriormente.



**Figura 2.58.** Ventana de Información y sus botones.

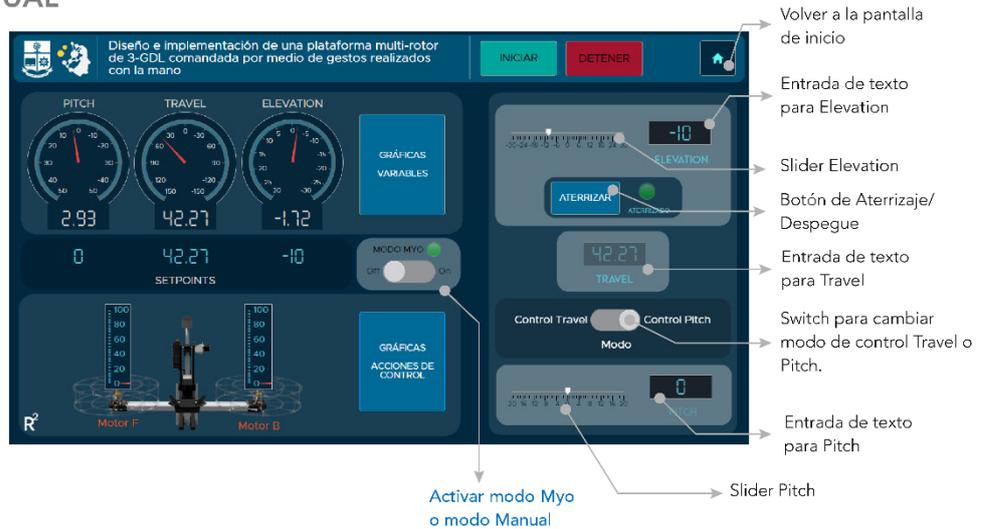
- **Ventana de aplicación**

Por último, la ventana de aplicación incluye el panel de control para el sistema. En esta se pueden visualizar los gestos proporcionados por el sistema de HGR y la realimentación de las variables en el helicóptero de 3-GDL. Además, en esta se puede escoger el modo de funcionamiento manual o con el sensor Myo Armband, de acuerdo con la metodología que se explica en los siguientes párrafos. En las Figuras 2.59 y 2.60 se muestran los elementos de la ventana de aplicación (o panel de control) durante el modo Myo y modo Manual respectivamente.



**Figura 2.59.** Ventana de aplicación en Modo Myo con sus elementos.

Panel de control  
**MODO MANUAL**



**Figura 2.60.** Ventana de aplicación en Modo Manual con sus elementos.

Una explicación más detallada del funcionamiento y uso de la interfaz de usuario se puede encontrar en el Anexo D.

#### 2.4.2.3. Conexión entre las dos instancias de Matlab

Como se explicó anteriormente, dado que el procesamiento del sistema de reconocimiento tiene un tiempo de retardo de más de 0.5s, la interfaz se realiza en una ventana de Matlab por separado de modo que pueda comunicarse con el modelo en Simulink, de acuerdo con lo mencionado en los párrafos anteriores. Para enlazar ambas instancias de Matlab se utilizó un “socket” UDP mediante el cual se comunican ambas ventanas a través de una dirección IP y un puerto único para cada ventana. La dirección utilizada es la local 127.0.0.1 y los puertos seleccionados fueron 1111 para la aplicación de HGR y 2021 para la ventana de la interfaz. Ambos programas corren en ventanas independientes de Matlab. Mediante este enlace se envía el gesto realizado desde la ventana del HGR hacia la ventana de la interfaz en forma de un solo carácter numérico del 1 al 6, de acuerdo con la codificación de la Figura 2.55.

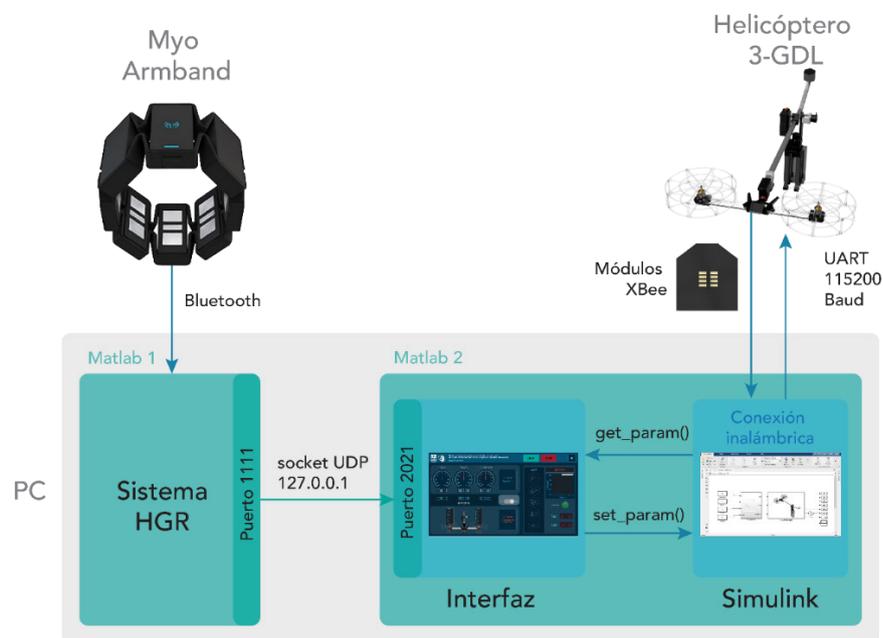
#### 2.4.2.4. Conexión entre Matlab y Simulink

Mediante el modelo en Simulink explicado en la parte anterior, se enlaza la PC con el microcontrolador en la plataforma. Sin embargo, existe la necesidad de enlazar el modelo en Simulink con la ventana de Matlab que maneja la interfaz gráfica. Esto es necesario ya que los valores de setpoint se generan en la interfaz y se comunican a la plataforma a través del modelo en Simulink. Esta conexión se realiza por medio de comandos propios

de Matlab que permiten acceder a los bloques de Simulink. En este caso los principales comandos que se utilizan son:

- `set_param()`: permite escribir parámetros sobre los bloques del modelo en Simulink. Mediante este comando se escriben los valores de setpoint en las entradas para la comunicación.
- `get_param()`: este comando permite acceder a la información de los bloques. Uno de los parámetros más importantes que se utilizan es el `RuntimeObject` ya que este es el que contiene la información del modelo durante la ejecución. Es decir, los datos en tiempo real del modelo.

Mediante los métodos explicados se integran las tres partes principales del sistema, y con esto se comanda el movimiento del helicóptero de 3-GDL por medio de gestos de la mano. En la Figura 2.61 se ilustran cada una de las partes y su funcionamiento.



**Figura 2.61.** Esquema de interconexión de los componentes del sistema.

### 2.4.3. CONTROL DE MOVIMIENTO MULTIMODO

El movimiento de la plataforma se controla mediante comandos que corresponden con los gestos mencionados anteriormente y también mediante movimientos (horizontal y vertical) del antebrazo. Estos dos tipos de control se denominan modo gesto y modo IMU, respectivamente. Adicionalmente, la plataforma también puede ser controlada sin la necesidad de utilizar el sensor Myo Armband, en cuyo caso el modo de operación se

denomina modo manual, ya que los valores de referencia se ingresan de forma numérica en la interfaz gráfica.

### 2.4.3.1. Modo Manual

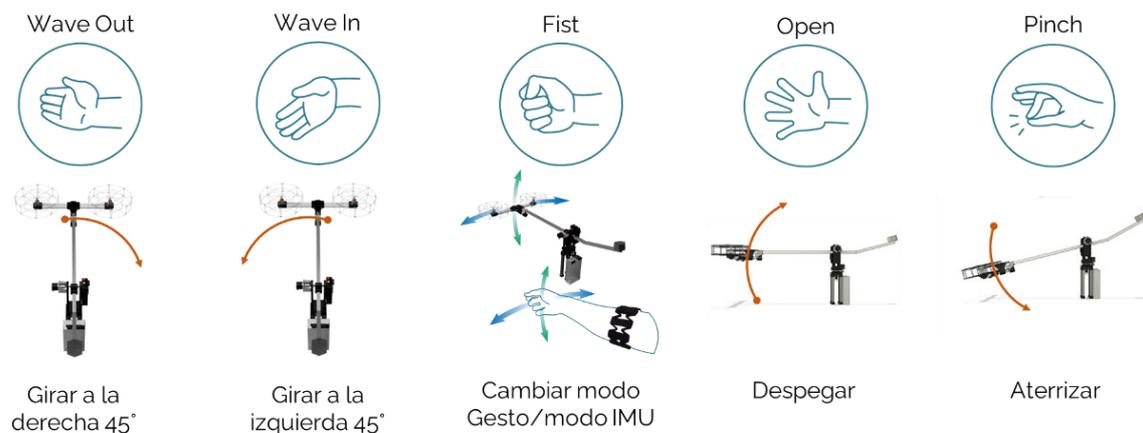
En este modo el usuario modifica los valores de setpoint directamente en la interfaz por medio de una entrada de texto o con sliders. Esta funcionalidad se denomina “Modo Manual” y en este caso se ignora la información que proporciona el sensor, tanto en forma de gestos como en forma de orientación con la IMU.

Para este modo de operación se debe escoger la variable que se va a modificar entre pitch y travel, ya que ambas están acopladas entre sí y no se pueden mover de forma independiente. Para esto se utiliza un switch que cambia la variable Modo a 0 para control de travel y 1 para control de pitch.

### 2.4.3.2. Modo Myo

#### 2.4.3.2.1. Modo Gesto

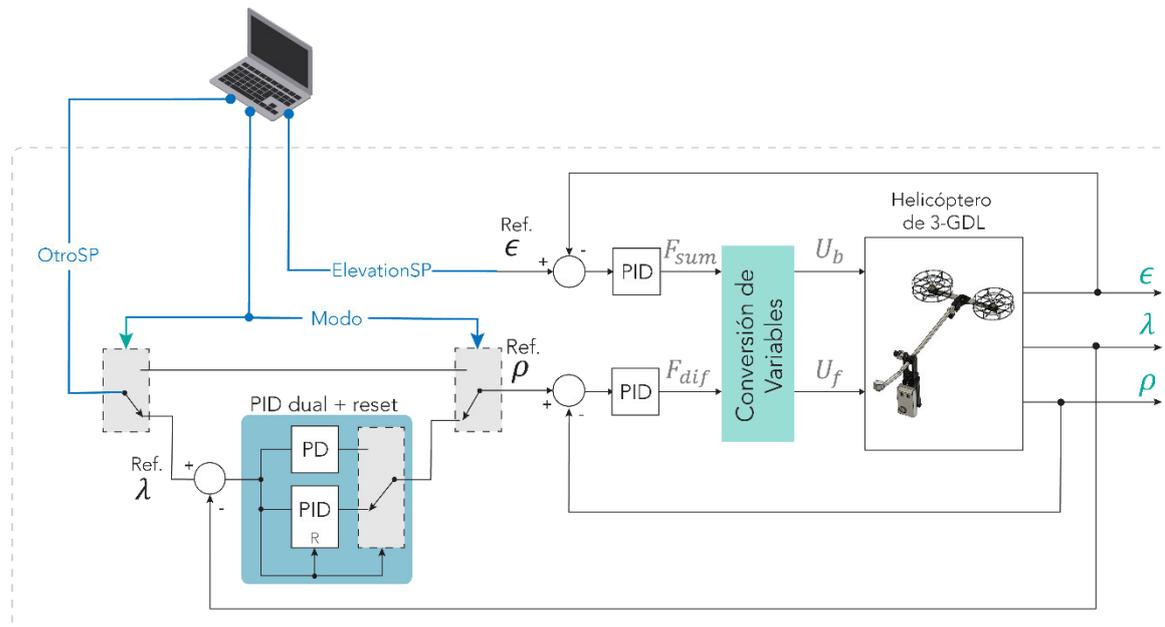
El modo gesto es la forma de operación que hace uso del sistema de HGR. Es decir, los gestos realizados sirven para generar un valor de setpoint en el sistema de control del helicóptero de 3-GDL. En este caso el número de “modo” en el paquete de datos toma el valor de 0 y el setpoint que se envía es el correspondiente al control de la variable travel. Es decir, se aplica el sistema de control en cascada y la referencia del controlador de pitch se genera de manera automática. Los comandos que generan cada gesto se detallan en la Figura 2.62.



**Figura 2.62.** Comandos de movimiento del helicóptero asociados a cada gesto de la mano.

#### 2.4.3.2.2. Modo IMU

Para acceder a este modo se realiza el gesto fist y durante este tipo de operación, el control de la plataforma se realiza mediante los movimientos horizontal y vertical del antebrazo. Para detectar estos movimientos se utiliza la unidad de medición inercial (IMU) que se encuentra en el sensor Myo, por esta razón recibe el nombre de *modo IMU*. Para lograr esto se desconecta el sistema de control de la variable travel y solo se mantiene activo el control de pitch. En este caso, el valor de “OtroSP”, en la Figura 2.63, corresponde a la referencia de pitch y el movimiento horizontal del antebrazo se mapea al de esta variable. Considerando este método, el esquema de control de la Figura 2.52 se modifica para incluir las variables de “Modo” y “OtroSP”, mismo que se presenta en la Figura 2.63 y que representa el esquema de control multimodo. En este diagrama las líneas de color azul simbolizan la conexión inalámbrica desde la PC al microcontrolador



**Figura 2.63.** Esquema de control con función multimodo.

Como se mencionó en el capítulo anterior, el sensor Myo posee un giroscopio, acelerómetro y magnetómetro, y realiza la fusión de las señales entregadas por cada elemento para entregar un valor de orientación en forma de cuaterniones [30]. Los cuaterniones son una extensión de los números complejos y resultan de gran utilidad para la descripción y cálculo de rotaciones tridimensionales [57]. Por esta razón se emplean para determinar la orientación del sensor Myo en el espacio.

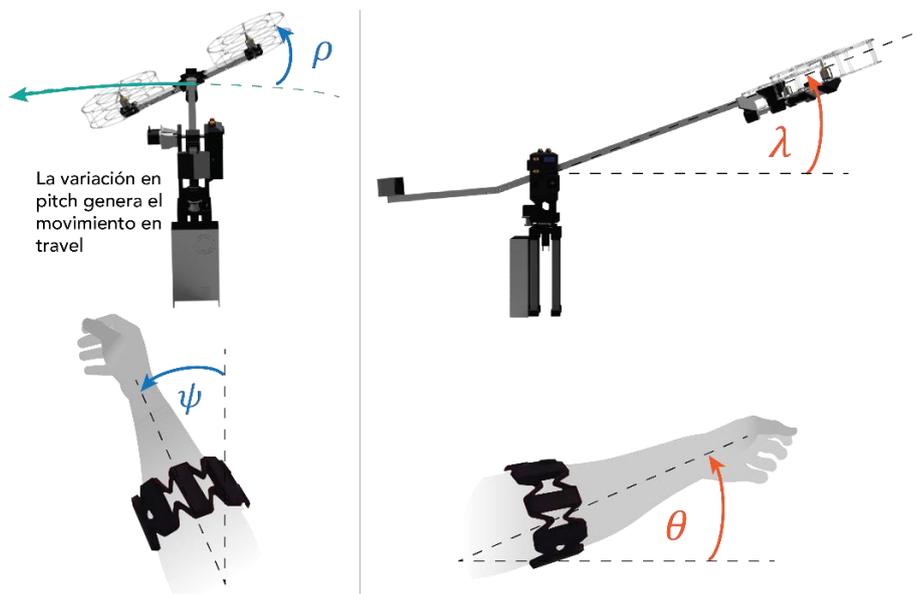
Para el control de movimiento en modo IMU se detectan los desplazamientos angulares del antebrazo en sentido horizontal para la variable travel (enviando un valor de referencia

al controlador de pitch) y en sentido vertical para la variable elevation (modificando directamente la referencia de elevation). Para detectar estos dos ángulos es necesario transformar la información en forma de cuaterniones a posiciones angulares (ángulos de Euler), para lo cual se utilizaron las expresiones de las Ecuaciones (2.64) y (2.65) [57].

$$\theta = \arcsin(2(q_1q_3 - q_2q_4)) \quad (2.64)$$

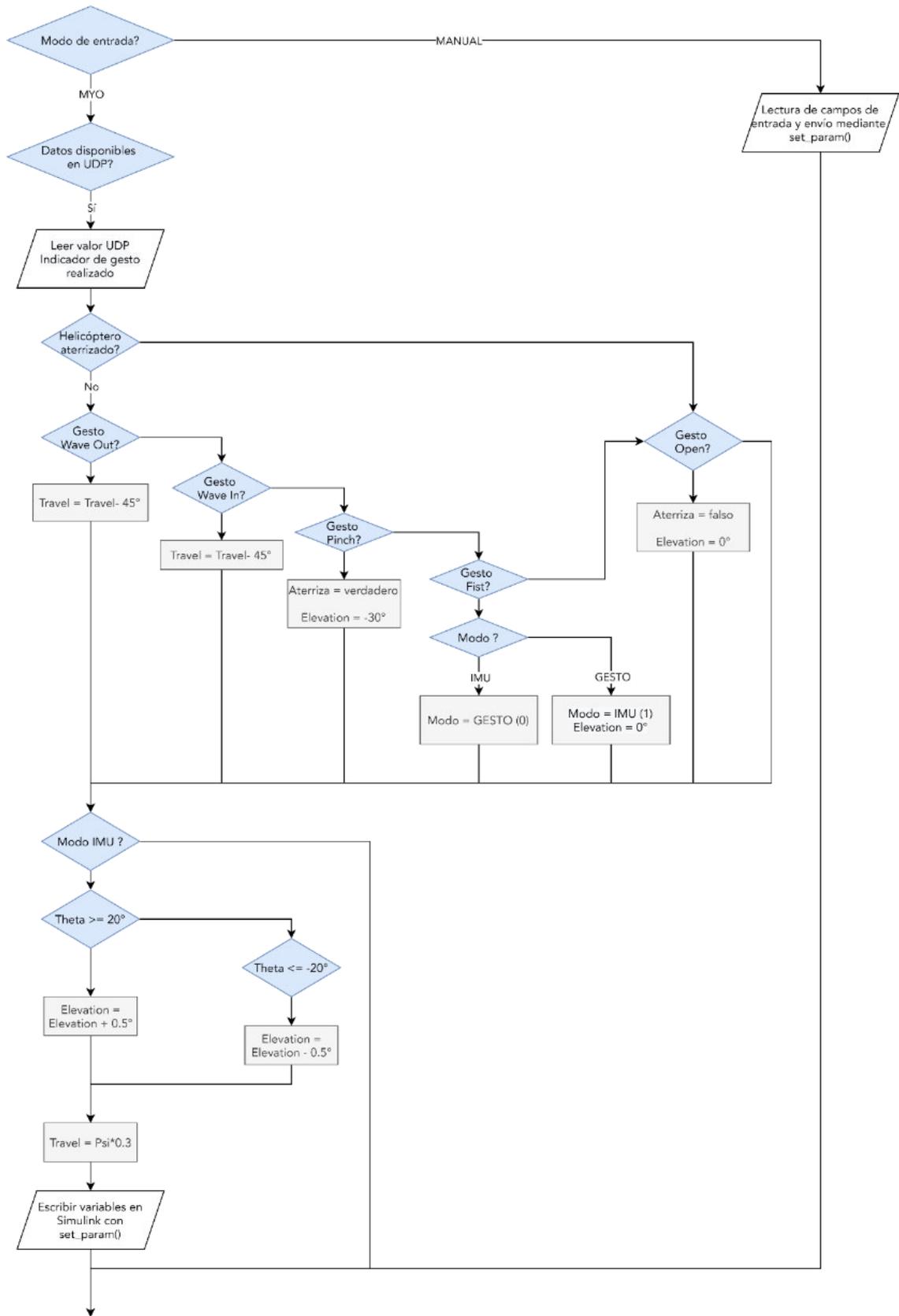
$$\psi = \arctan\left(\frac{2(q_1q_4 + q_2q_3)}{1 - 2(q_3^2 + q_4^2)}\right) \quad (2.65)$$

Donde  $\theta$  es el ángulo del antebrazo en dirección vertical (se relaciona con elevation) y  $\psi$  es el ángulo del antebrazo en dirección horizontal (se relaciona con travel). El funcionamiento implementado consiste en aumentar el valor del setpoint de elevation cuando  $\theta$  sea mayor a  $20^\circ$  y disminuirlo cuando sea menor a  $-20^\circ$ . Esto le da un rango de activación para evitar movimientos indeseados. Para el caso de travel el ángulo  $\psi$  se mapea directamente sobre el valor de  $\rho$ . Las variables  $q_1$  a  $q_4$  son los 4 cuaterniones que poseen la información de orientación del sensor Myo. El funcionamiento del modo IMU se ilustra en la Figura 2.64.



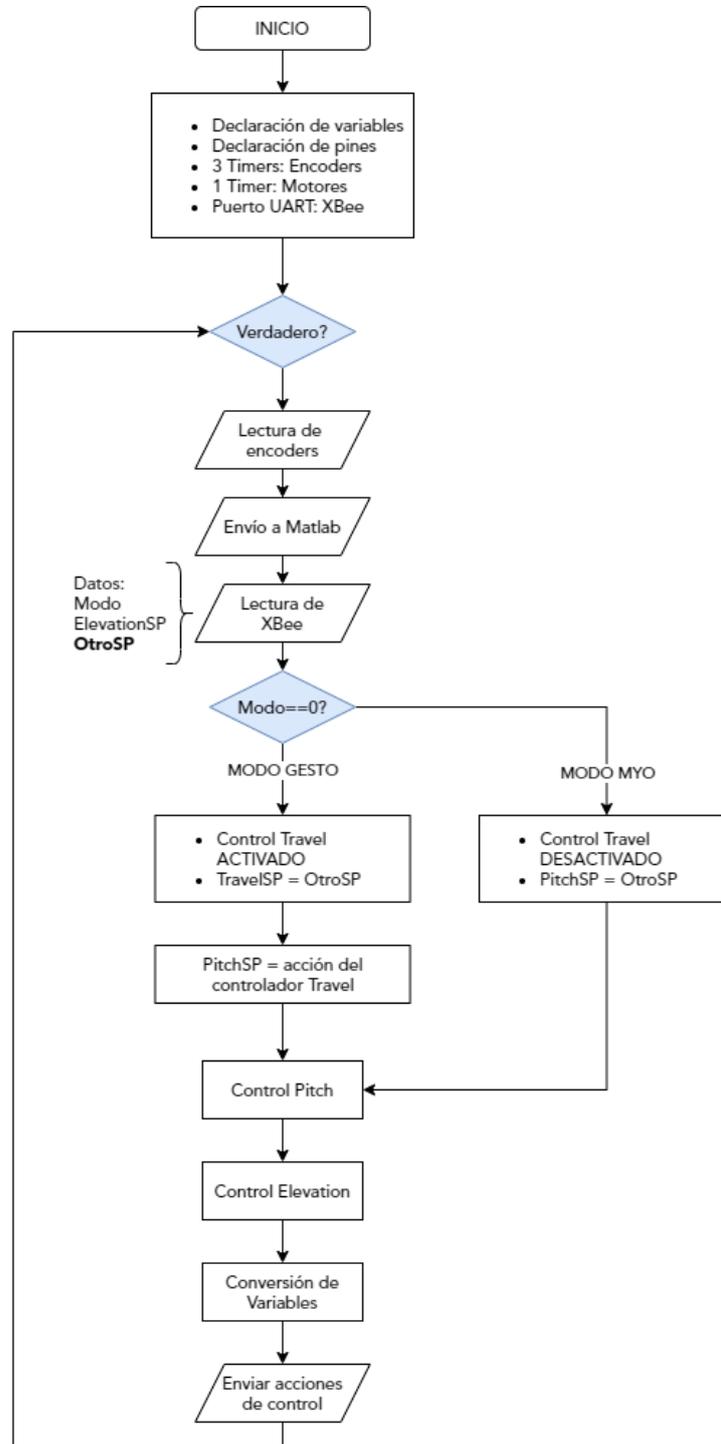
**Figura 2.64.** Esquema del funcionamiento del modo IMU.

La implementación del control multimodo se ha realizado en el código de la interfaz gráfica y la lógica de funcionamiento se muestra en el diagrama de flujo de la Figura 2.65. Como se puede observar, en este diagrama solo se generan los valores de setpoint, los cuales son enviados al microcontrolador. Una vez aquí ya se realiza todo el sistema de control en las diferentes variables de acuerdo con el setpoint generado.



**Figura 2.65.** Diagrama de flujo de la implementación de la característica de comandos multimodo.

Para la utilización de este método de control se ha implementado el diagrama de flujo de la Figura 2.66 en el microcontrolador. Como se puede observar en esta figura, no se utilizan variables adicionales para implementar el modo manual ya que internamente en el programa de Matlab se envía el valor de Modo 0 o 1 para controlar Travel o Pitch, respectivamente.



**Figura 2.66.** Diagrama de Flujo del microcontrolador Pyboard.

### **3. RESULTADOS Y DISCUSIÓN**

Con base en la metodología descrita en el capítulo anterior se ha elaborado el prototipo del helicóptero de 3-GDL, con el cual se realizaron pruebas del control de movimiento utilizando el sistema de reconocimiento de gestos. En las siguientes secciones se muestran imágenes del prototipo construido junto con la interfaz diseñada y los resultados de diferentes pruebas para la verificación del funcionamiento en los distintos modos de operación.

Para empezar, se presentan imágenes del prototipo construido y detalles de sus partes, así como un breve análisis de costos. Posteriormente se muestran las pruebas de los controladores diseñados con las características de las respuestas ante entradas paso para cada una de las variables. Para la variable travel se analiza el comportamiento del controlador PID normal comparado con las modificaciones descritas en el capítulo anterior.

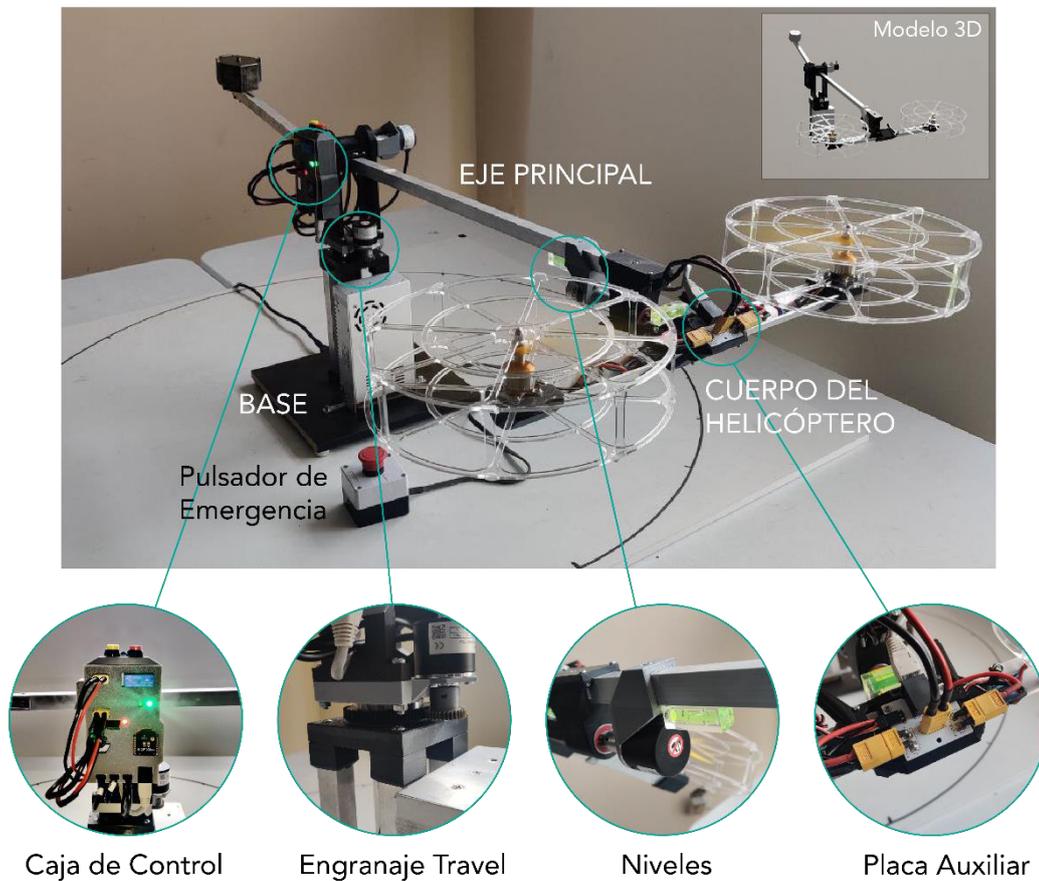
Finalmente se comprueba el funcionamiento del helicóptero de 3-GDL enlazado con el sistema de reconocimiento de gestos mediante la interfaz diseñada. Para esta parte se presenta la interfaz y su utilización para comandar la plataforma. En base a las pruebas realizadas con la interfaz se demuestra la funcionalidad del sistema de HGR para comandar el vuelo del helicóptero en los modos de gesto e IMU.

#### **3.1. PROTOTIPO CONSTRUIDO**

En la Figura 3.1 se presenta el prototipo del helicóptero de 3-GDL construido con cada una de sus partes, de acuerdo con lo explicado en la metodología. En la caja de control se ubica la PCB y los elementos correspondientes. El relé de 30A se encuentra por detrás de la placa, por lo que no se puede observar desde el frente. El fusible seleccionado es automotriz tipo BLADE verde (30A) debido a su tamaño reducido y su forma permite colocarlo en la parte lateral de la caja. En el cuerpo del helicóptero se tiene la placa auxiliar, en la cual se han soldado varillas de 2mm en ángulo recto a los conectores XT60 para poder alimentar a los ESCs de forma lateral.

Adicionalmente se han incluido niveles en el brazo de soporte y sobre el cuerpo del helicóptero. Estos permiten determinar la posición de origen de la plataforma ya que, como se mencionó en el anterior capítulo, se considera 0° para elevation y pitch cuando el brazo y el cuerpo se encuentran de forma completamente horizontal. Para fijar el punto de origen se sostiene al helicóptero manteniendo a las burbujas de los niveles en el centro y se enciende al sistema girando el botón de emergencia (por primera vez). En caso de ya tener el sistema encendido se presiona el botón de reset de la placa Pyboard, para lo cual se tiene una abertura en la tapa de la caja de control, mediante la cual se puede acceder a

dicho botón. Es importante tomar en cuenta que la alimentación de la placa Pyboard se toma desde el USB cuando se encuentra conectado a la PC, por esa razón se tiene el jumper de 3 pines que direcciona el voltaje desde la fuente cuando no se utilice el USB, tal como se explicó en la sección 2.2.1.4.1 de la metodología. El LED de color rojo indica si se encuentra conectado el jumper para evitar que se utilice el USB.



**Figura 3.1.** Prototipo construido y algunas de sus partes.

### 3.1.1. RESUMEN DE COSTOS

En la Tabla 3.1 se presenta un resumen de los costos de los componentes utilizados para la elaboración del prototipo del helicóptero de 3-GDL. Como se puede observar, los elementos eléctricos y electrónicos del sistema son los que mayor costo representan, sin embargo, son elementos fácilmente disponibles comercialmente, de igual manera las partes de aluminio poseen medidas estándar y las piezas de plástico se pueden elaborar fácilmente mediante impresión 3D o corte láser. Cabe mencionar que en este resumen de costos no se considera el correspondiente a diseño (ingeniería) ni el tiempo empleado en la elaboración del helicóptero, no obstante, como se observa el total de los materiales se

encuentra apenas por encima de \$500. En el Anexo E se puede encontrar una versión más detallada de la Tabla 3.1.

**Tabla 3.1.** Lista de costos de los componentes del prototipo construido.

	Elementos	Cantidad	Costo unitario	Costo total
Componentes eléctricos/ electrónicos	Motores Brushless 1000KV	2	\$ 15.99	\$ 31.98
	ESC 30A	2	\$ 16.49	\$ 32.98
	Helices (paquete de 4 pares)	1	\$ 12.99	\$ 12.99
	Anillo deslizante 30A	1	\$ 29.89	\$ 29.89
	Pyboard	1	\$ 44.95	\$ 44.95
	Encoders 1000PPR	3	\$ 18.99	\$ 56.97
	Fuente 12V / 30A	1	\$ 18.95	\$ 18.95
	XBee 3 antena PCB	2	\$ 17.95	\$ 35.90
	Adaptador XBee USB	1	\$ 25.95	\$ 25.95
	Adaptador regulado XBee 4 pines	1	\$ 10.95	\$ 10.95
	Botón de emergencia y caja	1	\$ 5.00	\$ 5.00
	Otros	1	\$ 26.62	\$ 27.22
	PCB	Placa principal (5 unidades)	1	\$ 23.14
Placa auxiliar (5 unidades)		1	\$ 6.26	\$ 6.26
3D	Impresiones 3D (todas)	1	\$ 49.00	\$ 49.00
Acrílicos	Protector inferior	2	\$ 6.50	\$ 13.00
	Protector superior	2	\$ 6.50	\$ 13.00
	Columna	16	\$ 0.35	\$ 5.60
Metales	Perfil "L" de aluminio	1	\$ 1.50	\$ 1.50
	Tubos de aluminio	1	\$ 2.05	\$ 2.05
	Contrapeso	1	\$ 6.10	\$ 6.10
	Otros	1	\$ 29.50	\$ 29.50
	Tornillos (todos)	1	\$ 7.39	\$ 7.39
	Otros componentes	1	\$ 11.00	\$ 11.00
			<b>TOTAL</b>	<b>\$ 501.27</b>

Las placas PCB fueron diseñadas en Autodesk Eagle y elaboradas en JLCPCB que es una fábrica de placas de circuito impreso en China [58]. El costo de fabricación fue de menos de \$10 para 5 placas de cada tipo (principal y auxiliar), sin embargo, se ha contemplado el valor del envío que se encuentra alrededor de los \$20. El resto de los elementos como tornillos, piezas metálicas, acrílicos e impresiones 3D se pueden conseguir dentro del país por los precios que se indican en la Tabla 3.1.

Para comparar el costo del prototipo construido con las versiones comerciales se ha encontrado en [59] que el experimento de helicóptero de 3 grados de libertad de la marca

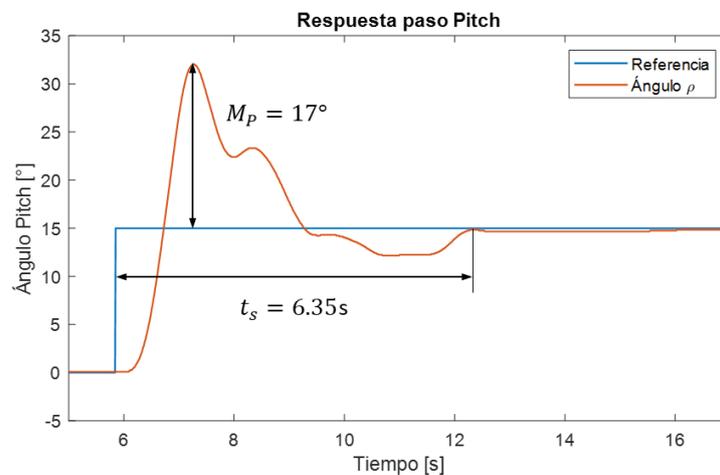
Quanser tiene un valor aproximado de \$13,029.75. Aunque no se han considerado los costos de diseño y fabricación, el valor del prototipo elaborado es una fracción muy reducida con respecto a este modelo comercial. Con esto se puede evidenciar el beneficio que representa el uso de impresión 3D y materiales accesibles para la fabricación de este tipo de productos. Un valor completo considerando el estimado del costo de ingeniería se puede encontrar en el Anexo E.

## 3.2. PRUEBAS DE CONTROLADORES

A continuación, se mostrará la respuesta de los controladores diseñados para cada una de las variables del helicóptero de 3-GDL. Se mostrará la respuesta del lazo de control para las variables pitch, elevation y travel ante cambios de referencia de tipo paso y la reacción del sistema ante perturbaciones externas.

### 3.2.1. CONTROLADOR PITCH

Para realizar pruebas en el controlador de esta variable se ha mantenido fijo al brazo de soporte de forma horizontal de modo que el único ángulo que se pueda modificar sea pitch,  $\rho$ . El funcionamiento del controlador se comprueba mediante 3 pruebas: en primer lugar, se presenta un cambio de referencia de 0 a 15° para observar las características en estado transitorio del sistema (Figura 3.2). Luego se hace un cambio de referencia de -20° a 20° que es el rango máximo admitido para el valor de referencia y se observa el valor del sobreimpulso (Figura 3.3). Finalmente se observa las características del rechazo a perturbaciones del controlador, para lo cual se realizan perturbaciones de tipo impulso (perturbaciones cortas con la mano) para diferentes valores de referencia (Figuras 3.4 y 3.5).

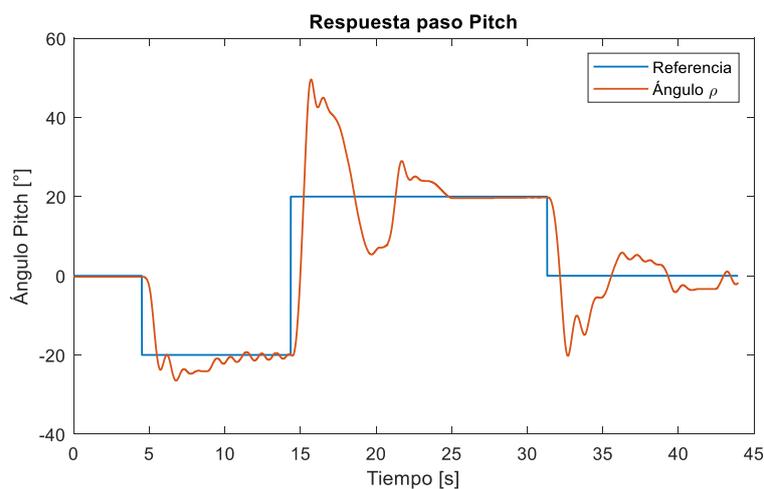


**Figura 3.2.** Respuesta paso de la variable pitch ante un cambio de 15°.

### 3.2.1.1. Respuesta transitoria

En la Figura 3.2 se observa la respuesta paso a un cambio de referencia de  $15^\circ$ . Como se puede observar, existe un sobreimpulso considerable con un tiempo de estabilización relativamente corto. Esta característica es favorable ya que este sobreimpulso va a generar una respuesta más rápida en el lazo de control externo de la variable travel.

Para la siguiente prueba se empezó con un cambio de referencia de  $-20^\circ$  seguido de un cambio hasta  $20^\circ$  (es decir de  $40^\circ$ ) y se obtuvo la respuesta que se indica en la Figura 3.3. Internamente en el código del microcontrolador se tiene saturada la referencia de pitch para encontrarse dentro del rango de  $-20^\circ$  a  $20^\circ$  ya que como se puede observar, dentro de este rango se puede alcanzar los límites mecánicos de  $-50^\circ$  a  $50^\circ$  debido a los sobreimpulsos.

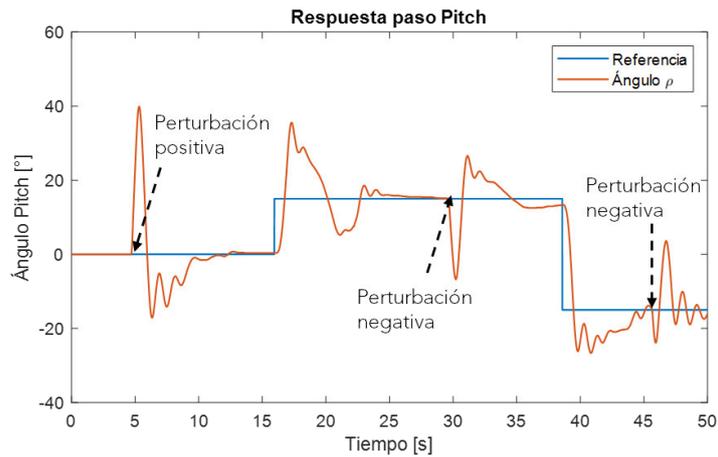


**Figura 3.3.** Respuesta de la variable pitch ante cambios de referencia de  $-20^\circ$  a  $20^\circ$ .

En la gráfica de la Figura 3.3 se puede notar que en el primer cambio de referencia a  $-20^\circ$  se encuentra un exceso muy pequeño de apenas  $6.5^\circ$ , mientras que, en el segundo se tiene un aumento mucho mayor debido a que el cambio de referencia es el doble del primero.

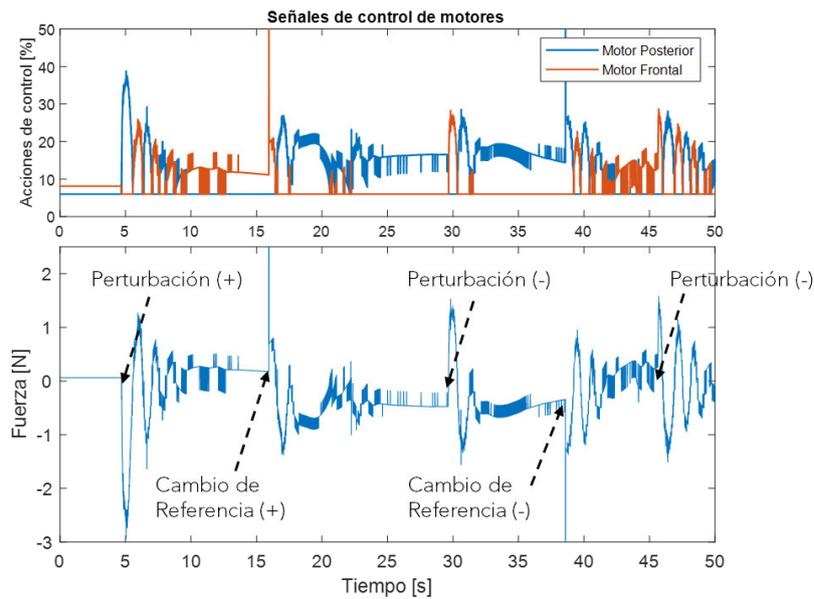
### 3.2.1.2. Rechazo a perturbaciones

Para la prueba final se comprueba la capacidad de rechazo a perturbaciones del controlador pitch. En este caso se establecieron cambios de referencia de  $-15^\circ$  y  $15^\circ$  y se realizaron ligeras perturbaciones cortas con la mano en los extremos del cuerpo del helicóptero que representan perturbaciones de tipo impulso. Los resultados se muestran en la Figura 3.4.



**Figura 3.4.** Respuesta ante perturbaciones de tipo impulso en la variable pitch.

Para esta prueba también se presentan las acciones de control en la Figura 3.5. En la gráfica de la parte superior se observan las señales de control enviadas a cada motor en forma de porcentaje, mientras que en la parte inferior se muestra la fuerza aplicada en el eje pitch, es decir  $F_{dif}$ . En esta gráfica se muestran las acciones de control al realizar tanto cambios de referencia como perturbaciones, por lo que se tiene un comportamiento similar para la prueba anterior que estaba enfocada a cambios de referencia.



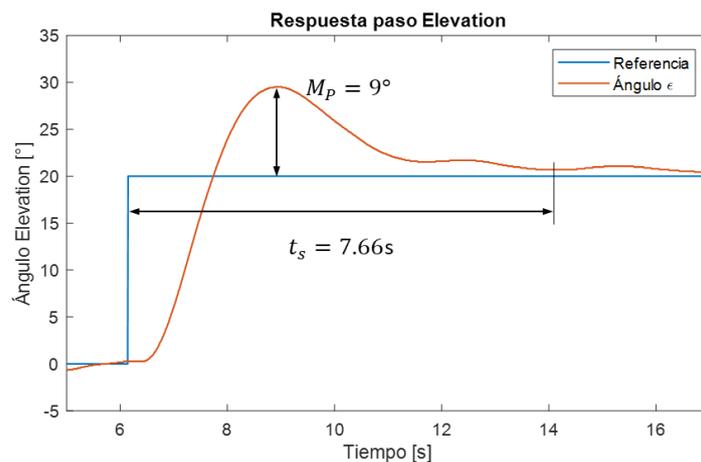
**Figura 3.5.** Acciones de control: señal en porcentaje (arriba) y fuerza  $F_{dif}$  en Newtons (abajo).

Para la gráfica de fuerza que se muestra en la parte inferior de la Figura 3.5 se ha empleado la Ecuación (2.5) que permite calcular  $F_f$  y  $F_b$  y posteriormente, el valor de  $F_{dif}$  se determina restando ambas.

### 3.2.2. CONTROLADOR ELEVATION

A continuación, se muestra el funcionamiento del controlador de la variable elevation. Para las pruebas realizadas se mantiene activo al controlador de la variable pitch de modo que se pueda sostener al cuerpo del helicóptero de forma horizontal. En primer lugar, se muestra la respuesta del sistema ante cambios de referencia de  $20^\circ$  y se obtienen las características transitorias (Figuras 3.6 y 3.7). Luego se realizan perturbaciones y se observa el comportamiento del sistema (Figuras 3.8 y 3.9).

#### 3.2.2.1. Respuesta transitoria

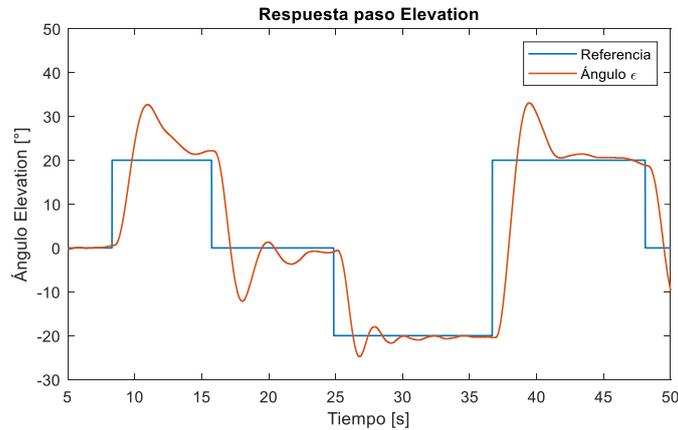


**Figura 3.6.** Respuesta paso de la variable elevation para un cambio de referencia de  $20^\circ$ .

Como se puede observar en la Figura 3.6 la respuesta es más suave que para la variable pitch. Esto se debe a que, en este caso, el sistema posee más masa (por lo tanto más inercia), por lo cual las aceleraciones serán menores, lo que hace que la dinámica del sistema sea más lenta que para la variable pitch lo cual a su vez da lugar a una respuesta más suave.

En la Figura 3.7 se muestra la respuesta del sistema luego de realizar cambios de referencia positivos y negativos. En este caso se puede notar que los cambios de referencia negativos presentan un comportamiento ligeramente más oscilatorio, pero con menor sobreimpulso y en general una respuesta más rápida. Esto se debe a que para hacer cambios de referencia negativos los motores deben reducir su velocidad de giro, con lo cual la única fuerza que genera la aceleración negativa es la de la gravedad, por lo que puede moverse rápidamente. Por otro lado, para generar movimientos hacia arriba, se debe generar una fuerza mayor ya que se debe contrarrestar el peso del prototipo. Esto se puede observar especialmente para el cambio de  $0^\circ$  a  $-20^\circ$ , donde se tiene un sobreimpulso

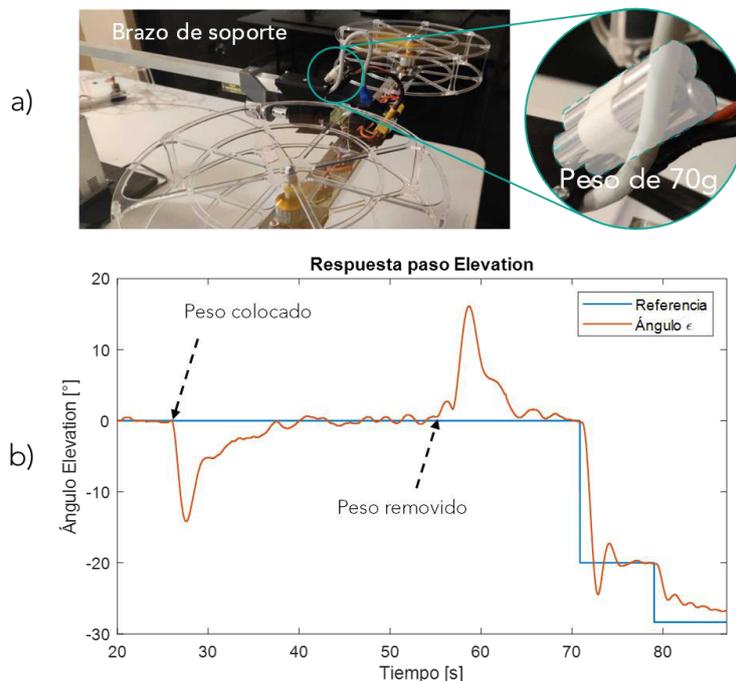
de menos de  $5^\circ$  lo cual denota claramente la diferencia entre el movimiento hacia arriba y abajo. En esta gráfica se muestra además un cambio de referencia considerable de  $-20^\circ$  a  $20^\circ$  con un sobreimpulso de apenas  $12^\circ$ .



**Figura 3.7.** Respuesta ante cambios de referencia positivos y negativos para elevation.

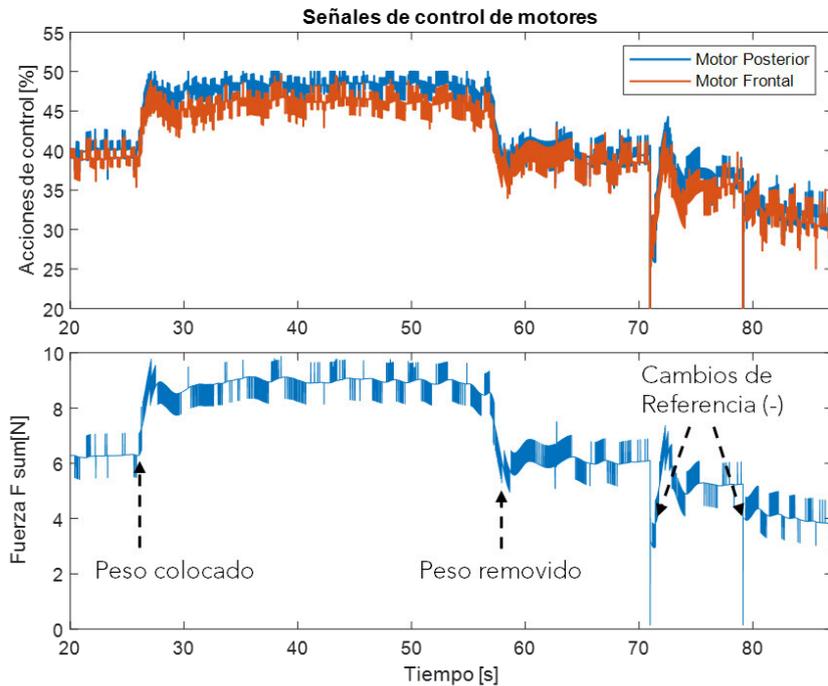
### 3.2.2.2. Rechazo a perturbaciones

Para la siguiente prueba se han realizado perturbaciones sobre el brazo de soporte del prototipo, de modo que se observe el efecto sobre la variable elevation. En este caso se ha colocado un peso de 70g en el extremo del brazo de soporte como se indica en la Figura 3.8 a). Los resultados de la prueba realizada se muestran en la gráfica de la Figura 3.8 b).



**Figura 3.8.** Respuesta a perturbaciones en elevation. a) Colocación del peso. b) Gráficas de la respuesta.

Finalmente, en la Figura 3.9 se observan las acciones de control para la prueba de rechazo de perturbaciones descrita en el párrafo anterior. Como se puede observar en esta gráfica, la señal de control en porcentaje aumenta notablemente al aplicar el peso, con lo cual la variable se regula a su posición de referencia.



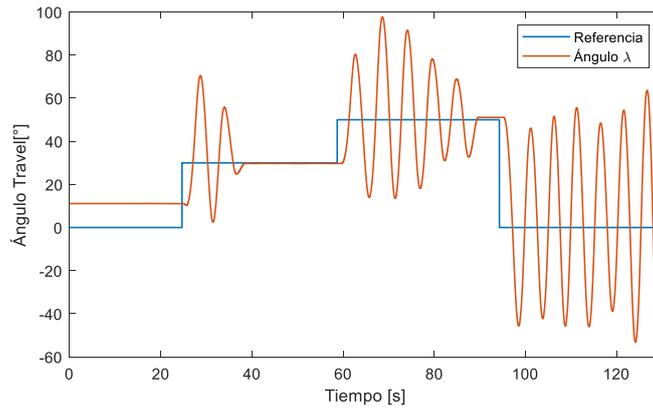
**Figura 3.9.** Acciones de control: señal en porcentaje (arriba) y fuerza  $F_{sum}$  en Newtons (abajo).

### 3.3.3. CONTROLADOR TRAVEL

En el caso de la variable travel, la respuesta del sistema ante entradas paso de  $30^\circ$ ,  $20^\circ$  y  $-50^\circ$  se indica en la Figura 3.10. Como se puede observar, el controlador PD diseñado es muy inestable y dependiendo del cambio de referencia puede o no llegar a estabilizarse. Por ejemplo, en el primer cambio logra llegar sin muchas oscilaciones al valor de referencia, sin embargo, para el segundo y tercero se desestabiliza totalmente. Esto se debe a que la ganancia proporcional obtenida en la sintonización con la herramienta PID Tool de Matlab es demasiado alta.

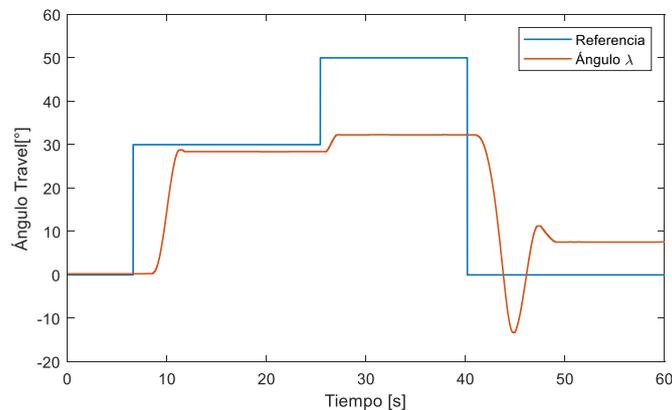
Tomando esto en cuenta se han reducido los valores de las ganancias al 60% del valor original, de modo que se obtiene la función del controlador PD (de forma paralela) que se muestra en la Ecuación (3.2).

$$C_\lambda(s) = 0.1926 + 0.3368s \quad (3.2)$$



**Figura 3.10.** Respuesta del controlador Travel diseñado. La respuesta es inestable porque las ganancias son muy altas.

Al poner a prueba el controlador de la Ecuación (3.2) se encuentra la respuesta que se observa en la Figura 3.11. Como se puede observar, el controlador PD diseñado presenta características transitorias satisfactorias, ya que el sistema se logra estabilizar en un tiempo menor a 6s y con poco o cero sobreimpulso. Sin embargo, como se puede notar, especialmente para cambios de referencia pequeños ( $\sim 20^\circ$ ) se presenta un error considerable en el estado estable. Como se explicó en el capítulo anterior, en la sección 2.3.2.3, dado que la estructura del sistema es similar a un doble integrador, el controlador de tipo PD debería ser suficiente para llegar a los valores de referencia deseados. No obstante, este modelo no considera la presencia del rozamiento en el rodamiento de la base travel, el cual es causante del error que se observa en la Figura 3.11.



**Figura 3.11.** Respuesta paso de la variable travel para varios cambios de referencia.

Para solucionar el problema del error en estado estable se añade un término integral al controlador de la variable Travel, lo cual, sin embargo, generó complicaciones adicionales que se solucionaron incluyendo un algoritmo de anti wind-up y un reset en la integral. La

implementación de estas mejoras en el controlador de travel se describe en los siguientes párrafos.

### 3.3.3.1. Controlador PID de Travel sintonizado mediante el método de Ziegler-Nichols

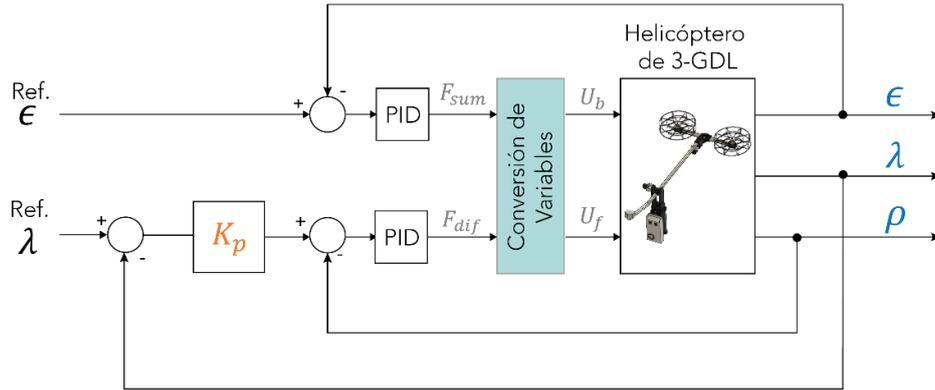
El modelo de función de transferencia mostrado en el capítulo anterior presenta algunas limitaciones como la que se acaba de mencionar ya que no se considera el rozamiento existente entre las piezas y el valor exacto de la fuerza  $F_{sum}$ . Por este motivo la sintonización mediante la herramienta PID Tool de Matlab resulta insuficiente para el control de esta variable, por lo cual se ha utilizado el método de Ziegler-Nichols en lazo cerrado ya que con esto no afecta la exactitud del modelo debido a que es una sintonización experimental.

El método de Ziegler-Nichols es un procedimiento heurístico para la sintonización de controladores PID. Fue desarrollado por John Ziegler un Nathaniel Nichols, de quienes recibe su nombre [54]. Este método consiste en fijar las constantes integral y derivativa en cero, de modo que se tenga un controlador en lazo cerrado únicamente de tipo proporcional. La ganancia  $K_p$  de este controlador se aumenta progresivamente desde cero hasta llegar a la denominada ganancia crítica  $K_c$ , en la cual el sistema presenta oscilaciones sostenidas y regulares de periodo  $T_c$ . Con base en estos parámetros se utilizan las ecuaciones de sintonización que se presentan en la Tabla 3.2. [54].

**Tabla 3.2.** Ecuaciones de sintonización para el método de Ziegler-Nichols [54].

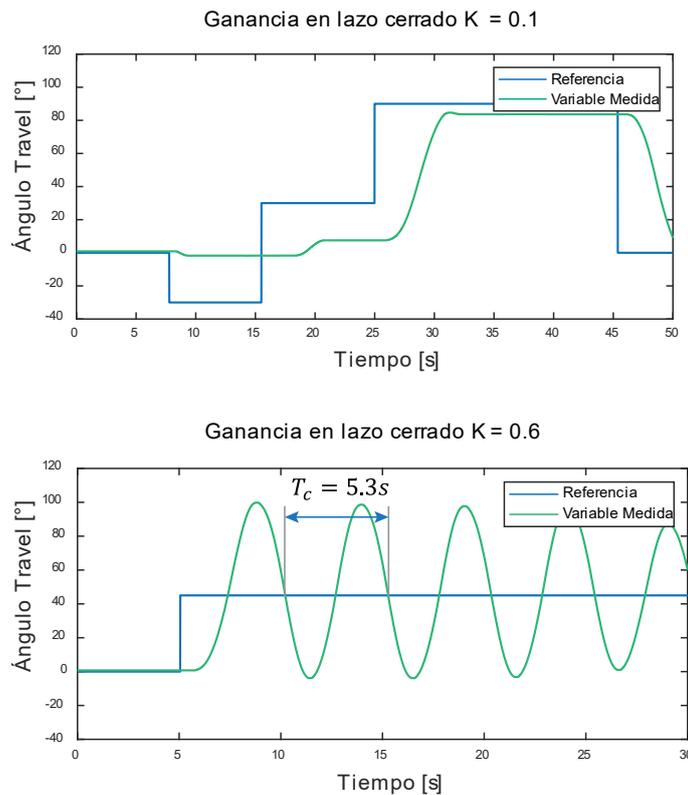
Tipo de Controlador	$K_p$	$K_i$	$K_d$
P	$0.5K_c$	–	–
PI	$0.45K_c$	$0.54K_c/T_c$	–
PD	$0.8K_c$	–	$0.1K_cT_c$
PID	$0.6K_c$	$1.2K_c/T_c$	$0.075K_cT_c$
Poco sobreimpulso	$\frac{1}{3}K_c$	$\frac{2}{3}K_c/T_c$	$\frac{1}{9}K_cT_c$

El esquema utilizado para este procedimiento se muestra en la Figura 3.12. Como se puede observar, las variables elevation y pitch poseen sus propios sistemas de control como se ha mostrado anteriormente. Esto es importante ya que con el controlador de elevation se mantiene horizontal al brazo de soporte y el controlador de pitch sirve como lazo interno de control de la variable travel, sobre la cual se realiza todo este procedimiento.



**Figura 3.12.** Sintonización del controlador Travel mediante Ziegler-Nichols. La constante  $K_p$  se aumenta hasta hacer oscilar el sistema. Las variables  $\epsilon$  y  $\rho$  poseen su propio sistema de control funcional.

De acuerdo con lo explicado en los párrafos anteriores, se ha aumentado paulatinamente el valor de  $K_p$  desde 0.1 hasta que se presentaron las oscilaciones regulares en un valor de  $K_p = K_c = 0.6$ . Como se indica en la Figura 3.13, al generar un cambio de setpoint de  $45^\circ$  se producen las oscilaciones con las cuales se puede determinar el valor del periodo  $T_c = 5.3s$ .

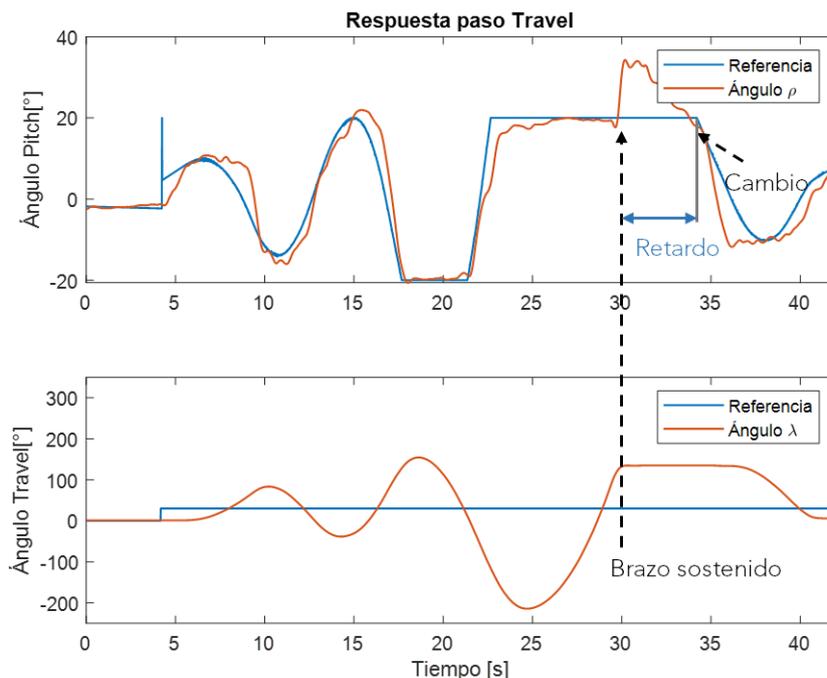


**Figura 3.13.** Respuesta del sistema travel para el valor mínimo de  $K_p=0.1$  (arriba) y para el valor crítico  $K_p = K_c = 0.7$  (abajo).

Con base en estos dos parámetros se escogen las ecuaciones de sintonización de la Tabla 3.2 correspondiente al controlador PID de *poco sobreimpulso*. Se escoge este tipo de controlador debido a que el sistema acelera rápidamente de forma natural ya que tiene una estructura parecida a un doble integrador. Tomando esto en cuenta el controlador diseñado se presenta en la Ecuación (3.3)

$$C_{\lambda}(s) = 0.2 + 0.0754 \frac{1}{s} + 0.3533s \quad (3.3)$$

En la Figura 3.14 se presenta la respuesta de la variable travel ante un cambio de referencia de 30° con el nuevo controlador PID diseñado. En la parte superior también se muestra la variable pitch y la referencia generada por el controlador, ya que es la variable interna del lazo. Como se puede observar, esta respuesta es completamente inestable, esto se debe a que la acción integral se acumula demasiado y produce un resultado incorrecto para el controlador. Esto se evidencia claramente alrededor de los 30s ya que en ese punto se mantuvo sostenido el brazo de soporte y sin embargo la acción de control empieza a disminuir después de casi 5s como se muestra en la gráfica de pitch de la Figura 3.14. Para solucionar este error se debe implementar un algoritmo de anti wind-up [60], con el cual se evita que la acción integral se acumule demasiado y desestabilice el sistema.



**Figura 3.14.** Respuesta paso del controlador PID travel ante un cambio de referencia de 30°. La respuesta es inestable debido a que no tiene anti wind-up en la acción integral.

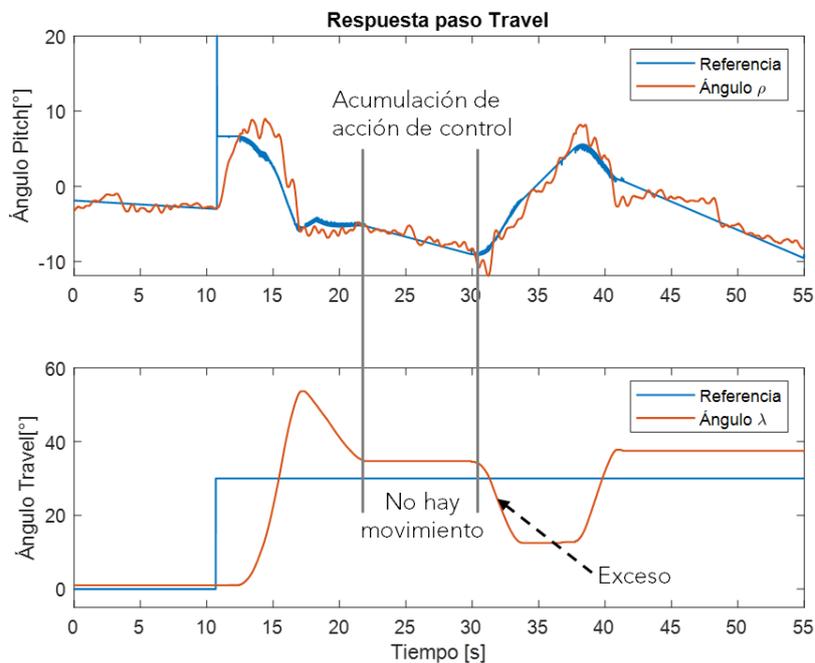
Cabe mencionar que la respuesta del controlador PD anterior parece presentar mejores resultados, sin embargo, la falta de una acción integral es la causa de que no se logre

conseguir la referencia deseada. En la sección 3.3.3.2 se muestran las adecuaciones que se realizaron en el controlador travel para conseguir una respuesta satisfactoria.

### 3.3.3.2. Modificaciones al controlador Travel: algoritmo de anti wind-up y reset

Para solucionar el problema de la acumulación, se utiliza un algoritmo denominado “anti-windup”. En este caso, la forma en que se implementó esta característica es mediante una estructura dual PD+PID de modo que cuando el error sea grande, el controlador sea de tipo PD y la acción integral se mantenga en cero y únicamente cuando se encuentre dentro de cierto rango se active la integral de modo que pueda llegar a la referencia deseada. El umbral considerado para la activación de la acción integral es  $-20^\circ < e_\lambda < 20^\circ$  ya que solo con el controlador PD el sistema tiene un error máximo de  $20^\circ$  de modo que, dentro de ese rango ya puede empezar a actuar la acción integral y reducir el error a  $0^\circ$ .

En la Figura 3.15 se muestra el comportamiento del sistema luego de implementar este esquema PD+PID.

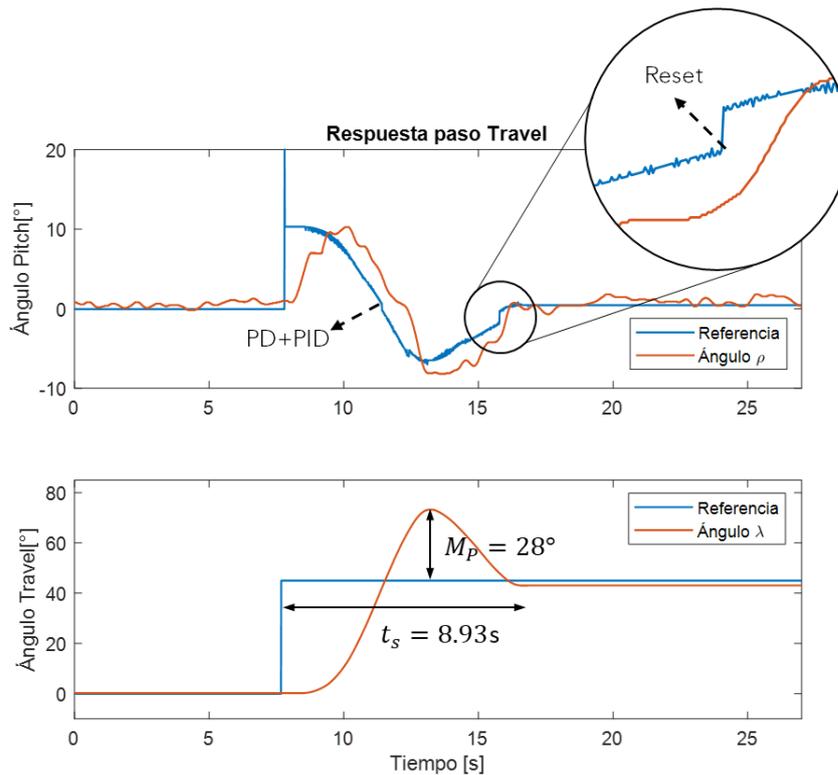


**Figura 3.15.** Respuesta luego de aplicar la estructura PD+PID como algoritmo de anti wind-up.

Como se puede observar en la Figura 3.15 la estructura PD+PID soluciona la desestabilización del sistema, sin embargo, presenta un comportamiento errático como se nota alrededor de los 35s.

Esto se debe a que la base de travel posee un rozamiento que se manifiesta como una zona muerta en el controlador, es decir, la acción de control (la variable pitch) aumenta, pero travel permanece sin cambios hasta después de un tiempo, tal como se indica entre los 20s y 30s de la Figura 3.15. La acción integral se acumula debido al error, pero ya que travel no presenta cambios se acumula demasiado y se pasa de la referencia hacia el otro lado. Este comportamiento se repite y en definitiva el sistema no logra alcanzar la referencia de forma satisfactoria. Para solucionar esto se ha implementado un reset en la acción integral con un rango de  $-3^\circ < e_\lambda < 3^\circ$  en el cual se anula la acumulación y se mantiene estable a la salida del sistema. Este rango de  $3^\circ$  se ha determinado en función del mismo comportamiento de la plataforma; si se escoge un valor menor el tiempo de establecimiento puede hacerse muy largo debido al movimiento irregular causado por el rozamiento explicado anteriormente. Y, por otro lado, si es un intervalo muy amplio, el máximo error posible se incrementa también, por lo que se ha determinado este valor de  $3^\circ$ .

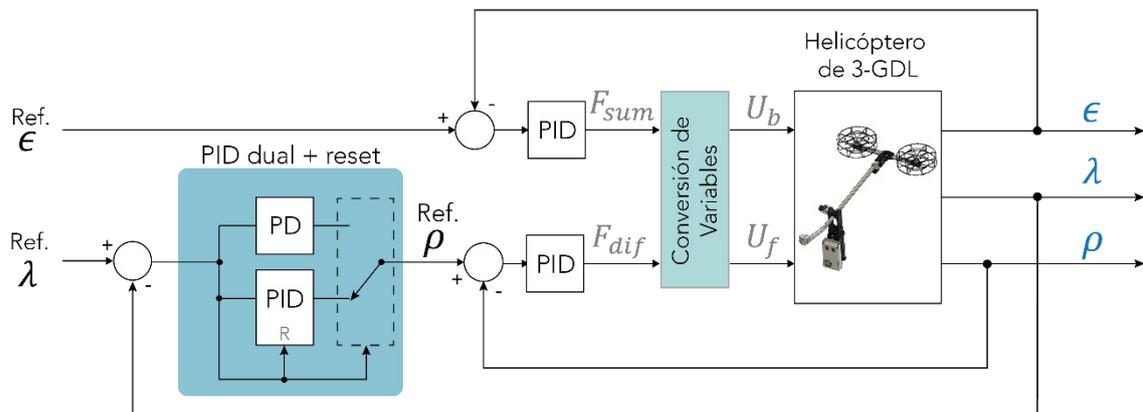
En la Figura 3.16 se indica el comportamiento del sistema luego de implementar esta nueva modificación. Como se puede observar, esta respuesta es mucho más estable y logra llegar a la referencia en un tiempo corto. Se puede notar un ligero error en estado estable, sin embargo, corresponde a los  $3^\circ$  del umbral establecido para el reset y es un valor admisible.



**Figura 3.16.** Respuesta del controlador travel ante un cambio de referencia de  $45^\circ$  con la estructura PD+PID+reset.

Adicionalmente, en la Figura 3.16 se pueden observar otras características importantes de la respuesta paso del sistema travel. En primer lugar, en la gráfica inferior se nota un retardo desde que se produce el cambio de referencia hasta que el sistema empieza a moverse; este retardo se debe al rozamiento que existe en esta variable como se explicó en los párrafos anteriores. La acción de control (ángulo pitch) empieza a aumentar antes de que exista movimiento, lo cual se manifiesta como un retardo en la variable. Por otro lado, en la acción de control (curva azul de la gráfica superior de la Figura 3.16) se nota un pico de muy corta duración en el momento en que se realiza el cambio de referencia; este pico se genera debido a la acción derivativa del controlador. Tomando en cuenta que se obtiene la derivada de una entrada paso y, dado que es un cambio repentino (entrada paso), la acción de control aumenta considerablemente, aunque solo de manera instantánea.

Tomando en cuenta todo lo mencionado hasta ahora, la estructura completa del sistema de control final se presenta en la Figura 3.17.

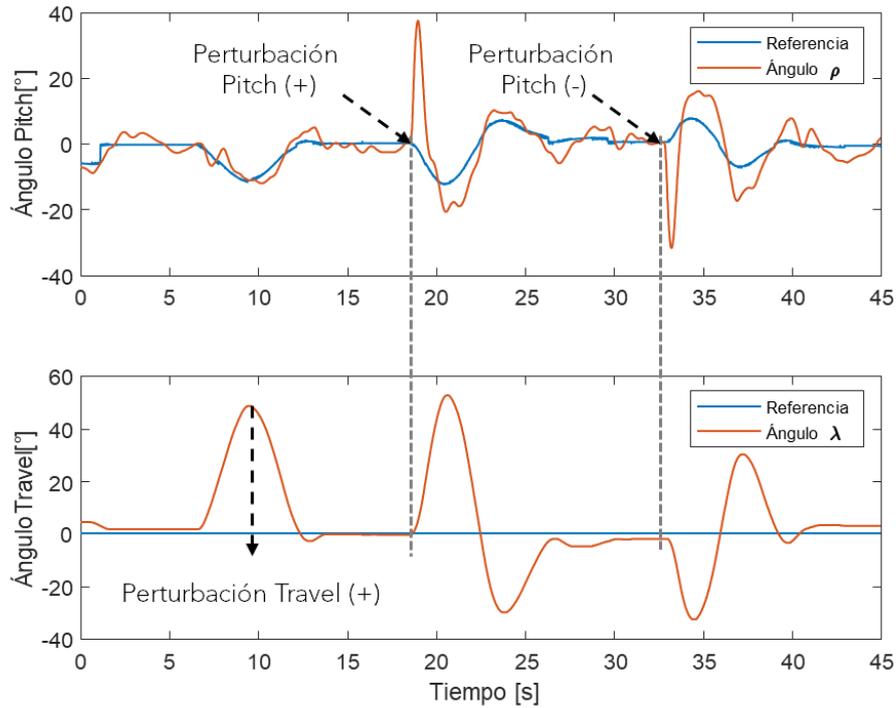


**Figura 3.17.** Diagrama del sistema de control para el helicóptero de 3-GDL.

### 3.3.3.3. Rechazo a perturbaciones

Como último punto se analizará el comportamiento del sistema ante perturbaciones externas. Para esta prueba se realizaron 2 tipos de perturbaciones: directamente sobre la variable travel y otras sobre el ángulo pitch (ya que se encuentran acopladas). En el primer caso se colocó en la referencia de  $0^\circ$  y se movió el brazo de soporte hasta alrededor de  $45^\circ$  para analizar el comportamiento del controlador para regularse nuevamente a  $0^\circ$ . En el segundo caso se realiza una perturbación impulsiva (ligera perturbación corta con la mano) en la variable pitch y se verifica el efecto que tiene sobre travel. La respuesta se muestra en la Figura 3.17. Cabe recalcar que las perturbaciones en el eje pitch afectan al

estado de la variable travel ya que, como es la variable interna del lazo de control en cascada, ambas se encuentran acopladas entre sí.



**Figura 3.18.** Rechazo a perturbaciones aplicadas sobre travel y pitch.

Como se observa, el sistema tiene la capacidad de regresar a la referencia de 0° en alrededor de 7s para todos los casos, lo cual es un tiempo adecuado para la aplicación que se está realizando.

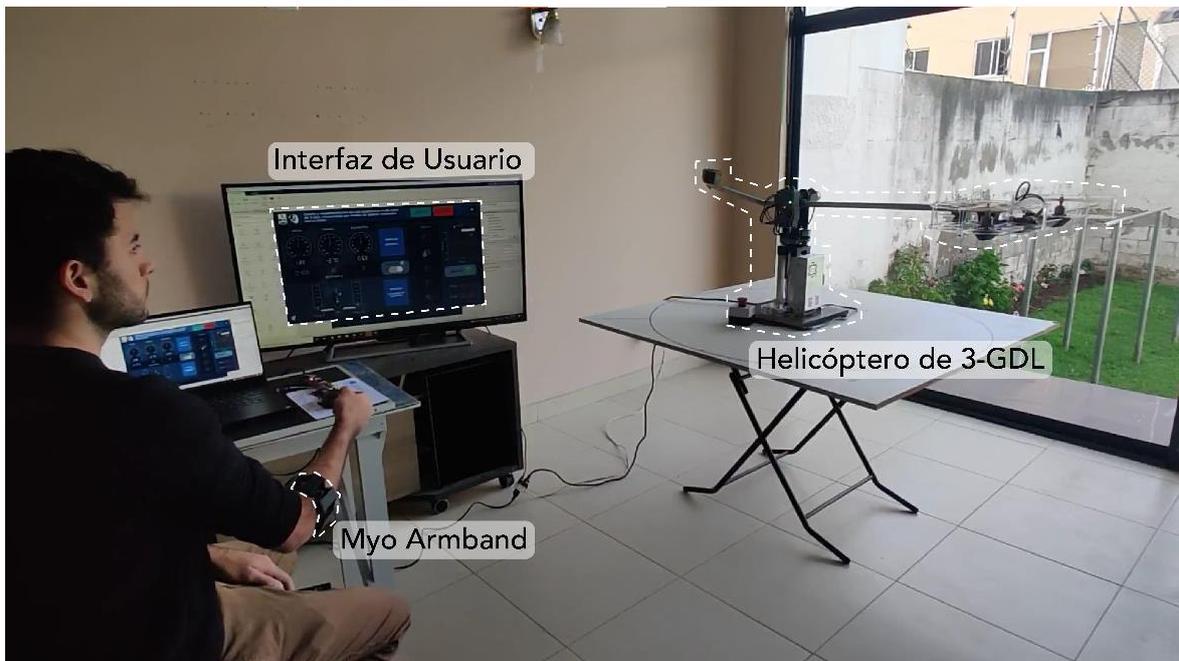
### 3.3. PRUEBAS DE COMANDOS MULTIMODO

Finalmente se mostrará la integración de todos los elementos descritos en la metodología y el funcionamiento del método de control multimodo para modificar el movimiento del helicóptero de 3-GDL. En primer lugar, en la Sección 3.3.1 se muestran los resultados del modo Myo, el cual a su vez incluye los modos Gesto e IMU. Para esta primera parte se analiza el movimiento de la plataforma luego de realizar cada uno de los cinco gestos descritos en la Sección 2.4 de la metodología. Luego se muestra el movimiento en modo IMU para lo cual se comparan las variables del helicóptero (travel, elevation y pitch) con la información obtenida por la IMU del sensor Myo armband durante la ejecución de la interfaz.

Por último, en la Sección 3.3.2 se muestran los resultados empleando el modo Manual. De forma similar al caso anterior, se analiza las diferentes variables del helicóptero y también se prueba el control de la variable travel y pitch de forma independiente.

### 3.3.1. MODO MYO

En la Figura 3.19 se pueden observar todos los elementos necesarios para la implementación del control del helicóptero de 3-GDL mediante gestos. Estos elementos son los que se describieron a lo largo del capítulo de metodología.



**Figura 3.19.** Elementos del sistema para control del helicóptero de 3-GDL mediante gestos.

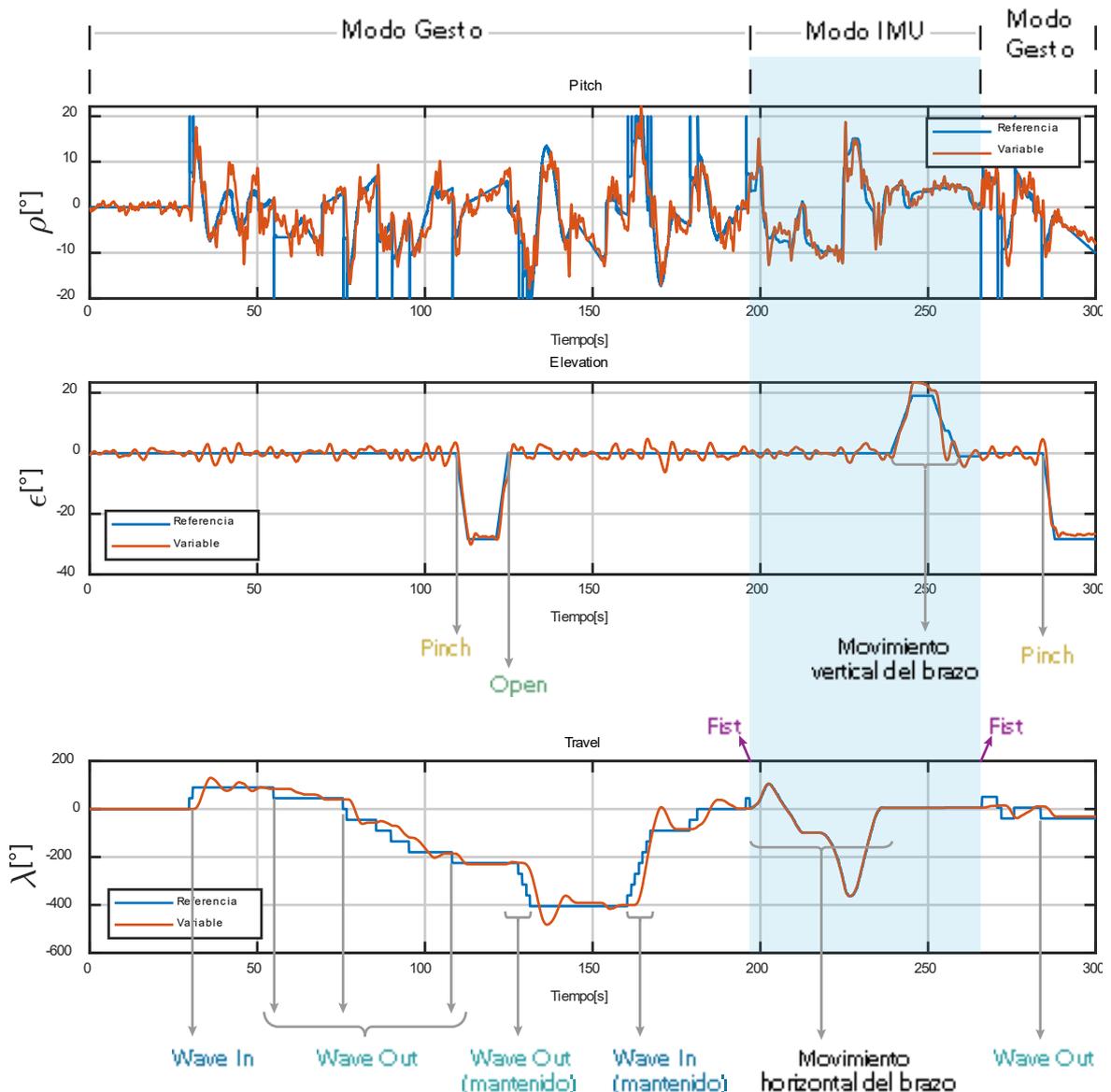
En la Figura 3.19 se puede observar una pantalla adicional, la cual muestra la interfaz diseñada para el funcionamiento del sistema. El usuario realiza los gestos y puede visualizar en esta pantalla el comando que se envía a la plataforma, la cual a su vez reacciona de acuerdo con el gesto o movimiento realizado.

#### 3.3.1.1. Modo Gesto

Durante el modo Myo la prueba realizada consistió en ejecutar todos los gestos y observar la respuesta obtenida en la plataforma física. Posteriormente se analizaron las gráficas obtenidas de esta prueba, las cuales se muestran en la Figura 3.20.

En la Figura 3.20 se puede observar la respuesta de cada una de las variables del helicóptero de 3-GDL al realizar los cinco diferentes gestos: Wave In, Wave Out, Fist, Open

y Pinch. Para los gestos de Wave Out y Wave In el cambio de referencia es muy pequeño para visualizarse en la escala de la gráfica. Sin embargo, alrededor de los 150s se puede observar el resultado de realizar estos gestos de forma sostenida con lo cual se generan varios cambios consecutivos de  $45^\circ$ .



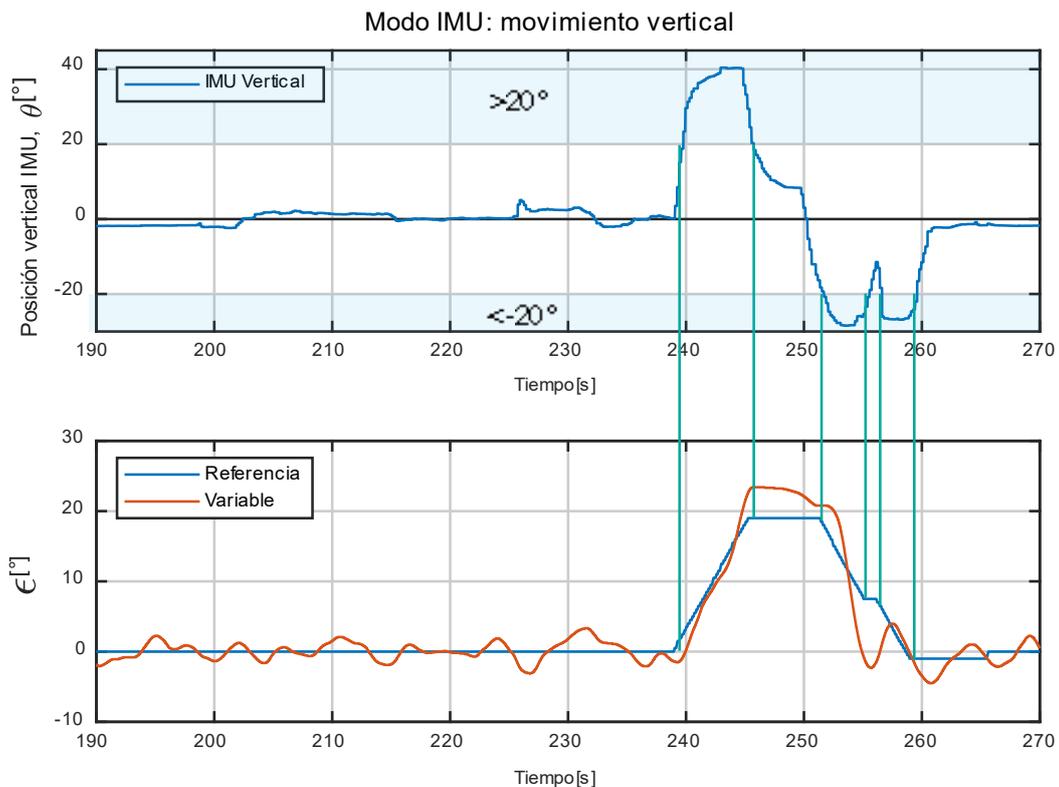
**Figura 3.20.** Gráficas de las variables del sistema para el control en modo Myo.

Como se mencionó en la Sección 2.4.3 de control de movimiento multimodo, durante el modo Gesto el controlador de travel funciona con un lazo interno que corresponde a la variable pitch. Esta es la razón por la cual al realizar los cambios de referencia (tipo paso) en la variable travel se observan picos en la referencia de pitch (gráfica azul) ya que corresponden a la acción derivativa del controlador travel.

En la gráfica de la Figura 3.20 se resalta en color celeste el movimiento de la plataforma durante el modo IMU. En los siguientes párrafos se describe esta parte con mayor detalle y se compara el movimiento de la plataforma con la información entregada por la IMU del sensor Myo Armband.

### 3.3.1.2. Modo IMU

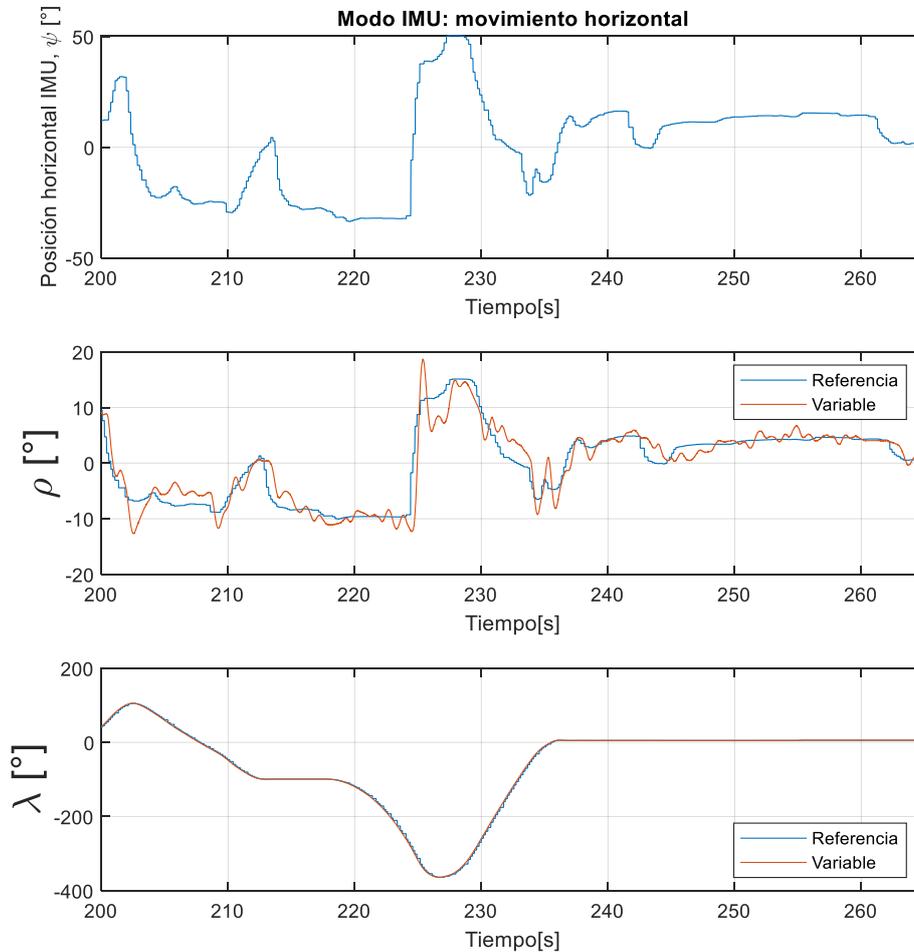
Como se explicó en la Sección 2.4.3 del capítulo anterior, al realizar el gesto Fist se activa el denominado modo IMU durante el control con el sensor Myo. Durante este modo de funcionamiento se desconecta el controlador de la variable travel y se envía un valor de referencia directamente al lazo de control de pitch. En el caso de elevation el sistema de control funciona de la misma forma y el valor de referencia se genera de acuerdo con la orientación vertical del brazo. En las Figuras 3.21 y 3.22 se observa una ampliación de la gráfica de la Figura 3.20 entre 190s y 270s que corresponde a los instantes en que se realiza el gesto Fist para activar el modo IMU. En estas figuras se ha separado el movimiento vertical con la variable elevation en una sola gráfica y el movimiento horizontal con travel y pitch en otra.



**Figura 3.21.** Movimiento vertical de la plataforma durante el modo IMU.

Como se puede observar en la Figura 3.21 existe una correspondencia entre la orientación vertical de la IMU (gráfica superior) y el movimiento de elevation (gráfica inferior). Se ha

resaltado en celeste las zonas en las que se modifica el setpoint de  $\epsilon$  que corresponden al rango en el que la orientación del sensor es mayor a  $20^\circ$  o menor a  $-20^\circ$  tal como se menciona en la sección de *Modo IMU* del capítulo anterior. De igual manera, cuando el valor de este ángulo se encuentra entre  $-20^\circ$  y  $20^\circ$  el valor de referencia de elevation se mantiene sin cambios.

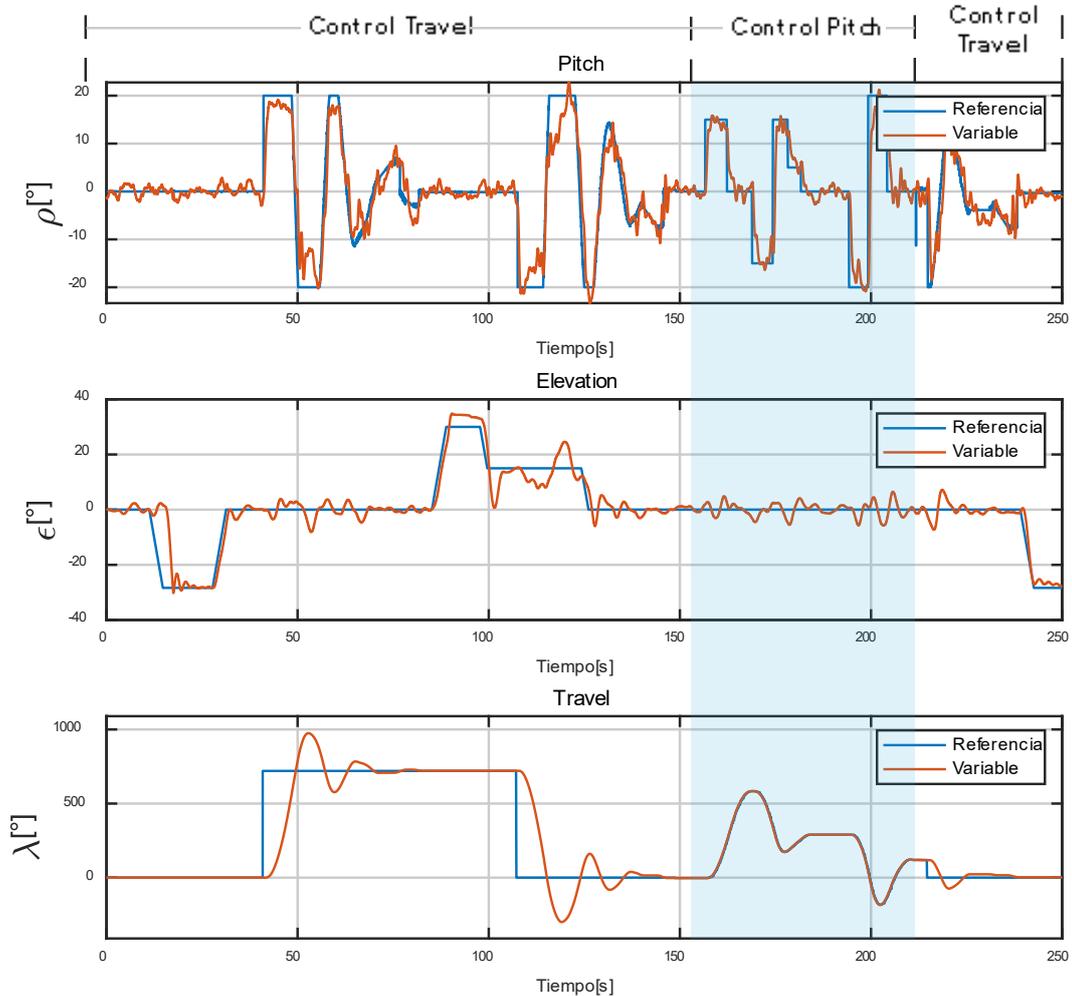


**Figura 3.22.** Movimiento horizontal de la plataforma durante el modo IMU.

En la Figura 3.22 se observa claramente la relación que existe entre el movimiento horizontal del brazo (IMU) y el setpoint de la variable pitch. Como se explicó en la sección de *Modo IMU* en la metodología, durante este modo de funcionamiento se desconecta el controlador de la variable travel y se mantiene únicamente el controlador de pitch. Por esta razón el movimiento horizontal de la IMU (gráfica superior de la Figura 3.22) se mapea directamente en el setpoint de pitch (gráfica media de la Figura 3.22) y se genera un desplazamiento correspondiente en la variable travel (gráfica inferior). En la gráfica inferior se puede observar también que el valor medido de la variable travel se copia al valor de referencia, al momento de regresar a modo gesto se mantenga el último valor medido.

### 3.3.2. MODO MANUAL

Adicional al control de la plataforma empleando el sensor Myo, el helicóptero de 3-GDL se puede manipular mediante comandos numéricos o con los sliders de la interfaz gráfica. En este *Modo Manual* se puede controlar el movimiento en travel o pitch por separado de forma correspondiente al modo gesto o IMU explicado en los párrafos anteriores.



**Figura 3.23.** Gráficas de las variables del sistema para el control en modo Gesto.

La capacidad de poder ingresar valores de setpoint de forma numérica permite hacer cambios de setpoint muy amplios en la variable Travel. Tal como se observa en la Figura 3.23, se han realizado cambios de setpoint de  $720^\circ$  (dos vueltas) en la variable Travel con una respuesta muy satisfactoria. De igual manera en la variable elevation se han probado para diferentes valores de referencia entre  $-30^\circ$  y  $30^\circ$ .

Finalmente, al igual que en el caso anterior, se puede escoger la activación del controlador de travel o pitch. Entre los 150s y 210s se activa el controlador de la variable pitch, en este caso se puede observar cambios de referencia tipo paso en dicha variable (gráfica superior

de la Figura 3.23) y se obtiene un desplazamiento correspondiente en la variable travel. Por otro lado, fuera de este intervalo de tiempo, la referencia de pitch es generada automáticamente por el lazo externo del controlador travel.

### **3.4. PUBLICACIÓN Y VIDEO**

Con base en los resultados expuestos en este capítulo y la metodología explicada en el capítulo anterior se ha elaborado un artículo titulado *Hand Gesture and Arm Movement Recognition for Multimodal Control of a 3-DOF Helicopter*, el cual actualmente se encuentra aceptado para su presentación en la novena conferencia “International Conference on Robot Intelligence Technology and Applications” (RITA) que se llevará a cabo el 16 y 17 de diciembre de 2021 en la ciudad de Daejeon en Corea del Sur. La presentación del artículo en cuestión está programada para el viernes 17 de diciembre a las 10h10 (hora de Corea).

Adicionalmente, los resultados expuestos en este capítulo se han resumido en un video que se encuentra disponible en la página web del Laboratorio de Investigación en Inteligencia y Visión Artificial “Alan Turing” de la EPN, al cual se puede acceder mediante el siguiente enlace:

[https://laboratorio-ia.epn.edu.ec/es/recursos/dataset-y-aplicaciones-2/2021\\_3dof\\_hgr\\_epn](https://laboratorio-ia.epn.edu.ec/es/recursos/dataset-y-aplicaciones-2/2021_3dof_hgr_epn)

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1. CONCLUSIONES

- En el presente proyecto se diseñó e implementó un prototipo de un helicóptero de 3 grados de libertad orientado al estudio de técnicas de control. Este prototipo se elaboró como una aplicación para el sistema de reconocimiento de gestos de la mano (HGR) elaborado en el proyecto de investigación PIGR-19-07. Mediante los cinco diferentes gestos que permite reconocer este sistema HGR se generan los distintos valores de referencia para los controladores de los tres ángulos de libertad del helicóptero, que se denominan travel, elevation y pitch.
- Debido a la gran cantidad de aplicaciones que se pueden otorgar a los vehículos aéreos no tripulados, su estudio y utilización ha cobrado cada vez mayor relevancia. Por esta razón, el uso del helicóptero de 3-GDL resulta importante ya que representa un banco de pruebas para comprobar la efectividad de los sistemas de control en UAVs de tipo multi - rotor.
- El sistema de reconocimiento de gestos de la mano utilizado en el presente proyecto permite reconocer cinco diferentes gestos de la mano que se denominan wave out, wave in, fist, open y pinch. Dicho sistema hace uso del sensor Myo Armband, el cual además permite obtener información de la orientación del brazo del usuario mediante una IMU. Utilizando estos cinco gestos y los datos de la IMU se implementó un sistema de comandos *multimodo* para controlar el movimiento del helicóptero de 3-GDL elaborado.
- Adicional al uso del sensor Myo, el helicóptero de 3-GDL puede ser controlado mediante comandos numéricos que se ingresan directamente en la interfaz gráfica elaborada en el entorno App Designer de Matlab. Esta forma de comandar el helicóptero se denomina *modo manual*.
- A partir de las medidas de elementos como rodamientos, tornillos y tubos de aluminio se elaboraron los modelos de las otras piezas en el software CAD Fusion 360. Partiendo de estos modelos se fabricaron las piezas mediante impresión 3D y corte láser, con lo cual fue posible elaborar piezas que calzan perfectamente ya que los modelos fueron realizados en base a las medidas de los materiales utilizados.
- Mediante el uso de materiales como perfiles y tubos de aluminio y técnicas como impresión 3D y corte láser fue posible implementar un prototipo de helicóptero de 3 grados de libertad completamente funcional por una fracción del costo de las

alternativas comerciales. Mediante estos materiales y métodos se elaboró dicho prototipo por un costo menor al 33% del valor de un modelo comercial similar.

- Mediante el uso de las ecuaciones de Lagrange se pudo obtener un modelo matemático dinámico del helicóptero de 3-GDL, el cual permitió reconocer el comportamiento de las variables y la relación que existe entre ellas. Mediante este modelo se pudo identificar tres subsistemas para cada ángulo de libertad  $\rho$ ,  $\epsilon$  y  $\lambda$  con las entradas  $F_b$  y  $F_f$ . Esto a su vez permitió desarrollar los controladores PID para cada una de las variables, con los cuales se regulan cada una de estas posiciones angulares.
- La obtención de un modelo matemático del sistema permite tener una idea clara del funcionamiento del mismo, lo cual es de gran ayuda para el diseño de controladores. Sin embargo, en la realidad existen comportamientos que pueden ser muy difíciles, o a veces imposibles de modelar como rozamientos, saturaciones, etc., lo cual resalta la importancia de la elaboración de un módulo didáctico para el aprendizaje y estudio de sistemas de control.
- A partir de los primeros resultados obtenidos se observó que el modelo encontrado para la variable travel presentaba algunas limitaciones debido a ciertas cualidades que no se pueden determinar fácilmente como por ejemplo el rozamiento y el valor de la fuerza  $F_{sum}$  para mantenerse en una elevación constante. Por este motivo se empleó el método heurístico de sintonización de Ziegler-Nichols, mediante el cual se encontraron las constantes del controlador PID que presentaron un mejor comportamiento que las determinadas a partir del modelo encontrado previamente.
- En primera instancia, la acción integral presentaba un comportamiento indeseable sobre todo en la variable travel. El efecto en cuestión se manifestaba como una inestabilidad en el sistema que se debía a la acumulación excesiva de la acción integral. Para contrarrestar este efecto se implementó un algoritmo de *anti wind-up* en la variable travel que consistió en “resetear” el controlador integral. Dicho reset se implementó en un rango de  $\pm 3^\circ$  y mayor a  $|20^\circ|$  en el cual se aplica simplemente un controlador de tipo PD. Con esta modificación se obtuvo una respuesta estable en la variable travel.
- El sistema de control y realimentación fue implementado en una placa Pyboard debido a la gran cantidad de ventajas que ofrece. El microcontrolador que posee es de tipo STM32 con una velocidad de 168MHz y permite la programación en Micropython, el cual facilita el manejo de periféricos como Timers, comunicación serial, etc. Adicionalmente, mediante esta placa se pudo hacer la lectura de los encoders de forma más directa ya que posee hardware dedicado en varios de sus pines.

- El uso de encoders para la realimentación de las variables proporciona la ventaja de tener una alta tolerancia ante el ruido. Puesto que la salida de los encoders utilizados es de tipo colector abierto NPN, las señales son simplemente valores lógicos de alto (3.3V) y bajo (0V). Esta cualidad permite realizar mediciones más robustas debido a que los rangos de valores lógicos en el microcontrolador son mayores que la amplitud del ruido que generalmente puede existir. Debido a esto el uso de encoders es preferible ante otras alternativas como sensores resistivos.
- La herramienta App Designer de Matlab se utilizó para elaborar una interfaz gráfica mediante la cual el usuario puede visualizar las variables del sistema en tiempo real, así como los gestos que realiza y el modo de funcionamiento. En esta interfaz se puede escoger el modo de funcionamiento manual (con comandos numéricos o usando sliders) o modo Myo (reconocimiento de gestos de la mano y movimientos del brazo).

## 4.2. RECOMENDACIONES

- Se recomienda mejorar el funcionamiento del sistema utilizando técnicas más avanzadas de control. Se puede utilizar el modelo obtenido para realizar técnicas de control óptimo o robusto basadas en modelo con lo cual se pueden comparar los diferentes desempeños de los controladores realizados.
- Se recomienda utilizar un sistema de reconocimiento de gestos de la mano como el presentado en este proyecto para controlar un robot móvil sin estar sujeto a una base que restrinja su movimiento. Aplicando un método multimodo con reconocimiento de gestos e IMU se podría controlar un UAV o un vehículo terrestre.
- El sistema de HGR elaborado por el proyecto de investigación PIGR-19-07 se encuentra actualmente incluyendo nuevas mejoras, como reconocimiento de un mayor número gestos y mayor exactitud. Se recomienda utilizar las nuevas versiones aplicadas al control de plataformas aéreas y sistemas de control en general.
- El manejo de la comunicación inalámbrica, el sistema de reconocimiento de gestos y la interfaz gráfica representó un cierto grado de complejidad debido a que en el entorno de Matlab se maneja cada herramienta por separado (Simulink, App Designer, scripts) e integrarlas puede ser complicado. Por esta razón se recomienda considerar diferentes entornos para la comunicación con el helicóptero y la elaboración de la interfaz, como por ejemplo alternativas basadas en Python u otros lenguajes.

## 5. REFERENCIAS BIBLIOGRÁFICAS

- [1] DJI, «DJI FPV - Redefine Flying,» 2021. [En línea]. Available: <https://www.dji.com/dji-fpv>. [Último acceso: 6 Julio 2021].
- [2] O. Liang, «The FPV Goggles Buyer's Guide – What Should You Buy,» 13 Junio 2021. [En línea]. Available: <https://oscarliang.com/fpv-goggles/>. [Último acceso: Julio 2021].
- [3] O. M. Govea R., «Revista Digital Robótica Industrial,» Issuu, 30 Enero 2016. [En línea]. Available: [https://issuu.com/maytiel/docs/30\\_-1-16](https://issuu.com/maytiel/docs/30_-1-16). [Último acceso: Julio 2021].
- [4] Federal Aviation Administration, «Pilot's Handbook of Aeronautical Knowledge,» de *Chapter 6: Flight Controls*, Oklahoma City, 2016, pp. 6.1-6.12.
- [5] A. F. El-Sayed, *Fundamentals of Aircraft and Rocket Propulsion*, Zagazig: Springer, 2016.
- [6] P. Cantrell, «Angle of Attack - Helicopter Aviation,» [En línea]. Available: [http://www.copters.com/aero/angle\\_of\\_attack.html](http://www.copters.com/aero/angle_of_attack.html). [Último acceso: Julio 2021].
- [7] NASA's Mars Exploration Program and the Jet Propulsion Laboratory, «Mars Helicopter Tech Demo,» 2021. [En línea]. Available: <https://mars.nasa.gov/technology/helicopter/#>. [Último acceso: Julio 2021].
- [8] B. Jackson, «Fixed Wing vs Multirotor Drones for Surveying,» COPTRZ.com, 24 Febrero 2021. [En línea]. Available: <https://coptrz.com/fixed-wing-vs-multirotor-drones-for-surveying/>. [Último acceso: Julio 2021].
- [9] A. Abdelkef y M. Hassanalian, «Classifications, applications, and design challenges of drones: A review,» *Progress in Aerospace Sciences*, vol. 91, pp. 99 - 131, 2017.
- [10] Quanser, «3 DOF Helicopter,» Nvision, 2021. [En línea]. Available: <https://www.quanser.com/products/3-dof-helicopter/>. [Último acceso: Julio 2021].
- [11] Googol Technology, «3DOF Helicopter Simulator,» Googol Technology, 2020. [En línea]. Available: [http://www.googoltech.com/pro\\_view-70.html](http://www.googoltech.com/pro_view-70.html). [Último acceso: Julio 2021].
- [12] Quanser, «3 DOF Hover - Flight dynamics and control of vertical lift-off vehicles,» nvision, 2021. [En línea]. Available: <https://www.quanser.com/products/3-dof-hover/>. [Último acceso: Julio 2021].
- [13] Googol Technology, «Four Rotor Hover Vehicle,» 2020. [En línea]. Available: [http://www.googoltech.com/pro\\_view-71.html](http://www.googoltech.com/pro_view-71.html). [Último acceso: Julio 2021].
- [14] R. Tedrake, *Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines*, MIT, 2009.

- [15] K. J. Åström, «PID Control,» de *Control System Design*, Lund, Sweden, 2002, pp. 216-251.
- [16] L. E. Romero, D. F. Pozo y J. A. Rosales, «Quadcopter stabilization by using PID controllers,» de *Actas del Congreso de Reconocimiento de Patrones, Control Inteligente, Comunicaciones e Ingeniería Biomédica*, Cuenca, 2014.
- [17] Statista, «Smartphone subscriptions worldwide 2016-2026,» 2021. [En línea]. Available: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>. [Último acceso: Julio 2021].
- [18] M. B. Jonathan Zea, «Real-time hand gesture recognition: A long short-term memory approach with electromyography,» de *International Conference on Computer Science, Electronics and Industrial Engineering (CSEI)*, Ambato, Ecuador, 2019.
- [19] Microsoft , «Azure Kinect DK,» Microsoft, 2021. [En línea]. Available: <https://azure.microsoft.com/en-us/services/kinect-dk/>. [Último acceso: Julio 2021].
- [20] Ultraleap, «Leap Motion Controller,» Ultraleap, 2021. [En línea]. Available: <https://www.ultraleap.com/product/leap-motion-controller/>. [Último acceso: Julio 2021].
- [21] S. Ahlawat , V. Batra, S. Banerjee, J. Saha y A. K. Ahlawat, «Hand Gesture Recognition Using Convolutional Neural Network,» de *International Conference on Innovative Computing and Communications*, Singapur, 2019.
- [22] CyberGlove Systems, «CyberGlove II,» CyberGlove Systems, 2017. [En línea]. Available: <http://www.cyberglovesystems.com/cyberglove-ii/>. [Último acceso: Julio 2021].
- [23] K. Minwoo, C. Jaechan , L. Seongjoo y J. Yunho, «IMU Sensor-Based Hand Gesture Recognition for Human-Machine Interfaces,» *MDPI: Inertial Sensors*, vol. 19, 2019.
- [24] J. A. Zea Guachamín, Implementación de un Sistema de Clasificación de Gestos del Brazo Humano Utilizando Myo Armband para Mando a Distancia de un Brazo Robótico de 3GDL, Quito: Escuela Politécnica Nacional, 2017.
- [25] P. Konrad, *The ABC of EMG: A Practical Introduction to Kinesiological Electromyography*, Scottsdale: NORAXON, 2005.
- [26] M. B. I. Raez, M. S. Hussain y S. Mohd-Yasin, «Techniques of EMG signal analysis: detection, processing, classification and applications,» *Biol Proced Online*, vol. 8, pp. 11-35, 2006.
- [27] K. Xing, Z. Ding, S. Jiang, X. Ma, K. Yang, C. Yang, X. Li y F. Jiang, «Hand Gesture Recognition Based On Deep Learning Method,» de *Third International Conference on Data Science in Cyberspace*, Guangzhou, China, 2018.
- [28] W.-T. Shi, Z.-J. Lyu, S.-T. Tang, T.-L. Chia y C.-Y. Yang, «A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study,» *Biocybernetics and Biomedical Engineering*, vol. 38, n° 1, pp. 126-135, 2018.

- [29] Thalmic Labs [2013-2018], «Myo Blog,» [En línea]. Available: <https://developerblog.myo.com/>. [Último acceso: Julio 2021].
- [30] P. Visconti, F. Gaetani, G. A. Zappatore y P. Primiceri, «Technical Features and Functionalities of Myo Armband: An Overview on Related Literature and Advanced Applications of Myoelectric Armbands Mainly Focused on Arm Prostheses,» *INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS*, vol. 0, nº 0, pp. 1-25, 2018.
- [31] L. Barona, L. Valdivieso, V. Vimos, J. Zea, J. P. Váscquez, M. Álvarez y M. Benalcázar, «An Energy-Based Method for Orientation Correction of EMG Bracelet Sensors in Hand Gesture Recognition Systems,» *MDPI Sensors*, vol. 20, nº 6327, 2020.
- [32] M. E. Benalcazar, L. Barona, L. Valdivieso, X. Aguas y J. Zea, «EMG-EPN-612 Dataset,» 6 Noviembre 2020. [En línea]. Available: <https://doi.org/10.5281/zenodo.4023305>. [Último acceso: Julio 2021].
- [33] V. H. Vimos, M. Benalcázar, A. F. Oña y P. J. Cruz, «A Novel Technique for Improving the Robustness to Sensor Rotation in Hand Gesture Recognition Using sEMG,» *Advances in Intelligent Systems and Computing*, vol. 1078, nº 1, pp. 226-243, 2019.
- [34] R. Gandhi, «Support Vector Machine — Introduction to Machine Learning Algorithms,» towards data science, 7 Junio 2018. [En línea]. Available: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>. [Último acceso: Julio 2021].
- [35] ST Microelectronics , «STM32F405RG,» 2021. [En línea]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f405rg.html>. [Último acceso: Agosto 2021].
- [36] N. H. Tollervey, Programming with MicroPython: embedded programming with microcontrollers and Python., Sebastopol, California, EEUU: O'Reilly, 2017.
- [37] Adafruit, «MicroPython, bringing Python to hardware for everyone,» 22 Octubre 2019. [En línea]. Available: <https://blog.adafruit.com/2019/10/22/micropython-bringing-python-to-hardware-for-everyone-ohm2019-oshwa-ohsummit-opensource-opensourcehardware-opensourceorg-micropython-micropython/>. [Último acceso: Agosto 2021].
- [38] D. P. George, P. Sokolovsky y contributors, «Quick reference for the pyboard,» 09 Julio 2021. [En línea]. Available: <https://docs.micropython.org/en/latest/pyboard/quickref.html>. [Último acceso: Agosto 2021].
- [39] CALT, «Economical solid shaft encoder–ES38 series,» Wordpress, 2021. [En línea]. Available: <https://www.china-encoder.com/product/economical-solid-shaft-encoder-es38-series/>. [Último acceso: Agosto 2021].
- [40] B. Kletzander, «The benefits of brushless motors, with the application simplicity of brushed motors,» 2016. [En línea]. Available:

[https://www.onsemi.com/site/pdf/ONSAR2751\\_EDN\\_Europe\\_0616.pdf](https://www.onsemi.com/site/pdf/ONSAR2751_EDN_Europe_0616.pdf). [Último acceso: Agosto 2021].

- [41] Rhydolabz, «A2212/13T TECHNICAL DATA,» [En línea]. Available: [https://www.rhydolabz.com/documents/26/BLDC\\_A2212\\_13T.pdf](https://www.rhydolabz.com/documents/26/BLDC_A2212_13T.pdf). [Último acceso: Agosto 2021].
- [42] RC Electric Parts, «User Guide: RC Electric Parts' Electric Speed Controller (ESC) for Brushless Motors,» 2020. [En línea]. Available: <https://www.rcelectricparts.com/esc-user-guide.html>. [Último acceso: Septiembre 2021].
- [43] C. Toapanta y J. Villafuerte, Diseño e Implementación de un Módulo para el Aprendizaje del Control de Orientación de un Cuadricóptero, Quito: EPN, 2019.
- [44] D. Chicco, M. J. Warrens y G. Jurman, «The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,» *PeerJ Computer Science*, vol. 7, n° 623, pp. 1-24, 2021.
- [45] DIGI, «Digi XBee 3 Zigbee 3 RF Module,» 2021. [En línea]. Available: <https://www.digi.com/products/embedded-systems/digi-xbee/xf-modules/2-4-ghz-rf-modules/xbee3-zigbee-3>. [Último acceso: Septiembre 2021].
- [46] Connectivity Standards Alliance, «Zigbee: the full-stack solution interlacing all your smart devices.,» 2021. [En línea]. Available: <https://zigbeealliance.org/solution/zigbee/>. [Último acceso: Septiembre 2021].
- [47] E. Georgakakis, D. .. Vergados, D. D. Vergados y C. Douligeris, «An Analysis of Bluetooth, Zigbee and Bluetooth Low Energy and Their Use in WBANs,» de *Wireless Mobile Communication and Healthcare*, Ayia Napa, Chipre, 2010.
- [48] Shenzhen Taida Century Technology Co., Ltd., «Taidacent 2 Wires 15A 20A 30A 40A 60A Slip Ring Collector Ring Rotary Electrical Contact (30A),» Amazon.com, 2021. [En línea]. Available: [https://www.amazon.com/Taidacent-Collector-Rotary-Electrical-Contact/dp/B07PY1SZ2T/ref=sr\\_1\\_1\\_sspa?crd=18DPJD9U9PHEM&dchild=1&keywords=30a%2Bslip%2Bring&qid=1630901923&sprefix=30a%2Bslip%2Caps%2C231&sr=8-1-spons&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUFRFRERXVFRXR](https://www.amazon.com/Taidacent-Collector-Rotary-Electrical-Contact/dp/B07PY1SZ2T/ref=sr_1_1_sspa?crd=18DPJD9U9PHEM&dchild=1&keywords=30a%2Bslip%2Bring&qid=1630901923&sprefix=30a%2Bslip%2Caps%2C231&sr=8-1-spons&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUFRFRERXVFRXR). [Último acceso: Septiembre 2021].
- [49] D. George, «pyboard.lbr,» Github, 22 Diciembre 2013. [En línea]. Available: <https://github.com/micropython/pyboard/blob/master/eagle/pyboard.lbr>. [Último acceso: Septiembre 2021].
- [50] Fusion 360, «Center of mass,» Autodesk, 2021. [En línea]. Available: <https://help.autodesk.com/view/fusion360/ENU/?guid=GUID-EB8C9FD9-2CA6-4937-887D-0014E538BF2C>. [Último acceso: Septiembre 2021].
- [51] F. S. Erik Bodin, Modelling & Control of a 3DOF Helicopter, Linköping: Linköping Univeristy Institute of Technology, 2015.

- [52] G. Abramson, *Notas de Mecánica Clásica*, Mendoza, Argentina: Universidad Nacional de Cuyo, 2018.
- [53] U. Pérez-Ventura, L. Fridman, E. Capello y E. Punta, «Fault tolerant control based on continuous twisting algorithms of a 3-DoF helicopter prototype,» *Control Engineering Practice*, vol. 101, pp. 1-14, 2020.
- [54] J. H. Lee, J. H. Choi y K. S. Lee, *Overview of Process Control*, Daejeon, Korea, 1997.
- [55] P. Leica, *Clases de Sistemas de Control Discreto*, Quito: EPN, 2019.
- [56] M. Tomaszewski, «Myo SDK MATLAB MEX Wrapper,» Github, 24 Julio 2017. [En línea]. Available: <https://github.com/mark-toma/MyoMex>. [Último acceso: Septiembre 2021].
- [57] M. Ben-Ari, *A Tutorial on Euler Angles and Quaternions*, Mountain View, California: Creative Commons, 2017.
- [58] JLC PCB, «Welcome to JLCPCB,» 2021. [En línea]. Available: <https://jlcpcb.com/aboutUs>. [Último acceso: Septiembre 2021].
- [59] Texas A&M University, «Purchase Order for Laboratory Equipment,» 3 Agosto 2018. [En línea]. Available: [https://assets.system.tamus.edu/files/budgets-acct/pdf/SB20\\_Contracts/AB0382653.pdf](https://assets.system.tamus.edu/files/budgets-acct/pdf/SB20_Contracts/AB0382653.pdf). [Último acceso: 16 Enero 2021].
- [60] C. Bohn y D. P. Atherton, «An analysis package comparing PID anti-windup strategies.,» de *IEEE/IFAC Joint Symposium on CACSD*, Tucson, Arizona, USA, 1995.

# **ANEXOS**

A continuación, se presentan los anexos con información adicional referente a la elaboración del presente proyecto.

**ANEXO A.** Detalles de los diseños de las placas PCB.

**ANEXO B.** Planos de las piezas elaboradas en Fusion 360 para la implementación del prototipo mecánico

**ANEXO C.** Guía de Instalación y Uso de Software de Reconocimiento HGR de Myo con Matlab

**ANEXO D.** Manual de Usuario: Información y uso del prototipo y la interfaz gráfica.

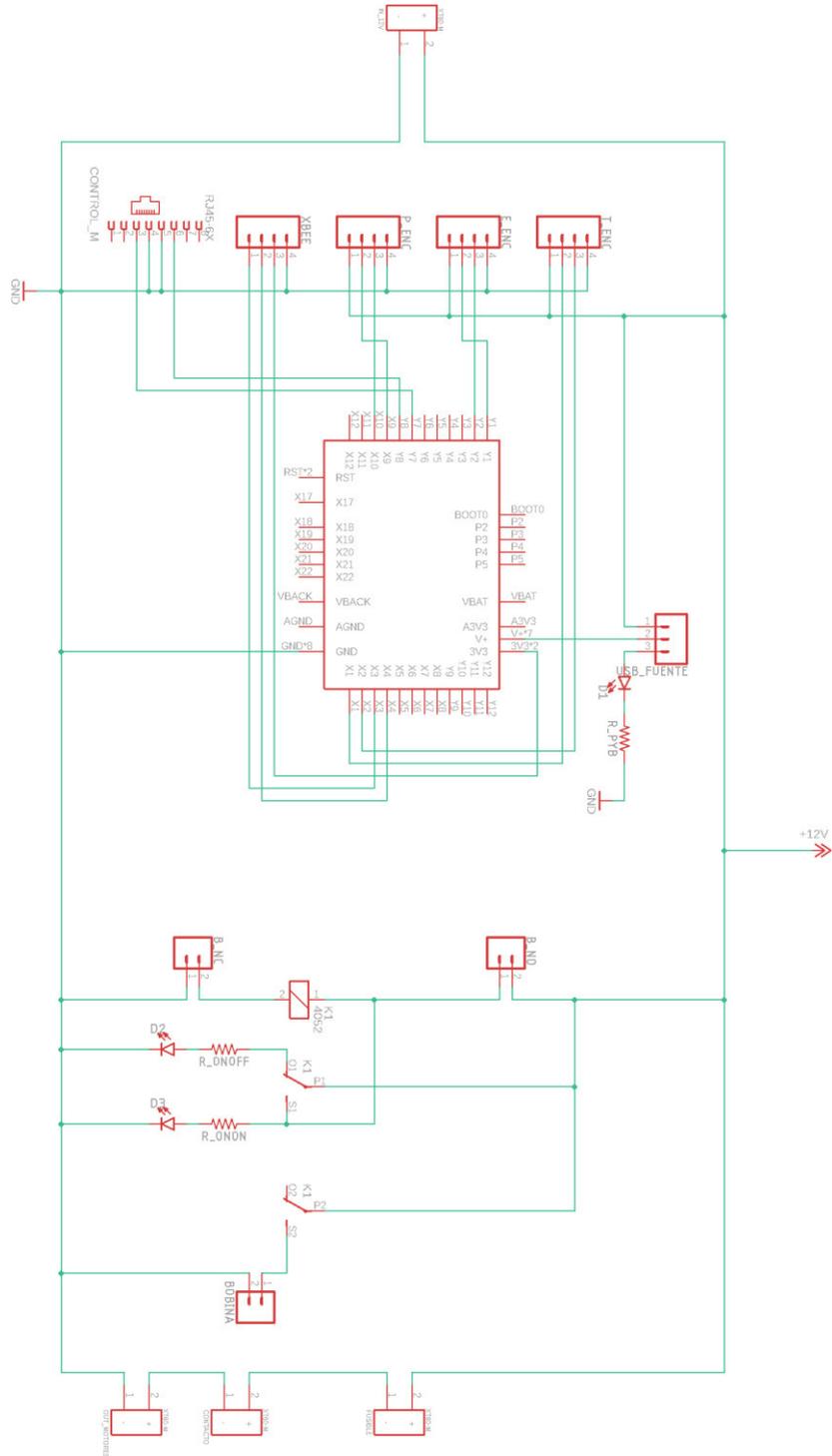
**ANEXO E.** Resumen de costos de elaboración del prototipo de helicóptero de 3-GDL.

# ANEXO A

Detalles de los circuitos de las placas de circuito impreso PCBs

## A.1. PLACA PRINCIPAL

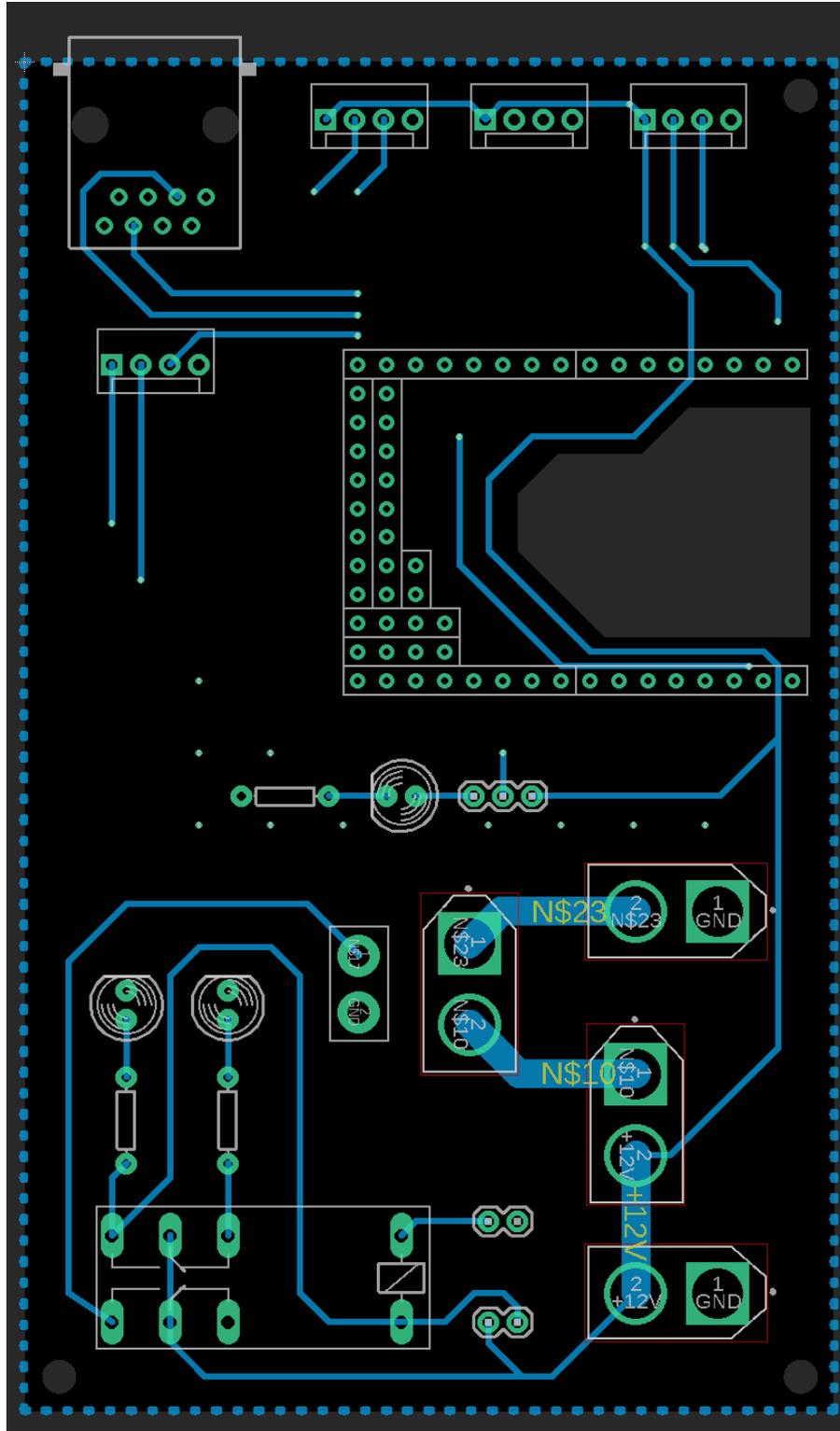
### A.1.1. Diagrama circuitual





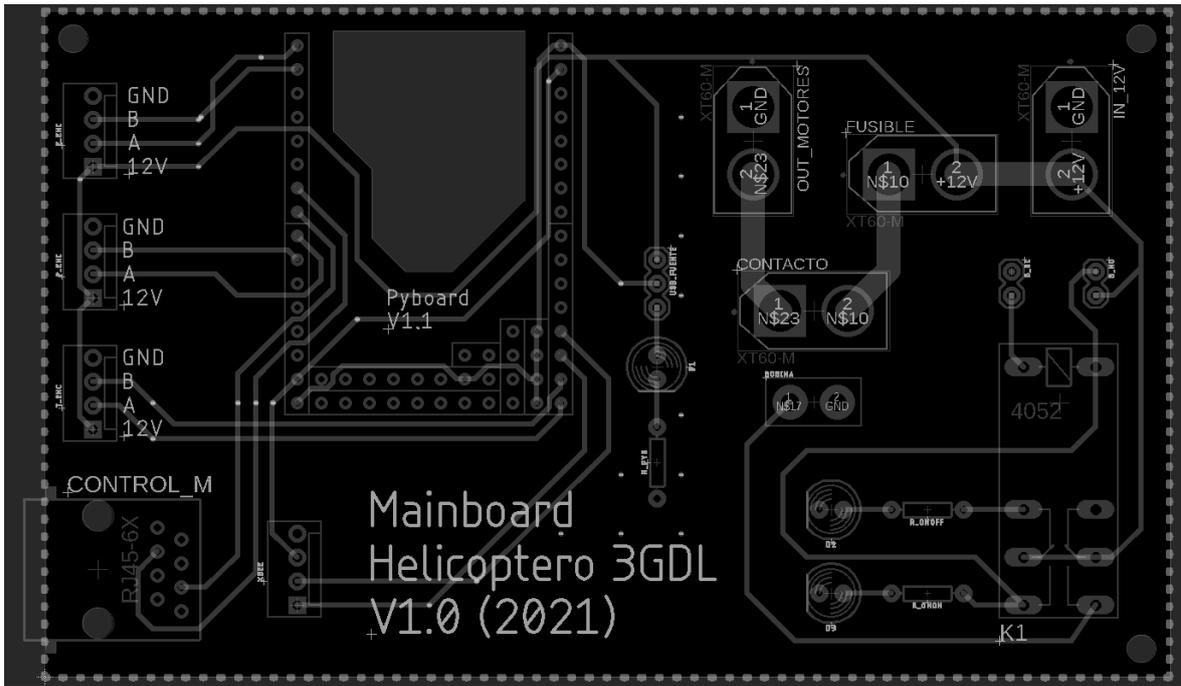
### A.1.3. Placa de circuito impreso: capa inferior

En esta imagen se muestra vista de la capa inferior de la placa en azul (Bottom) y la malla de tierra (rectángulo de líneas entrecortadas). Además, también se observa la capa superior de ubicación de los elementos (tPlace) junto con Vías y Pads.

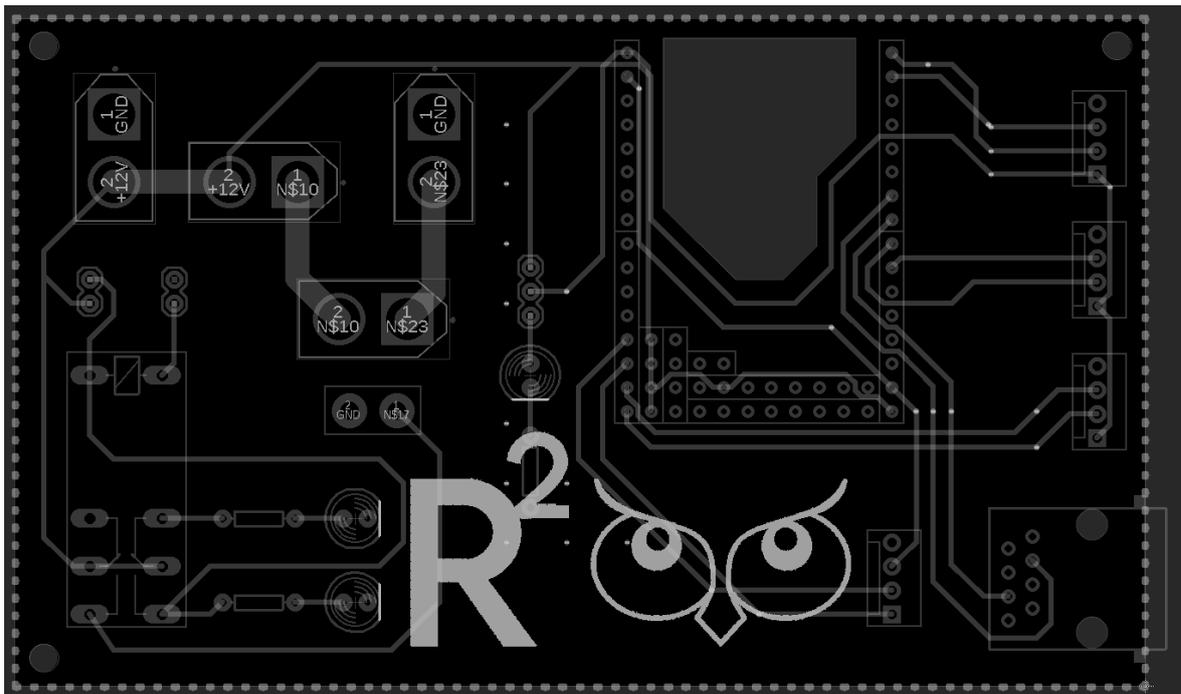


#### A.1.4. Placa de circuito impreso: nombres y logos

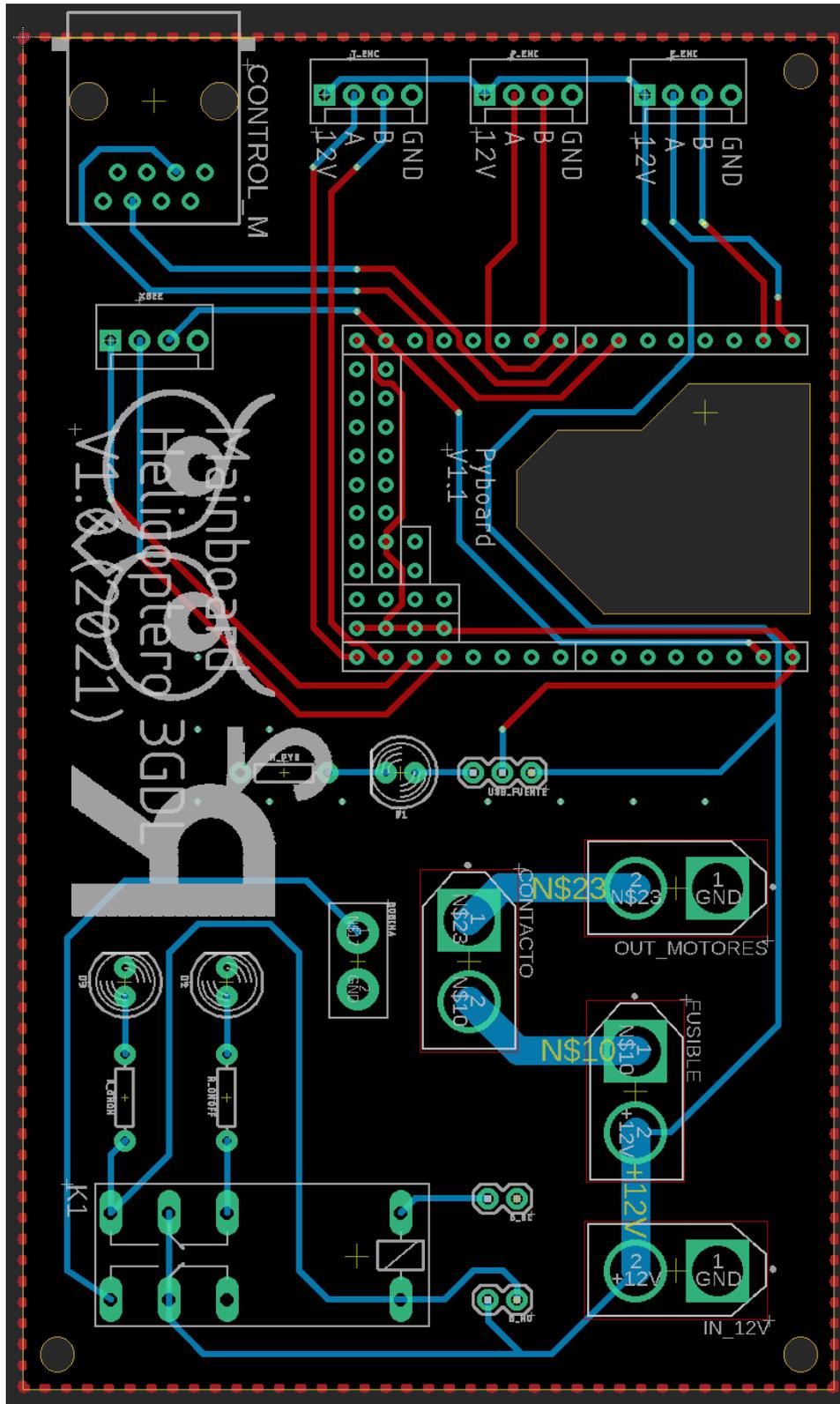
Vista de la capa de los nombres ubicados en la parte superior de la placa (tNames).



Vista de los logos en la parte inferior (bPlace). La placa se encuentra rotada.

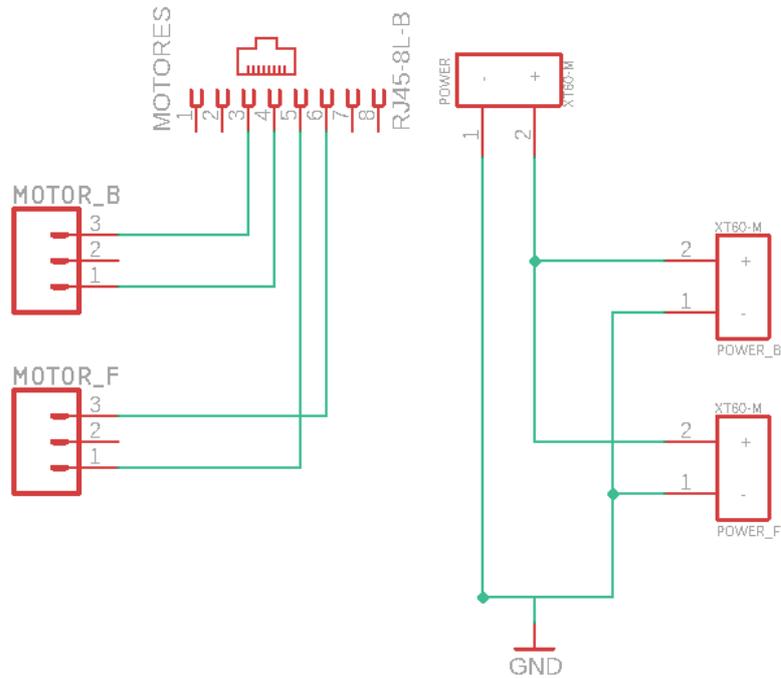


### A.1.5. Vista de la placa completa



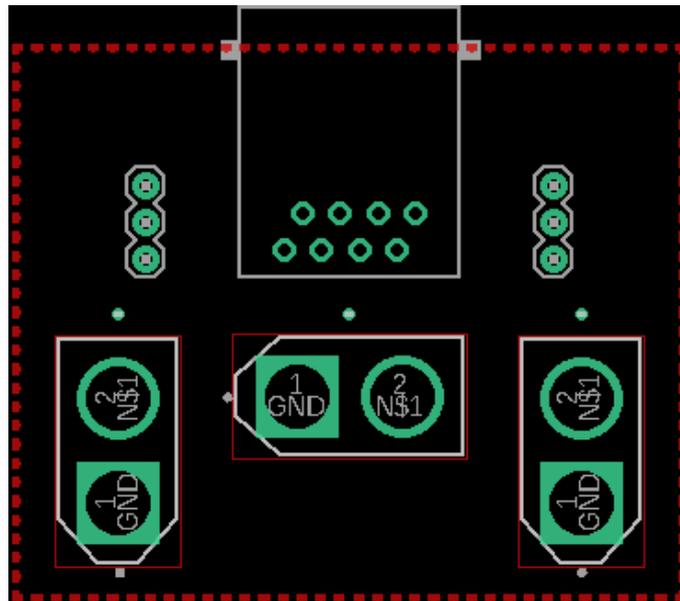
## A.2. PLACA AUXILIAR

### A.2.1. Diagrama circuital



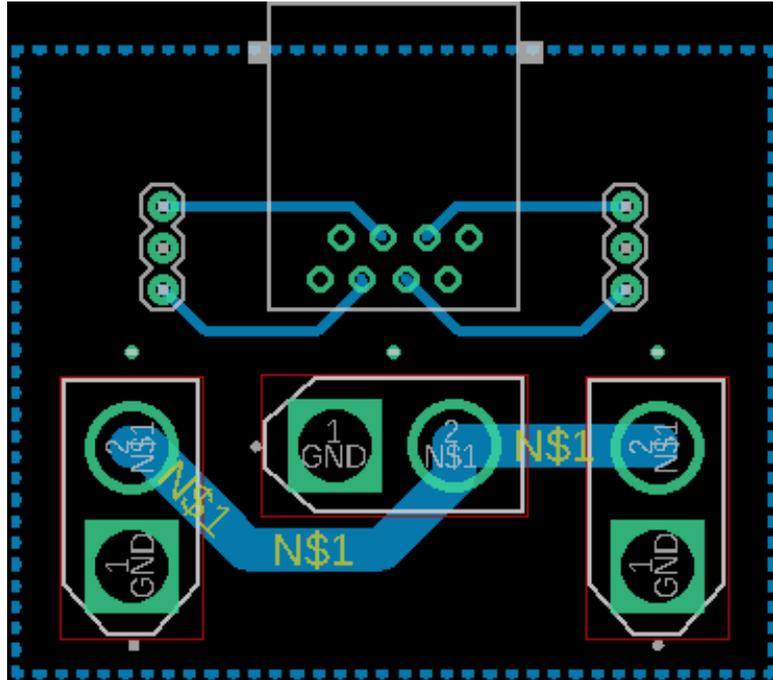
### A.2.2. Vista de la placa auxiliar: capa superior

En esta imagen se muestra la vista de la capa superior de la placa en color rojo (Top), malla de tierra, capa superior de ubicación de los elementos (tPlace), Vías y Pads.



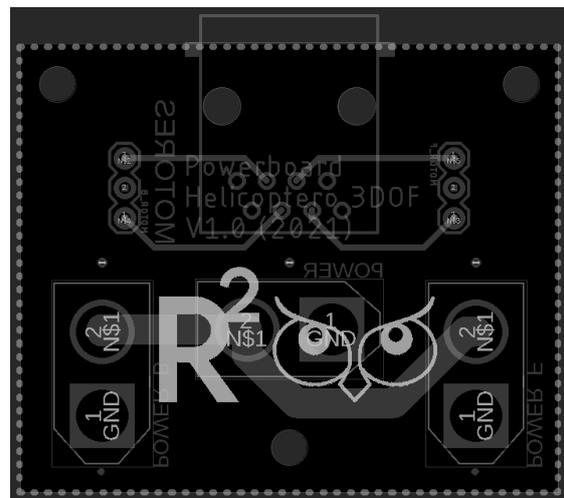
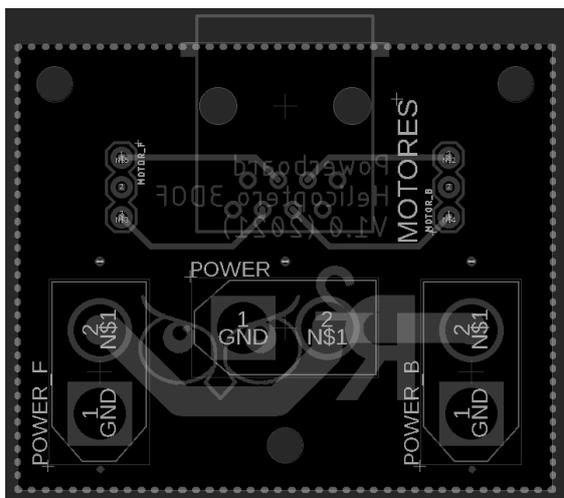
### A.2.3. Vista de la placa auxiliar: capa inferior

En esta imagen se muestra una vista de la capa inferior de la placa en color azul (Bottom), malla de tierra, capa superior de ubicación de los elementos (tPlace), Vías y Pads.

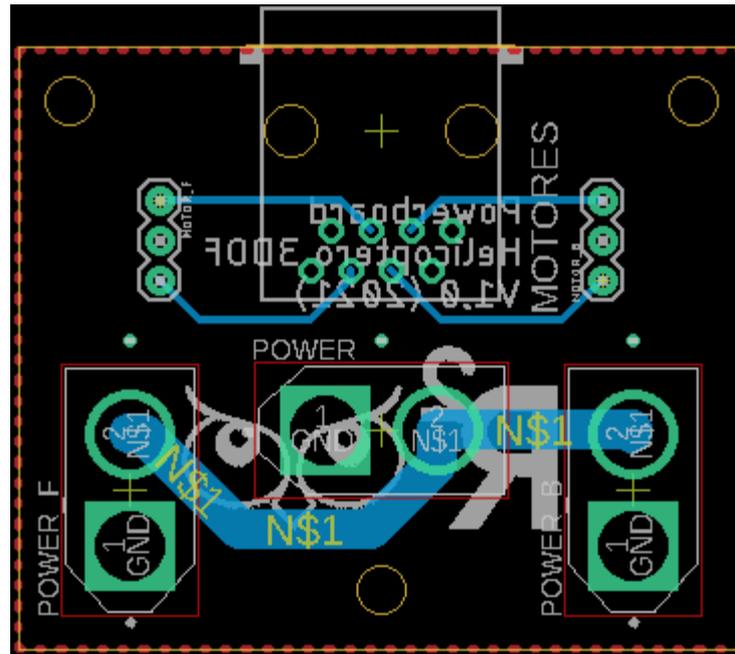


#### A.2.4. Vista de la placa auxiliar: nombres y logos

Vista de la capa de los nombres ubicados en la parte superior (tNames) a la izquierda y logos ubicados en la parte inferior (bPlace) a la derecha. En esta imagen la placa se encuentra rotada.



### A.2.5. Vista de la placa completa.

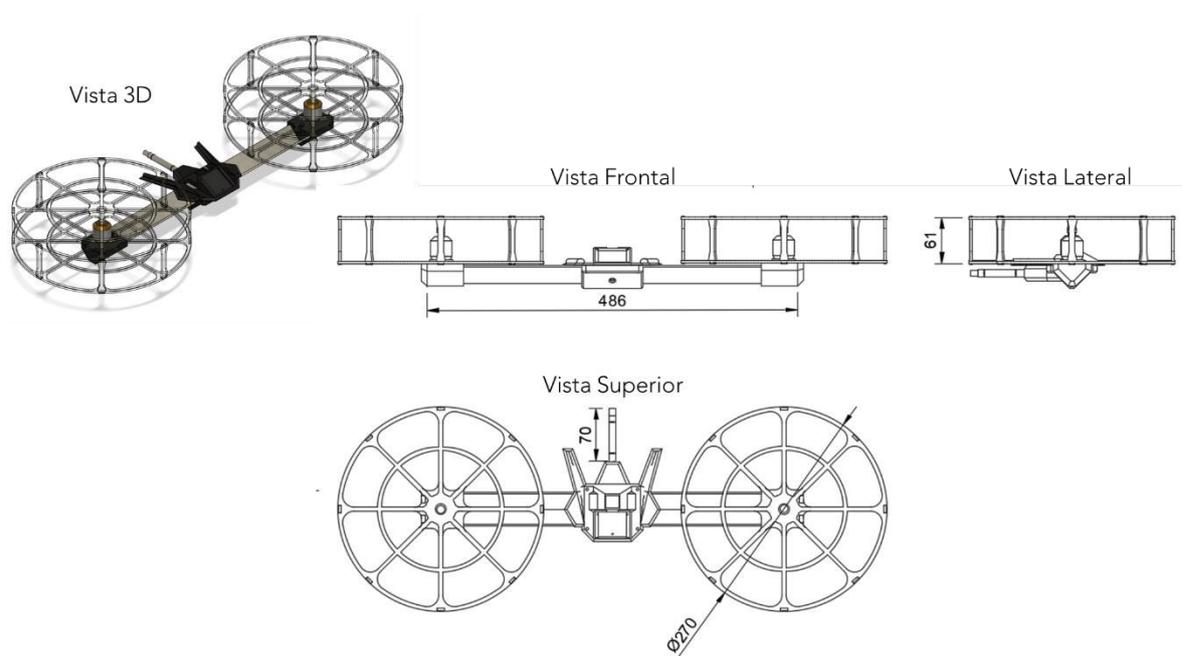


# ANEXO B

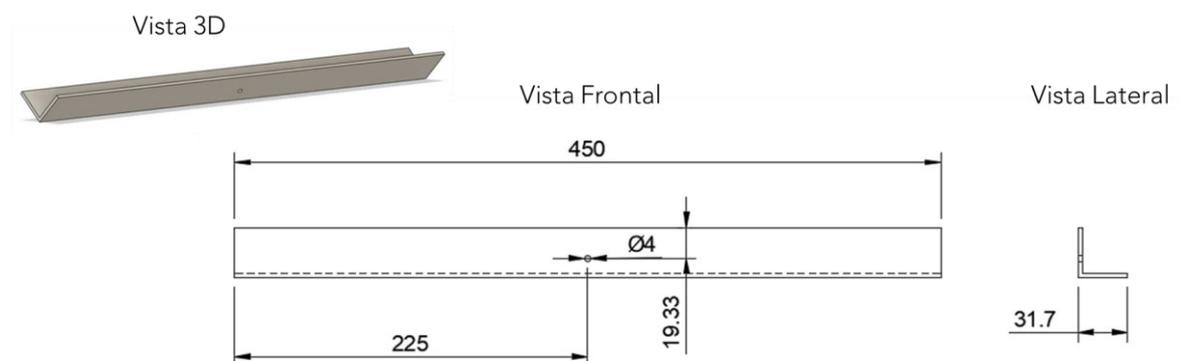
## Planos de las piezas elaboradas en Fusion 360

En este anexo se muestran los planos con detalles de dimensiones y proyecciones de todas las piezas elaboradas en Fusion 360 para el ensamblado del helicóptero de 3-GDL de acuerdo con lo explicado en la Sección 2.2.2. CONSTRUCCIÓN DEL PROTOTIPO MECÁNICO. Todas las dimensiones se encuentran en mm.

### B.1. CUERPO DEL HELICÓPTERO



#### B.1.1. BARRA CENTRAL DEL HELICÓPTERO

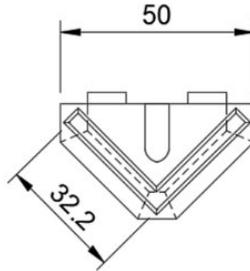


### B.1.2. SOPORTES DE MOTORES

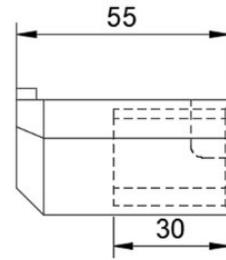
Vista 3D



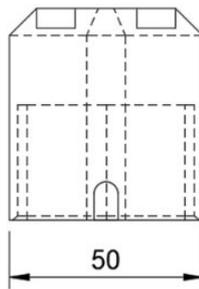
Vista Frontal



Vista Lateral

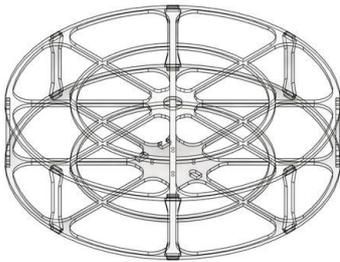


Vista Superior

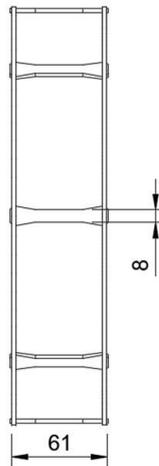


### B.1.3. PROTECTORES DE ACRÍLICO

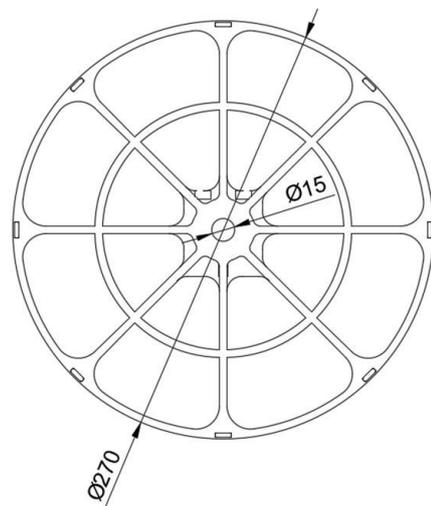
Vista 3D



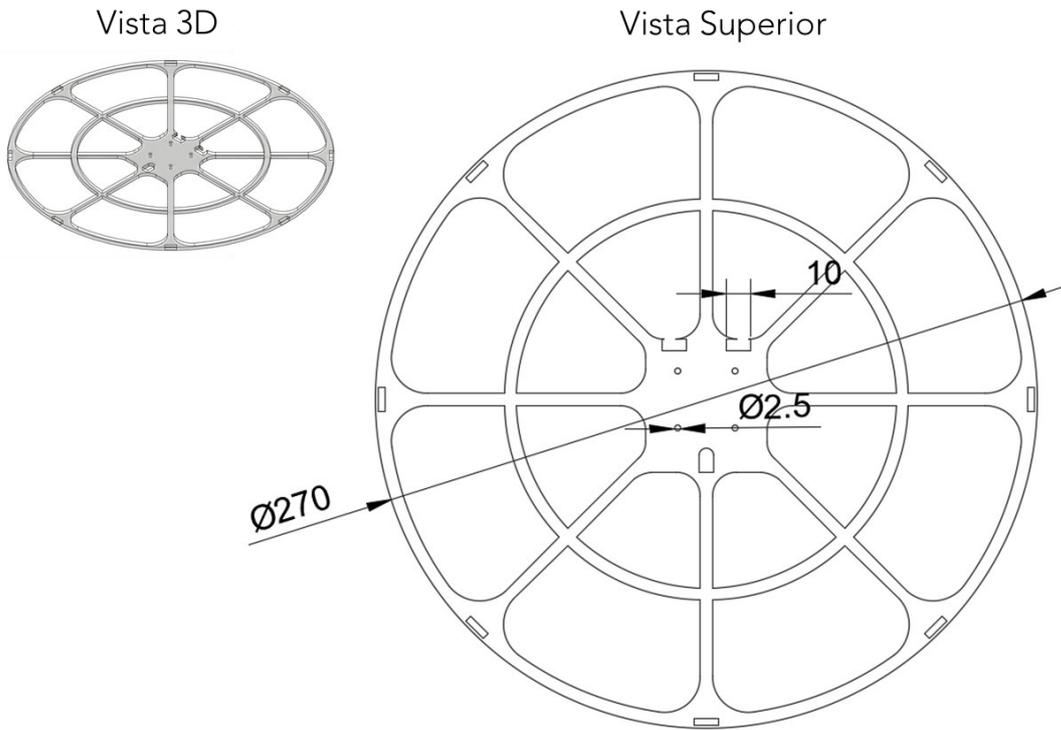
Vista Lateral



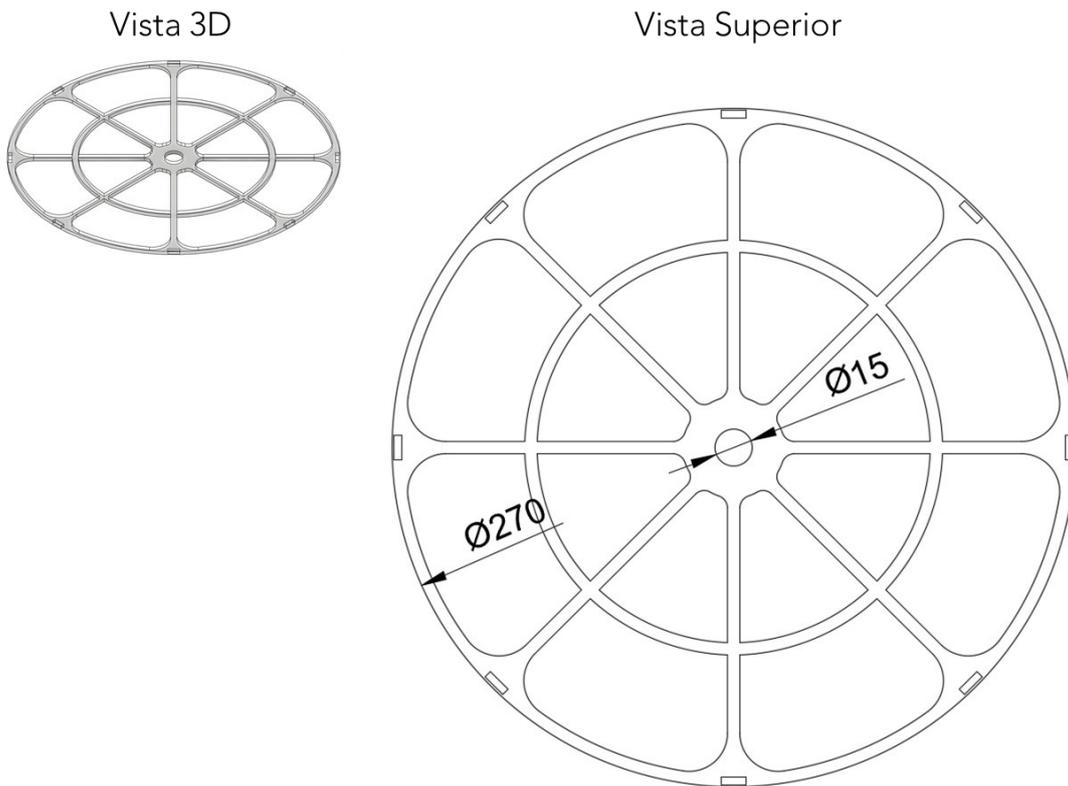
Vista Superior



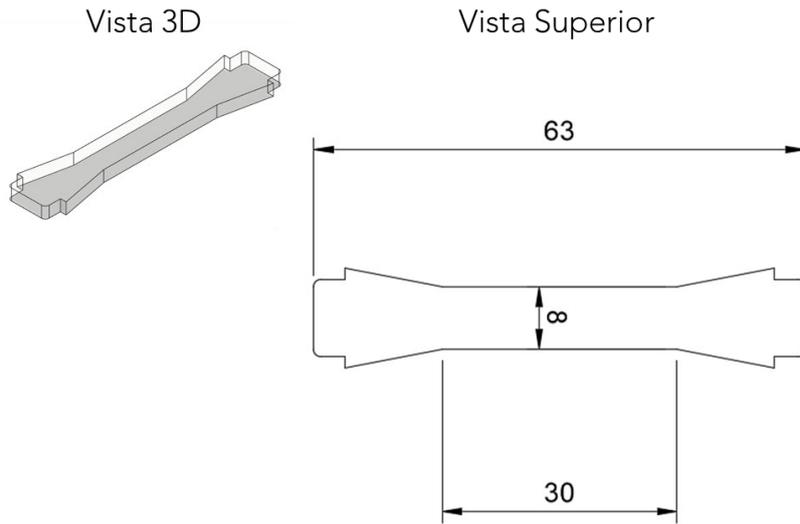
### B.1.3.1. Protector Inferior



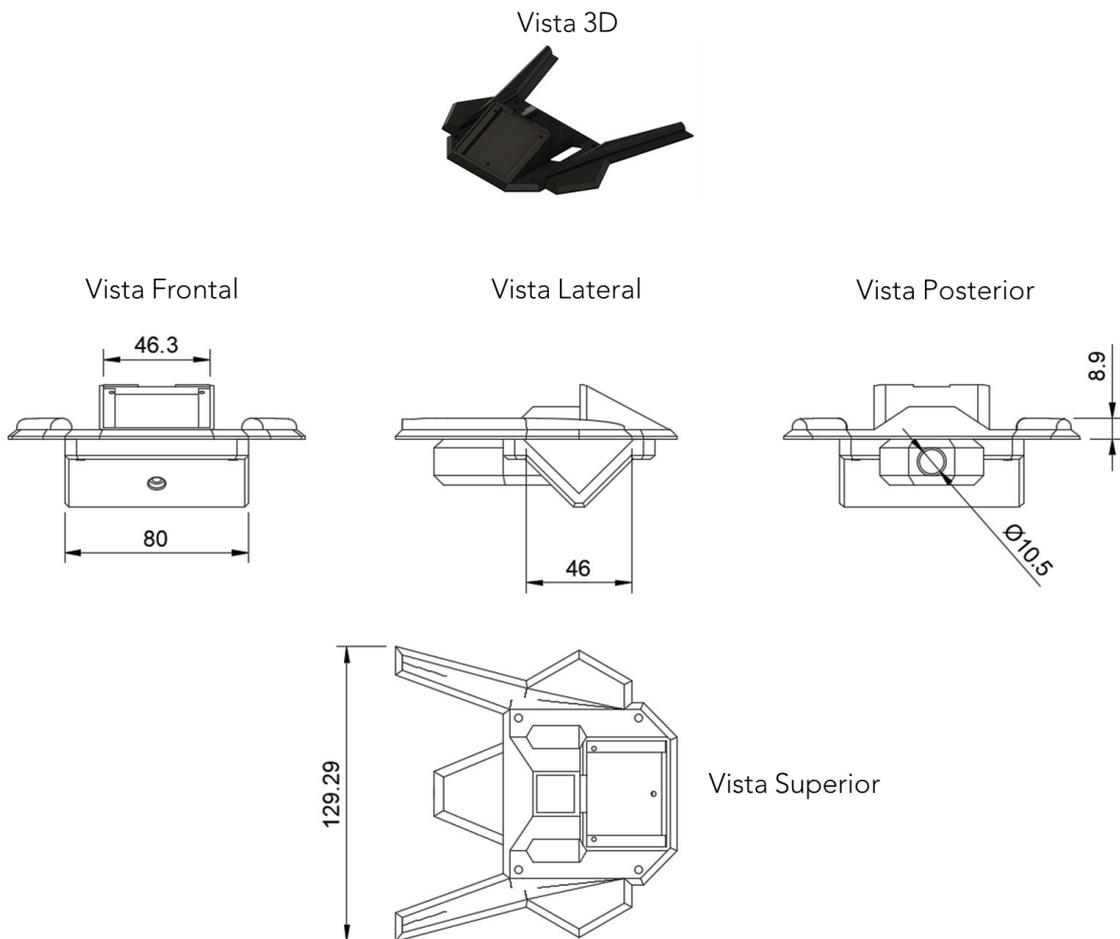
### B.1.3.2. Protector Superior



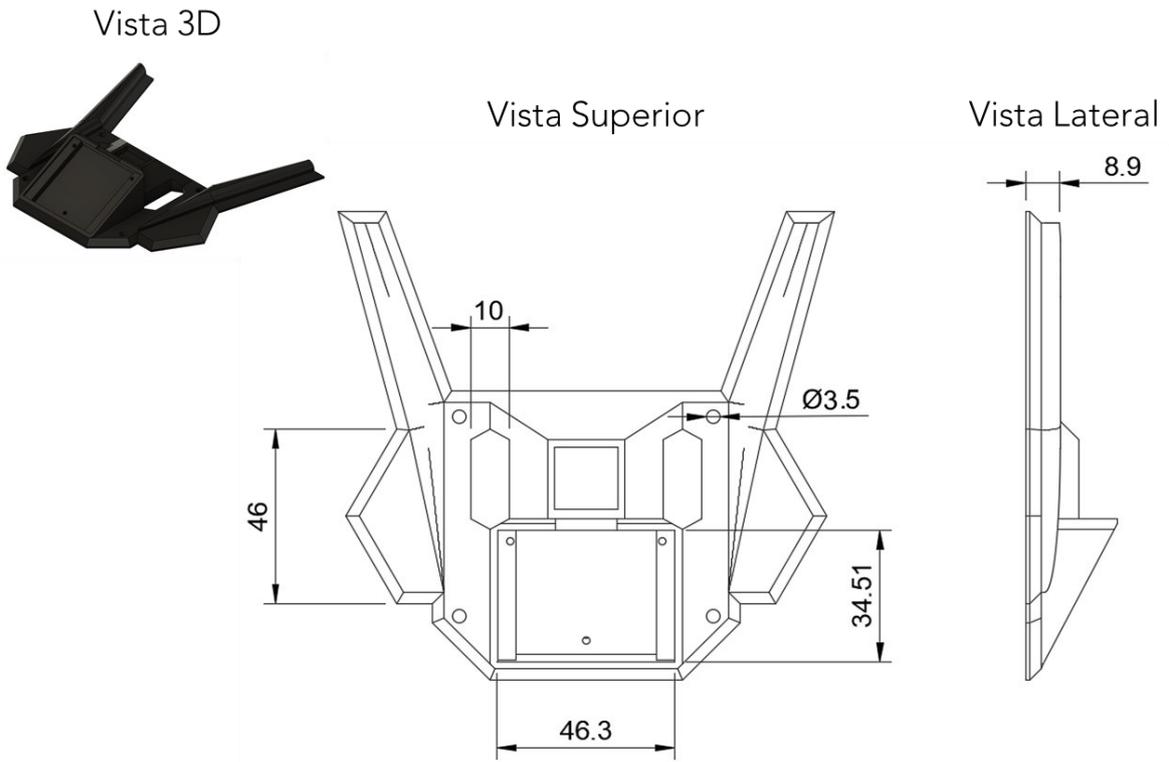
### B.1.3.3. Columna



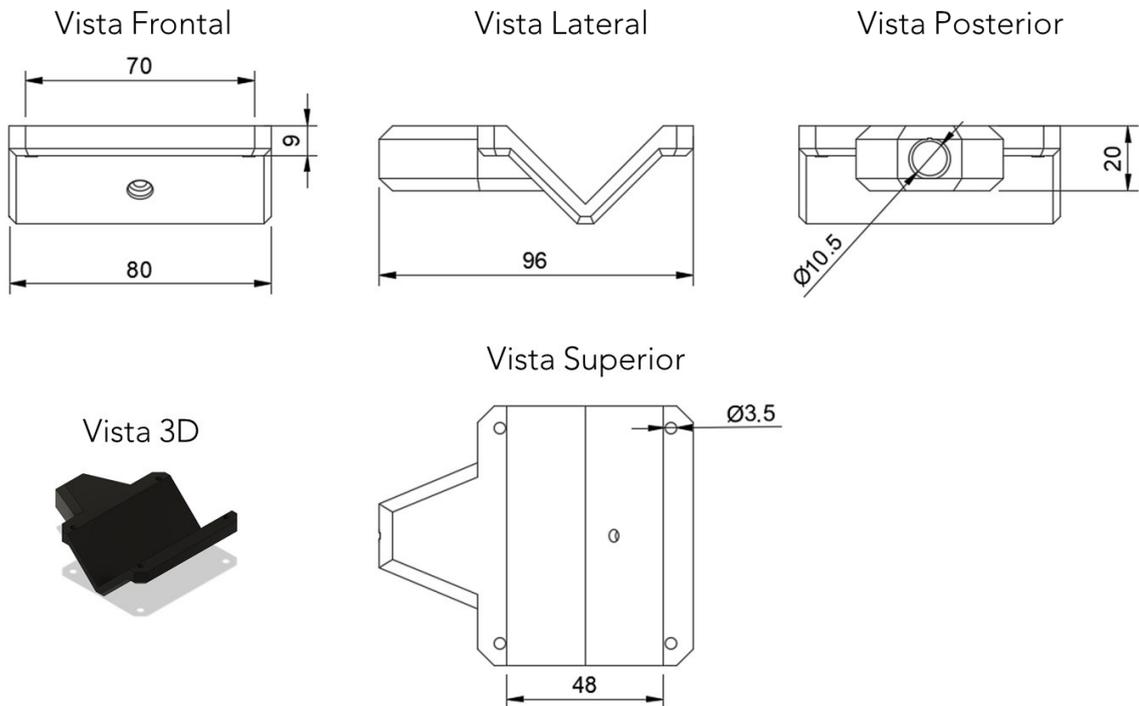
### B.1.4. SOPORTE CENTRAL



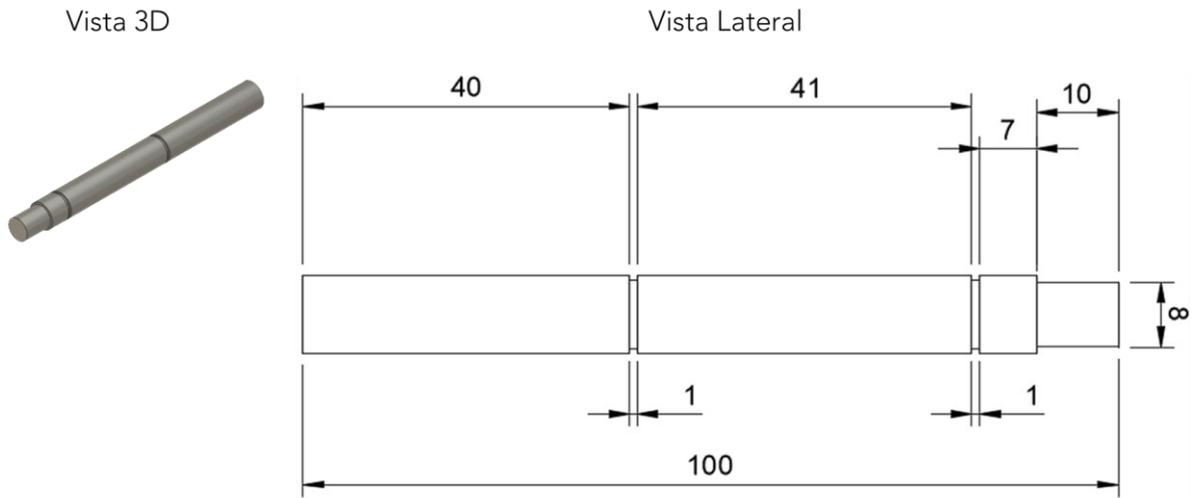
### B.1.4.1. Tapa



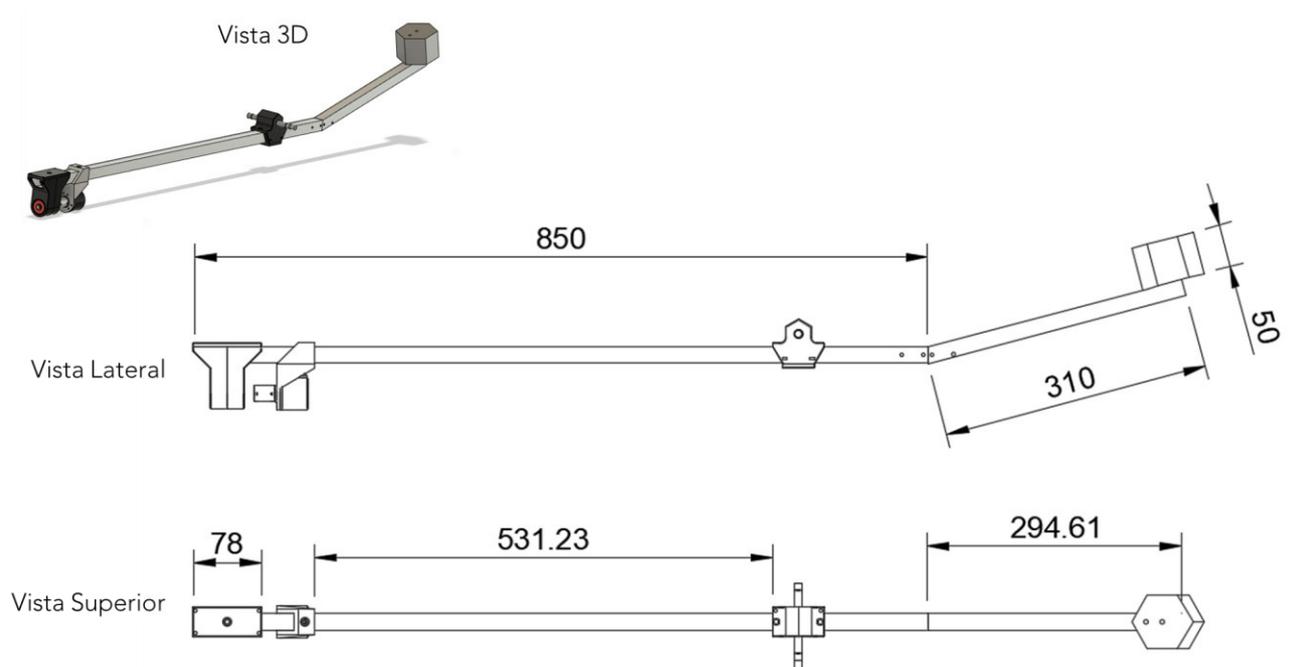
### B.1.4.2. Fondo



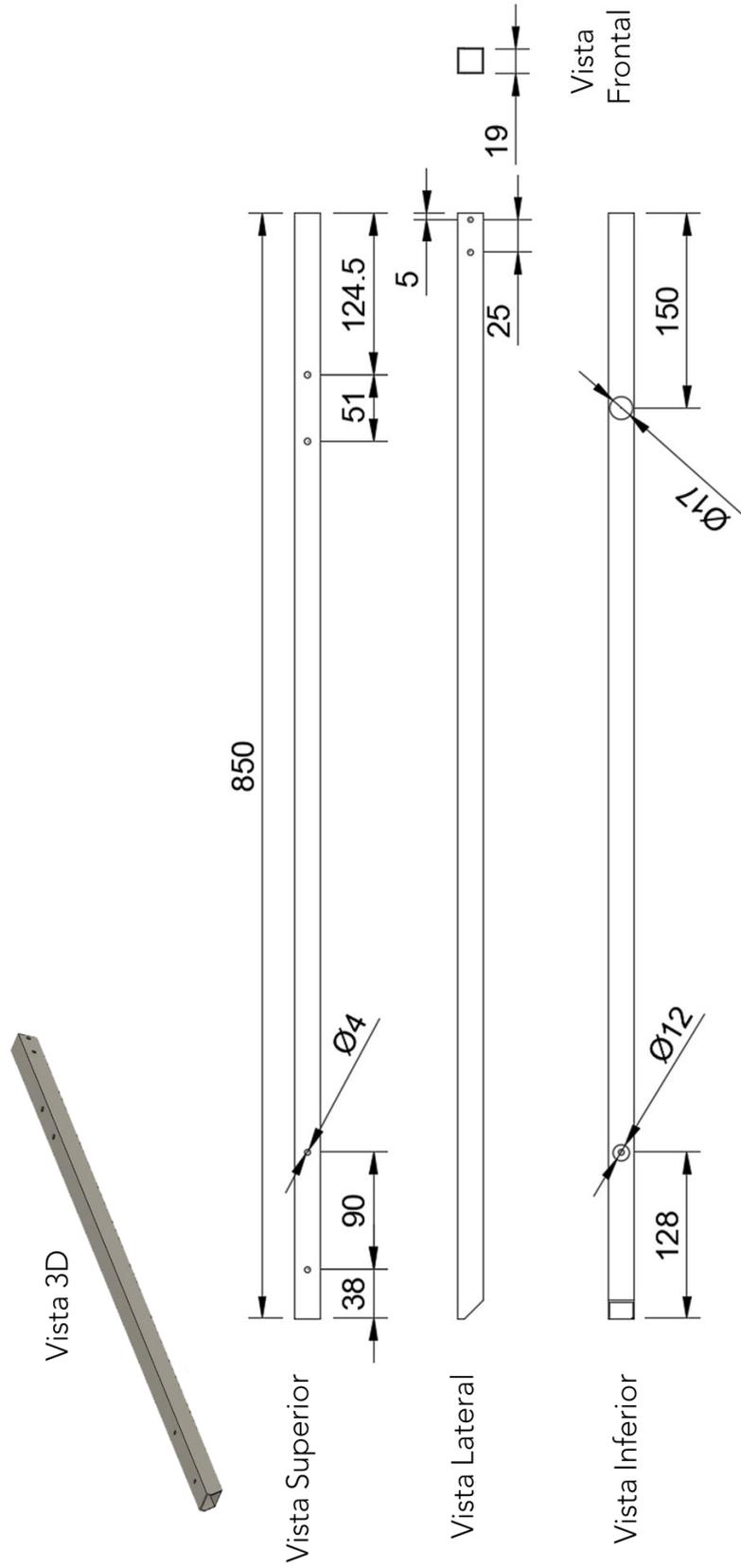
### B.1.5. EJE PITCH



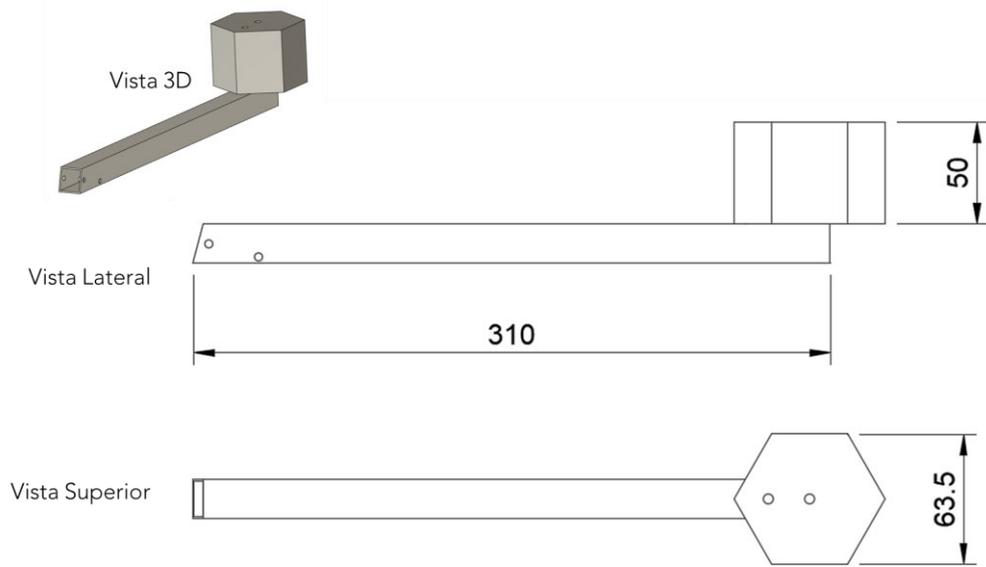
### B.2. EJE PRINCIPAL



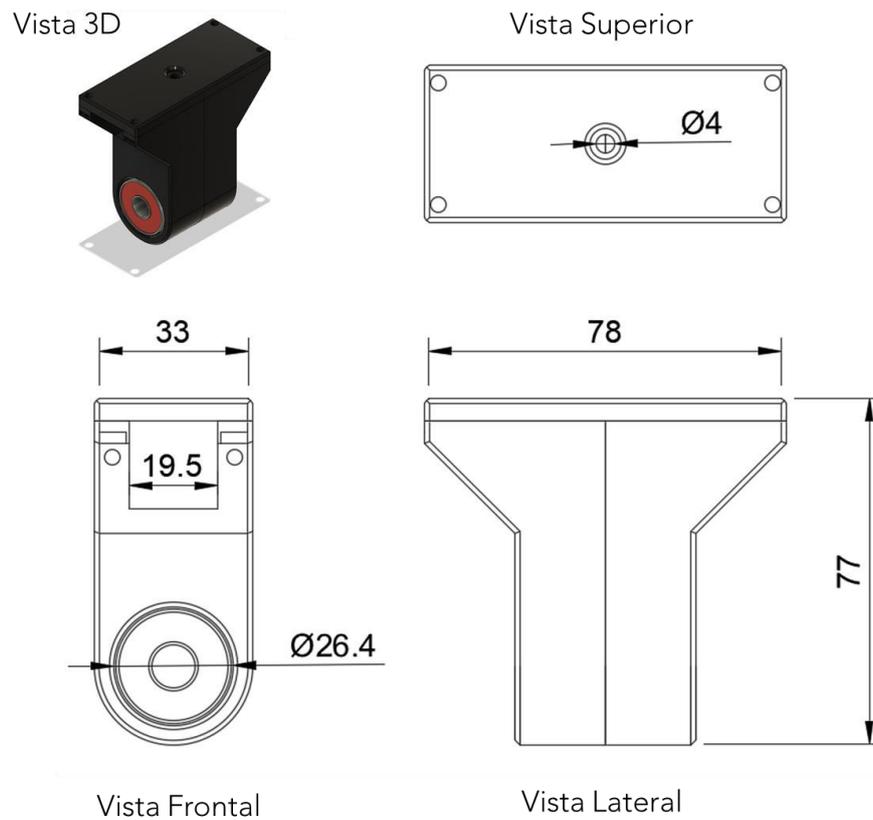
## B.2.1. BRAZO DE SOPORTE



### B.2.2. CONTRAPESO



### B.2.3. SOPORTE DEL EJE PITCH

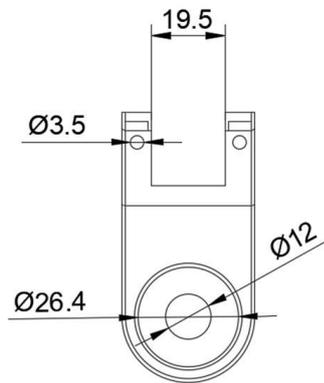


### B.2.3.1. Soporte de rodamiento

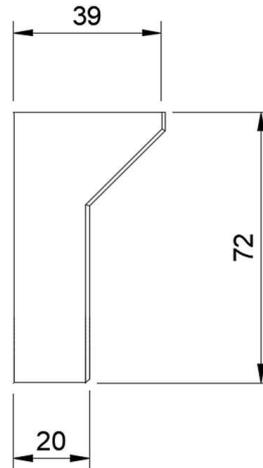
Vista 3D



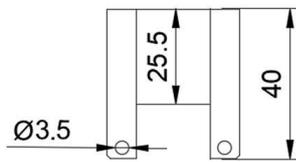
Vista Frontal



Vista Lateral



Vista Superior

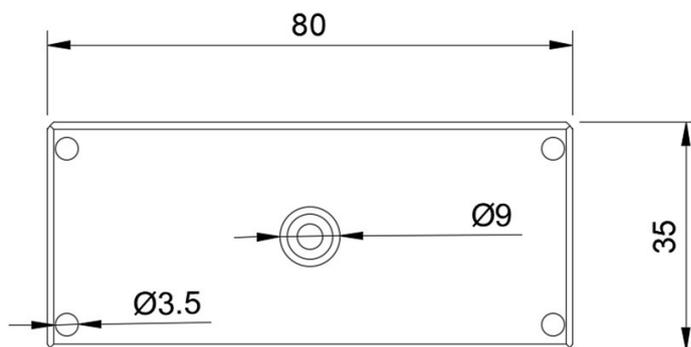


### B.2.3.2. Tapa

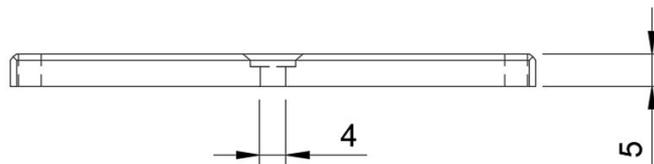
Vista 3D



Vista Superior



Vista Lateral

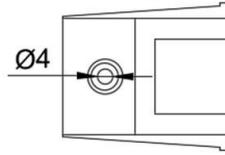


### B.2.4. SOPORTE PARA ENCODER PITCH

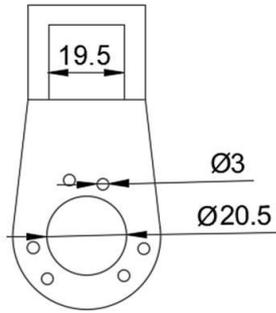
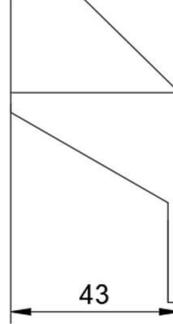
Vista 3D



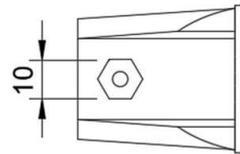
Vista Superior



Vista Lateral



Vista Frontal



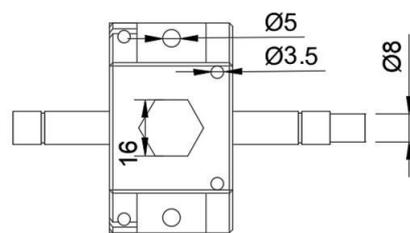
Vista Inferior

### B.2.5. SOPORTE PARA EJE ELEVATION

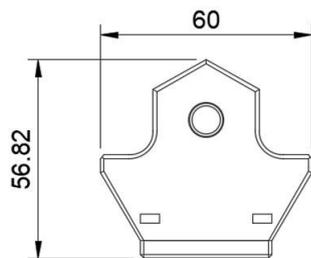
Vista 3D



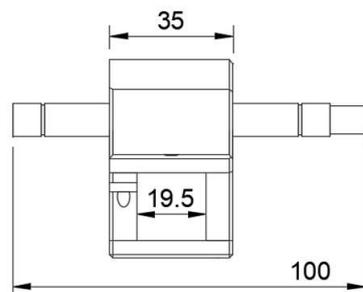
Vista Inferior



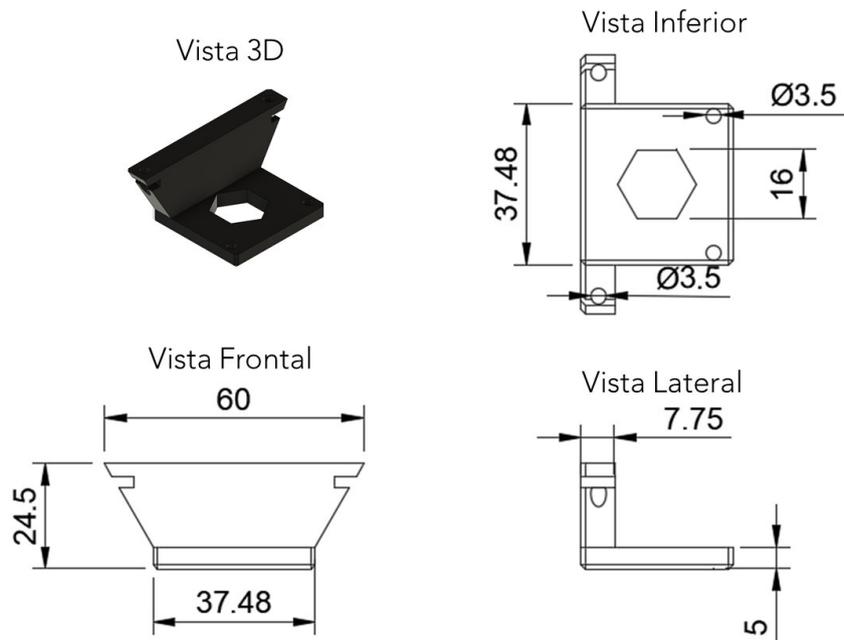
Vista Frontal



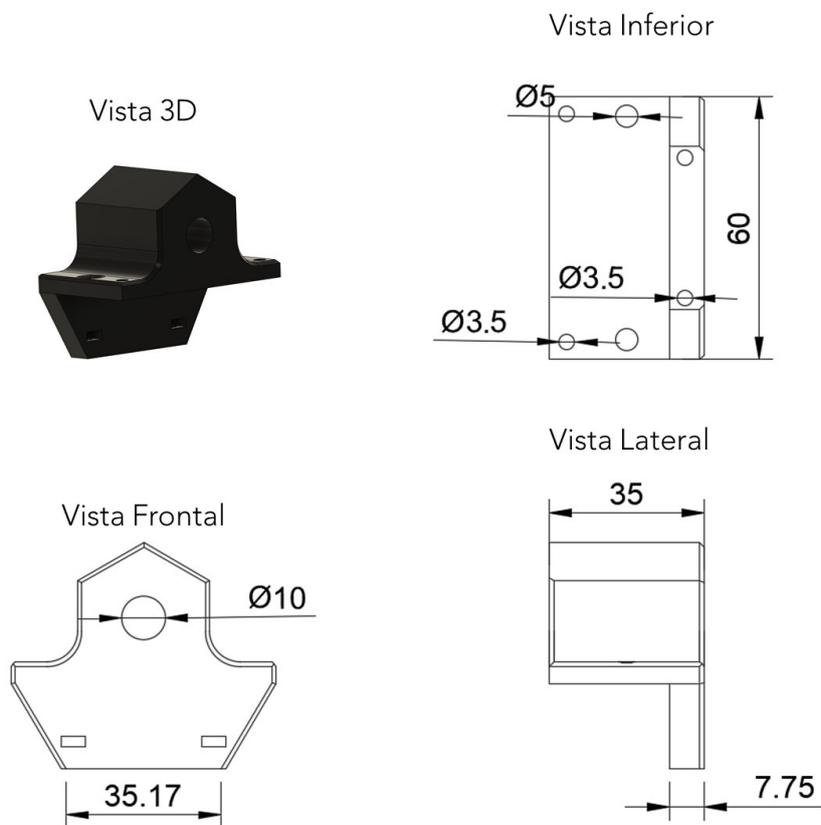
Vista Lateral



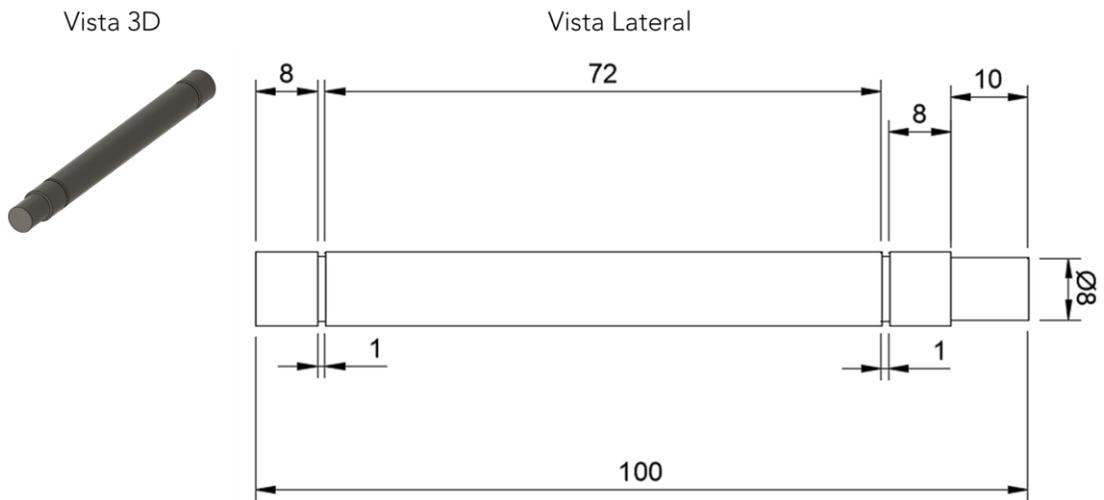
### B.2.5.1. Soporte "L" inferior



### B.2.5.2. Soporte "L" superior



### B.2.5.3. Eje elevation

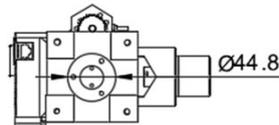


### B.3. BASE

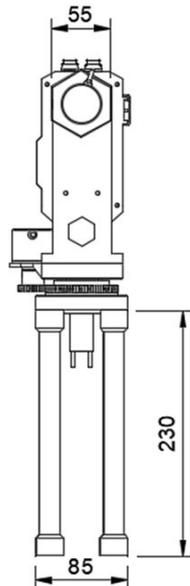
Vista 3D



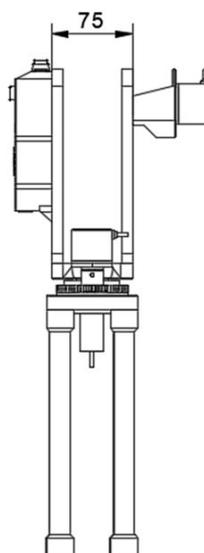
Vista Inferior



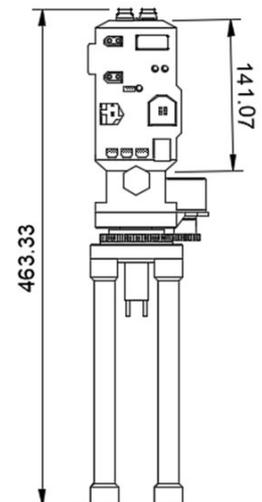
Derecha



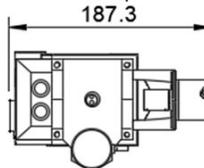
Vista Frontal



Izquierda



Vista Superior

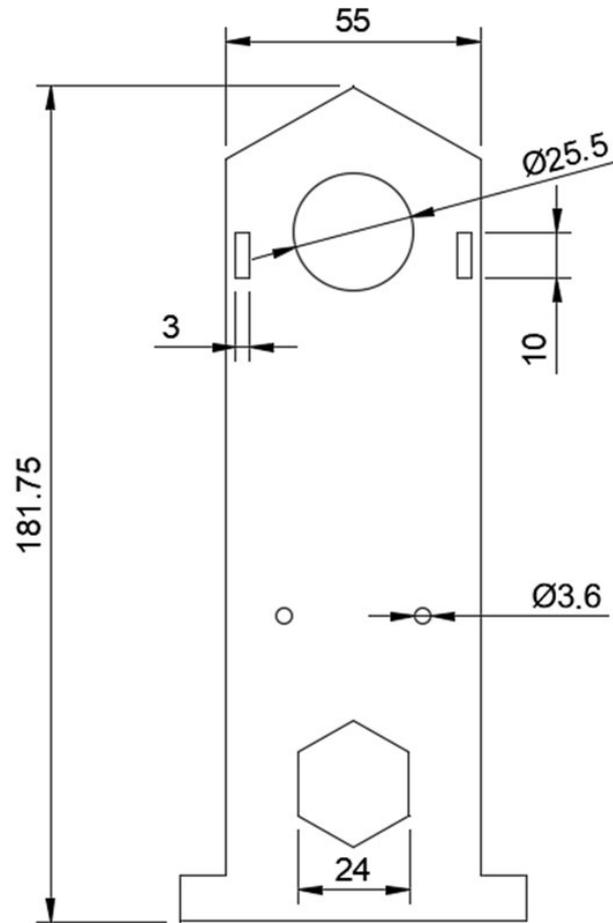


### B.3.1. BASES ELEVATION

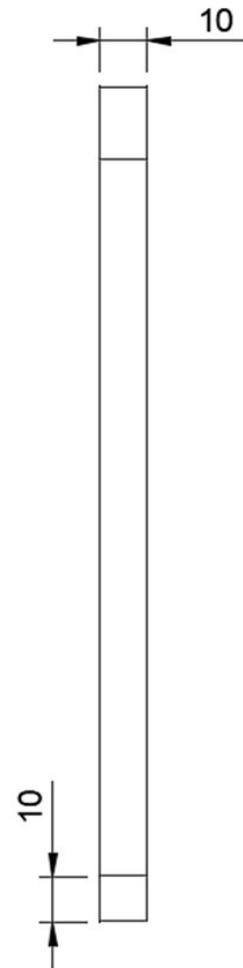
Vista 3D



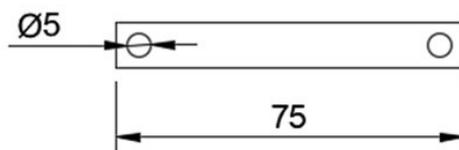
Vista Superior



Vista Lateral



Vista Inferior

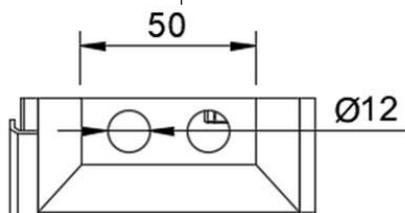


### B.3.2. CAJA DE CONTROL

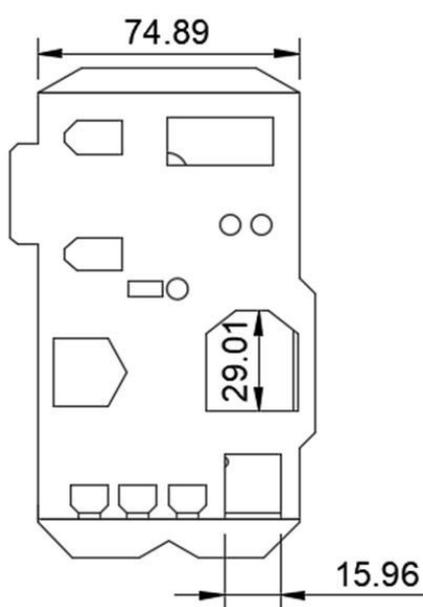
Vista 3D



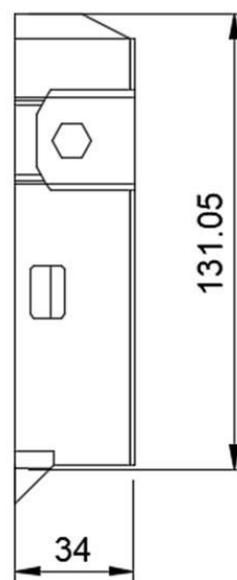
Vista Superior



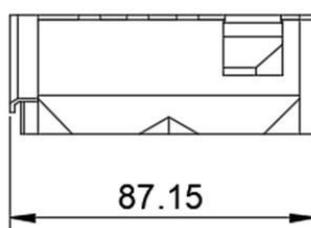
Vista Frontal



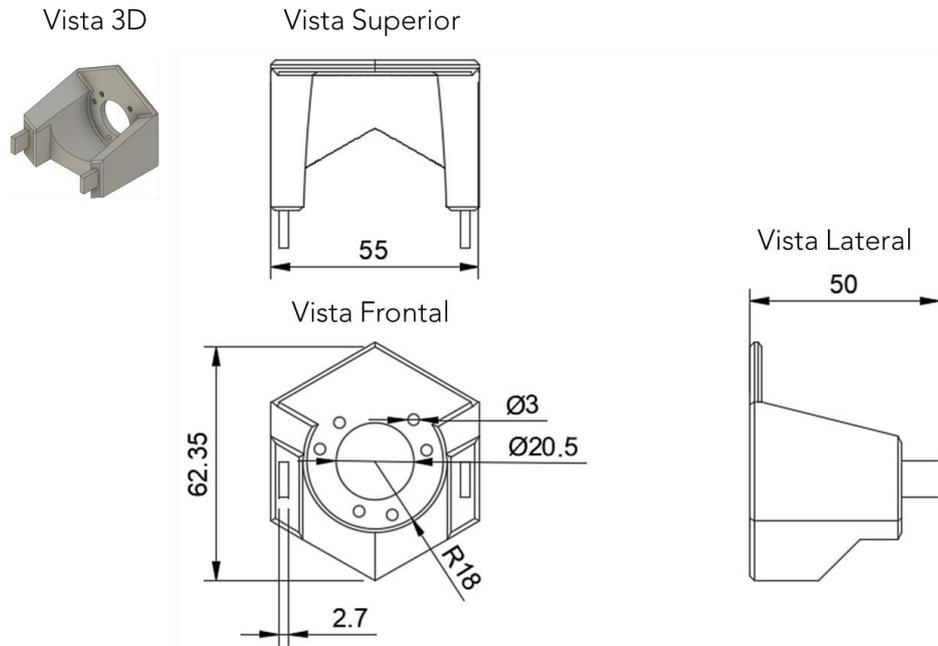
Vista Lateral



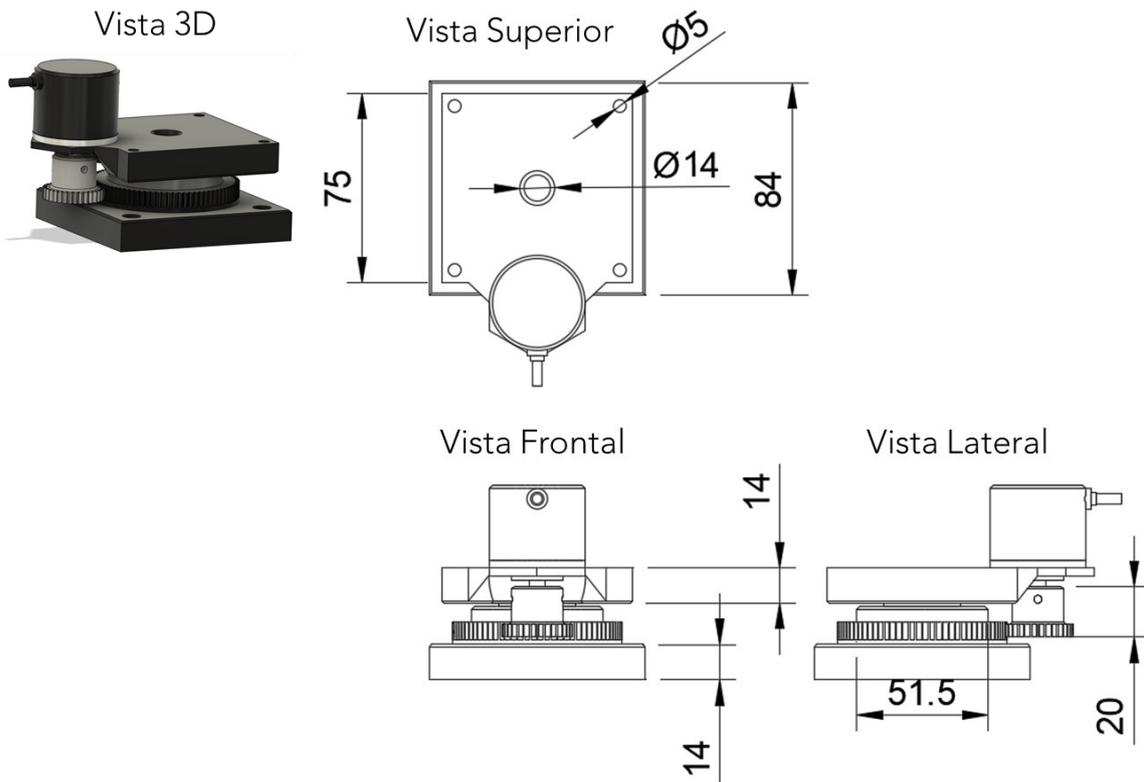
Vista Inferior



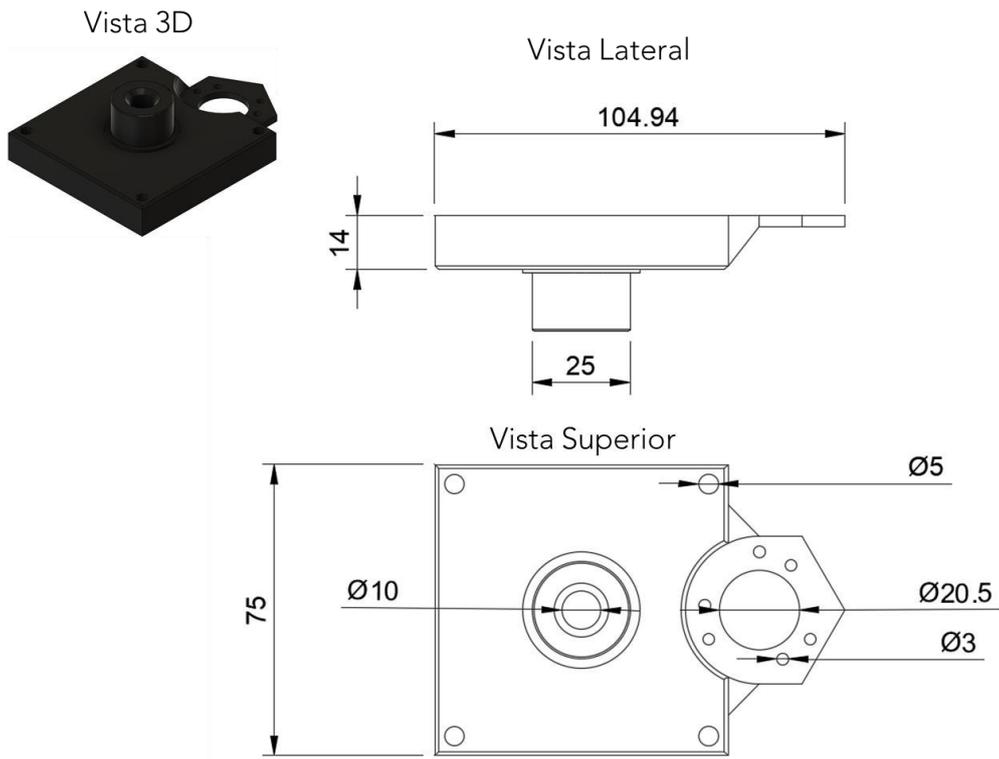
### B.3.3. SOPORTE ENCODER ELEVATION



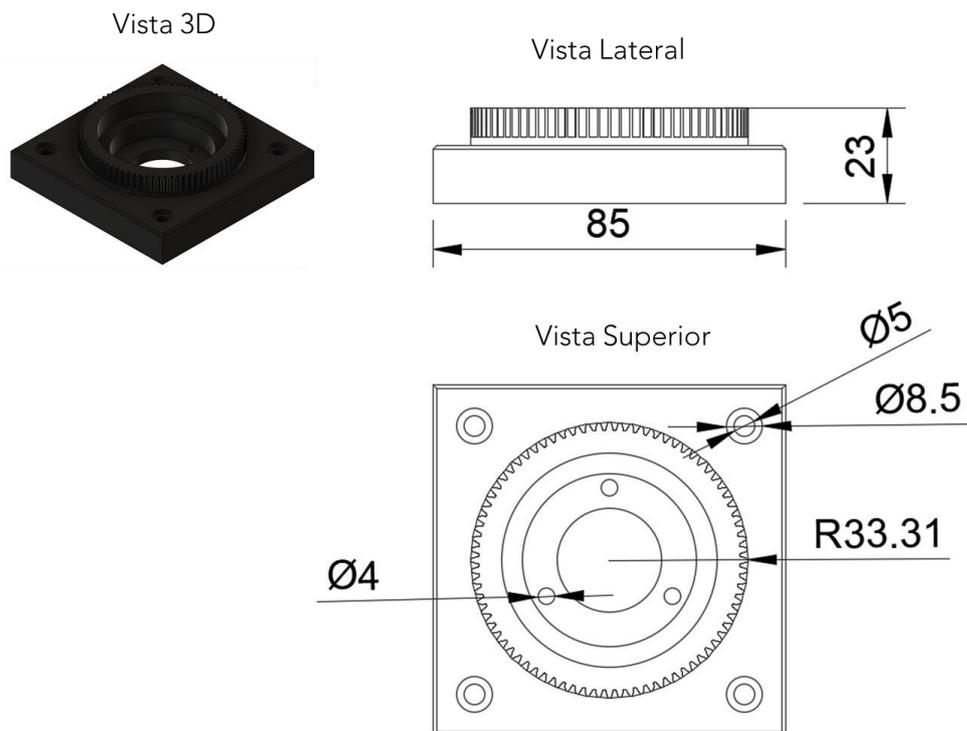
### B.3.4. BASE TRAVEL



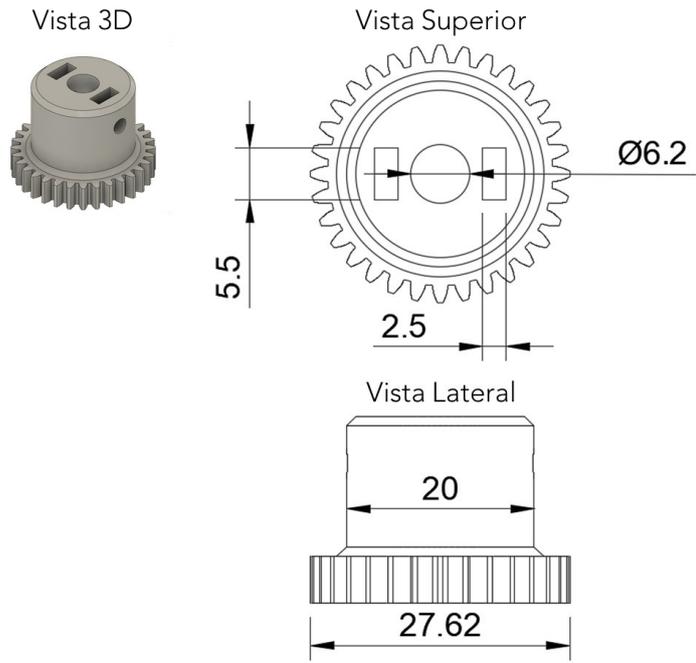
### B.3.4.1. Base superior



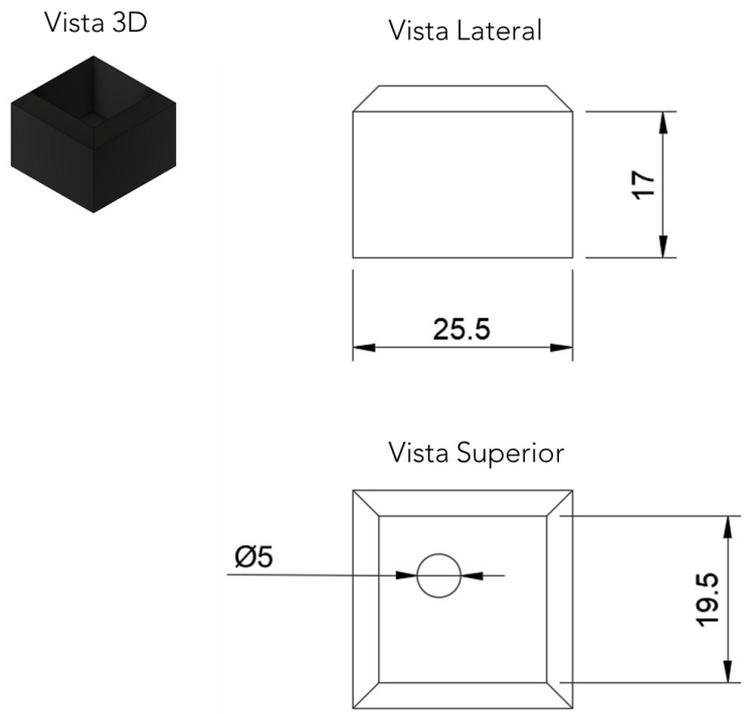
### B.3.4.2. Base Inferior



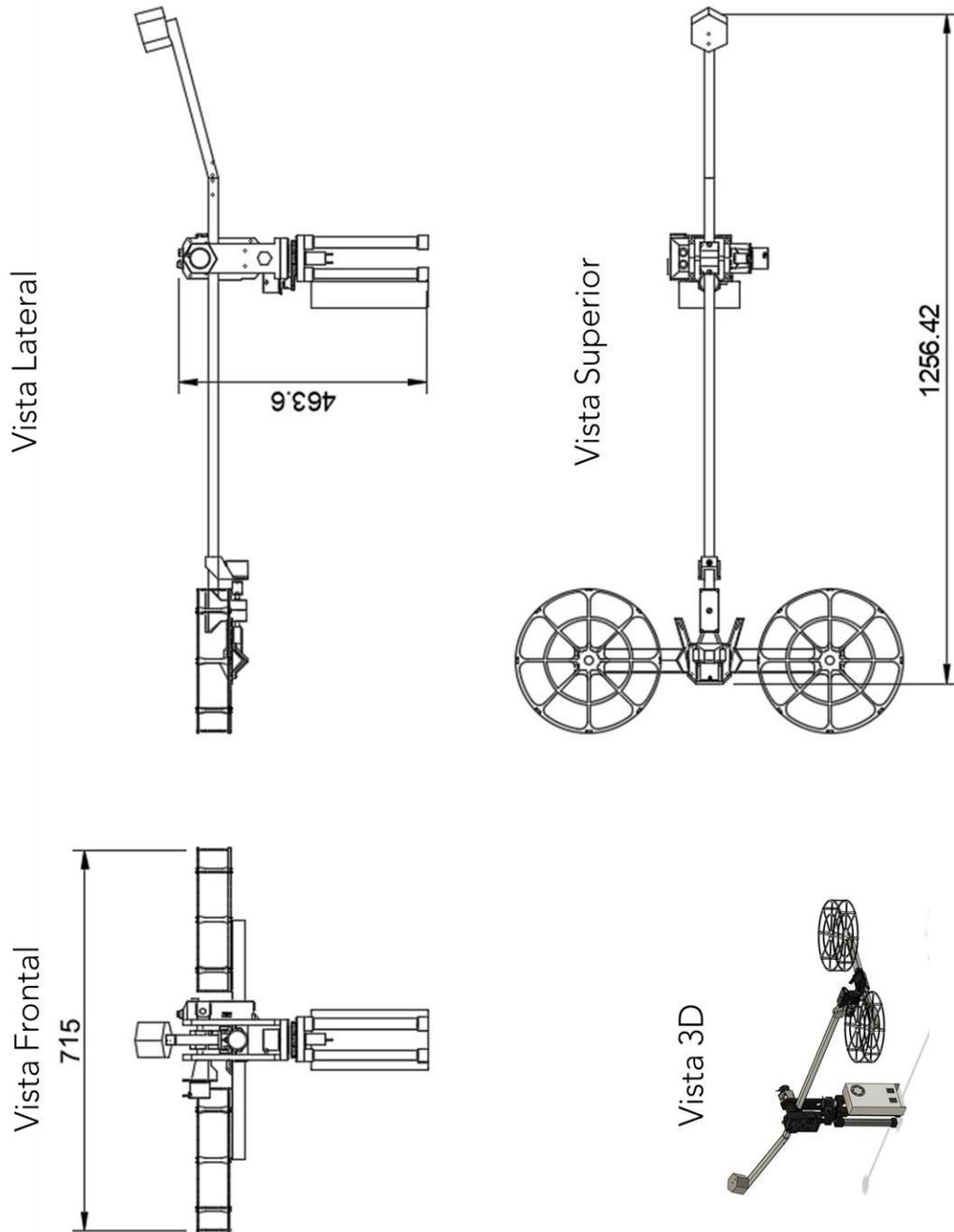
### B.3.4.3. Acople encoder



### B.3.5. APOYOS

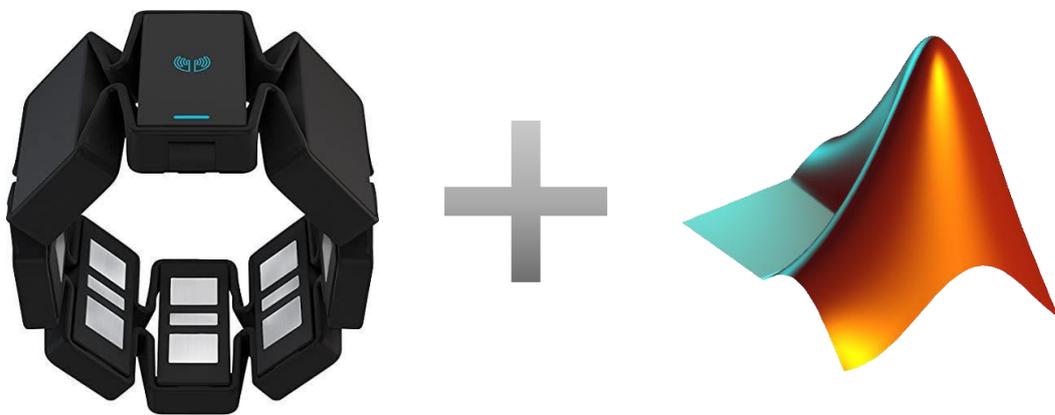


## B.4. HELICÓPTERO COMPLETO



## ANEXO C

# Guía de Instalación y Uso de Software de Reconocimiento HGR de Myo con Matlab



**Ver. 01**

Ricardo Romero

Septiembre 2020

# Instalación Software de Reconocimiento HGR

## Índice

### 1. Requerimientos previos

#### 1.1. [Descargas:](#)

##### 1.1.1. [GeneralHGR](#)

##### 1.1.2. [Mark-toma-MyoMex](#)

##### 1.1.3. [myo-sdk-win-0.9.0](#)

##### 1.1.4. [Myo Connect](#)

#### 1.2. [Add-On Matlab](#)

### 2. Instalación

#### 2.1. [Carpeta myo-sdk-win-0.9.0](#)

#### 2.2. [Agregar la carpeta al PATH de Matlab](#)

#### 2.3. [Carpeta Mark-toma-MyoMex](#)

#### 2.4. [Carpeta GeneralHGR](#)

#### 2.5. [Añadir Myo SDK al PATH de la variable de entorno de Windows](#)

#### 2.6. [Instalación Myo Mex](#)

#### 2.7. [Instalación Myo Connect](#)

### 3. Uso del Software

#### 3.1. [Estructura del software de Reconocimiento de gestos de la mano](#)

## 1. Requerimientos previos

### 1.1. **Descargas:**

En la carpeta compartida del proyecto en OneDrive se encuentran los siguientes archivos en la dirección [PIGR 19 07 > 0 PIGR 19 07 SharedFolder > SoftwareReconocimientoIA](#)

#### 1.1.1. **GeneralHGR**

Son los archivos de la aplicación desarrollada. Se tienen 3 carpetas: Classifier, Features, Figures y dos archivos '.m' que son AppCode y Main.

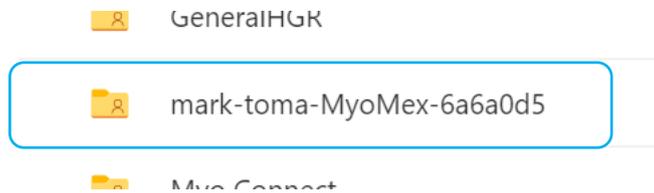
... > PIGR\_19\_07 > 0\_PIGR\_19\_07\_SharedFolder > SoftwareReconocimientoIA > Software > **GeneralHGR**

Nombre	Modificado	Modificado por	Tamaño de arch...	Compartir
Classifier	24 de julio	ANGEL LEONARDO VALDI	6 elementos	Compartido
Features	24 de julio	ANGEL LEONARDO VALDI	7 elementos	Compartido
Figures	24 de julio	ANGEL LEONARDO VALDI	6 elementos	Compartido
AppCode.m	24 de julio	ANGEL LEONARDO VALDI	266 bytes	Compartido
Main.m	24 de julio	ANGEL LEONARDO VALDI	1,51 KB	Compartido

### 1.1.2. Mark-toma-MyoMex

Es la aplicación que permite acceder a las funciones de Myo mediante aplicaciones '.m' desarrolladas en Matlab. Fue implementada por Mark Tomaszewski. Para descargar se tienen 2 opciones:

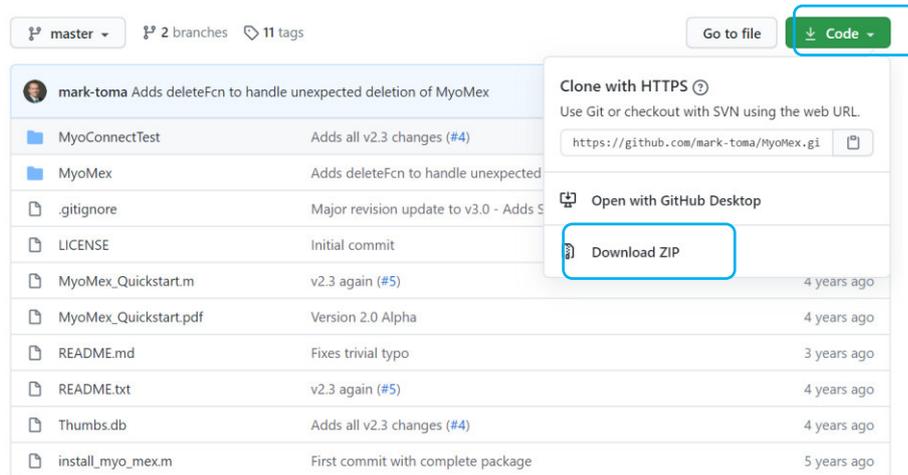
- Desde la carpeta compartida:



- o directamente desde el repositorio en Github:

<https://github.com/mark-toma/MyoMex>

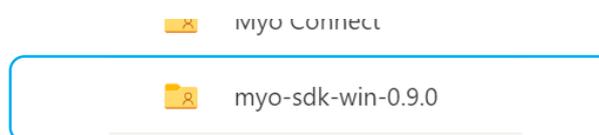
Dar click en el botón Code y luego Download ZIP



### 1.1.3. myo-sdk-win-0.9.0

Es el paquete para desarrollo de software para poder hacer uso de las funciones de Myo mediante aplicaciones en Matlab, Python, etc. Para descargar también se tienen 2 posibilidades:

- Desde la carpeta compartida:



- o directamente desde la página de soporte de Thalmic Labs (North):

<https://support.getmyo.com/hc/en-us/articles/360018409792-Myo-Connect-SDK-and-firmware-downloads>

Dar click en el enlace de descarga para Windows.

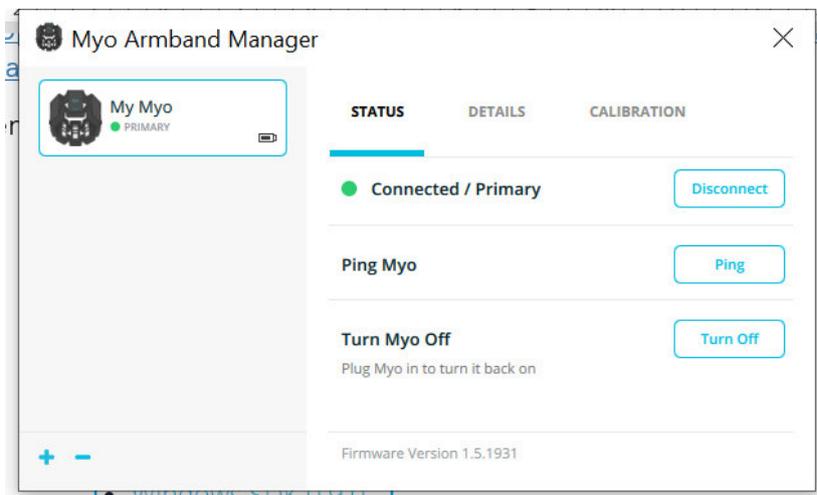
### Myo SDK

\*The Unity package is included in Windows and macOS SDKs

- [Android SDK 0.10.0](#)
- [iOS SDK 0.5.2](#)
- [macOS SDK 0.9.0](#)
- [Windows SDK 0.9.0](#)

#### **1.1.4. Myo Connect**

Es un software propietario que permite la conexión del sensor a la PC mediante el receptor bluetooth. Además de permitir conectar el dispositivo se tienen otras funciones como indicador de nivel de batería, estado de conexión, apagar el dispositivo, etc.



**Nota:** este programa no es parte de la aplicación como tal en la que se está trabajando, pero es importante para conectar el dispositivo y conocer el estado del mismo.

Este programa se encuentra disponible de manera gratuita en el siguiente enlace:

<https://support.getmyo.com/hc/en-us/articles/360018409792-Myo-Connect-SDK-and-firmware-downloads>

Para el caso de Windows se debe dar click en el vínculo que se muestra a continuación. La descarga iniciará automáticamente.

## Myo Connect

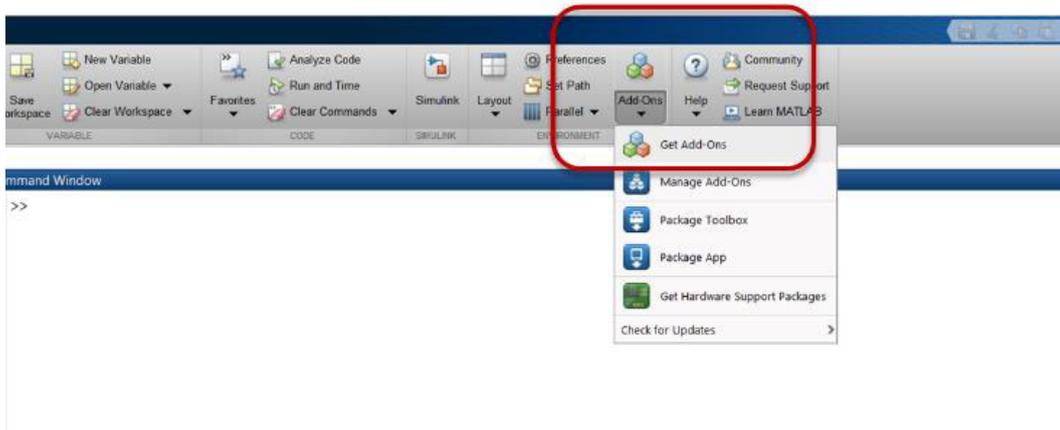
- Myo Connect for Windows 1.0.1
  - Myo Connect for macOS 1.0.1
- 

### **1.2. Add-On Matlab**

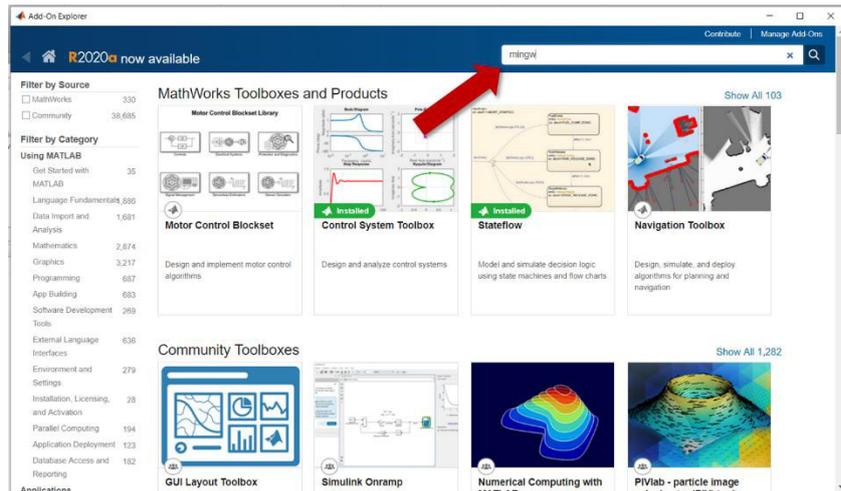
Además de todas las descargas que se mencionaron anteriormente, es necesario instalar el siguiente complemento para compilador de C/C++ en Matlab:

#### **MATLAB Support for MinGW-w64 C/C++ Compiler**

1. Dar click en el botón Add-Ons de la barra de inicio de Matlab y seleccionar la opción Get Add-Ons



2. Se abrirá una ventana de explorador de complementos. Una vez ahí se ingresa 'mingw' en la barra de búsqueda



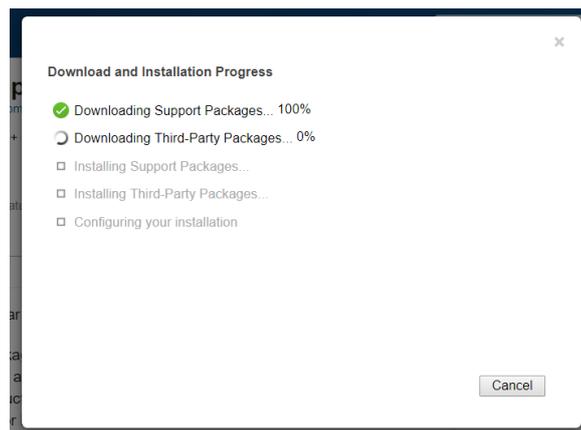
3. Seleccionar la opción de *MATLAB Support for MinGW-w64 C/C++ Compiler*



4. Dar click en el botón 'Install' y seleccionar la opción de instalar. En este punto se pedirá el ingreso de email y contraseña con las que se encuentre registrada la cuenta de Matlab.

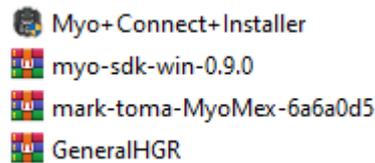


5. Luego de aceptar el acuerdo de licencia se da click en el botón Next para iniciar la instalación y se desplegará una nueva ventana con barras de estado de descarga:



Luego de esto la instalación estará completa y el complemento se encontrará listo para usar.

Además del complemento que se acaba de mencionar, la carpeta de descargas deberá contener los siguientes archivos:



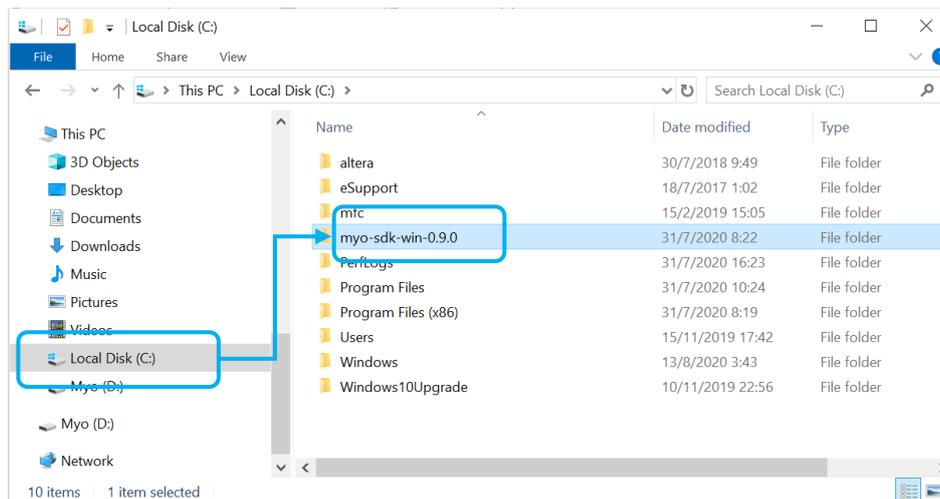
Luego de descomprimir los archivos se puede seguir al siguiente paso.

## 2. Instalación

Una vez descargados todos los archivos mencionados se deberá realizar lo siguiente:

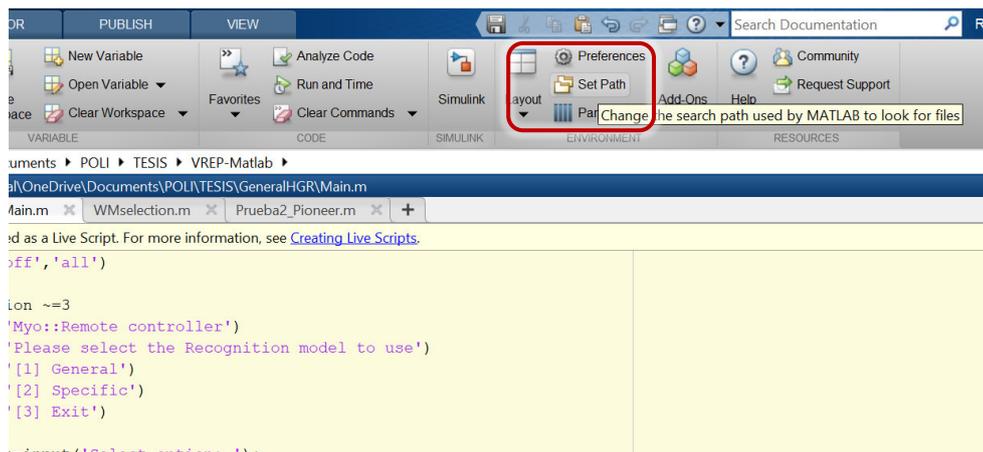
### 2.1. Carpeta myo-sdk-win-0.9.0

Copiar esta carpeta en el disco local C:

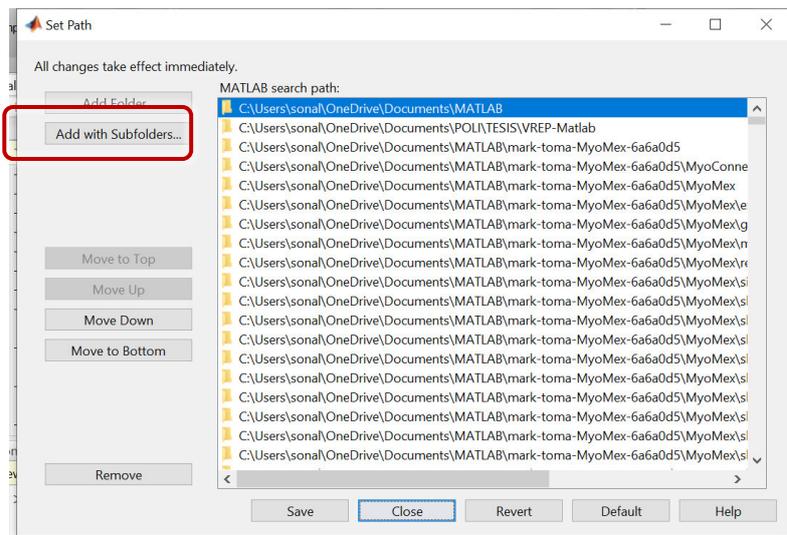


### 2.2. Agregar la carpeta al PATH de Matlab

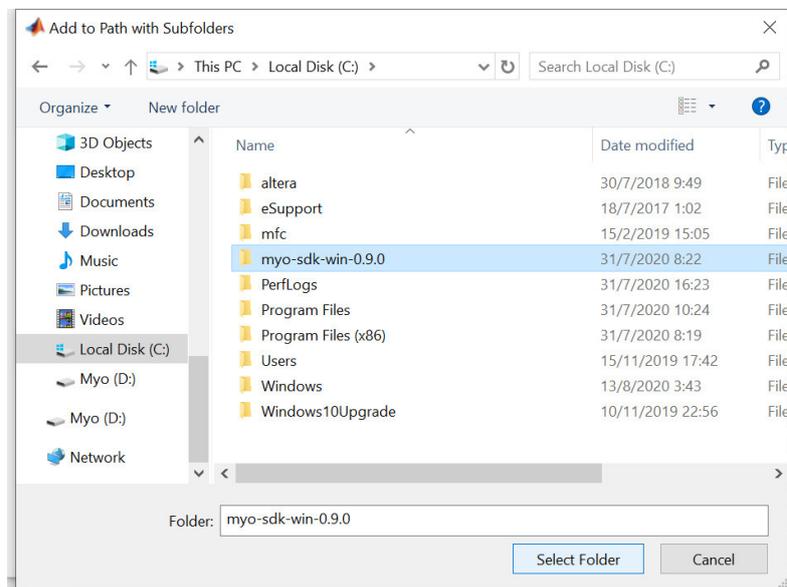
- Dar click en Set Path en la barra de inicio.



- Se despegará una ventana. En esta se debe dar click en 'Add with Subfolders'



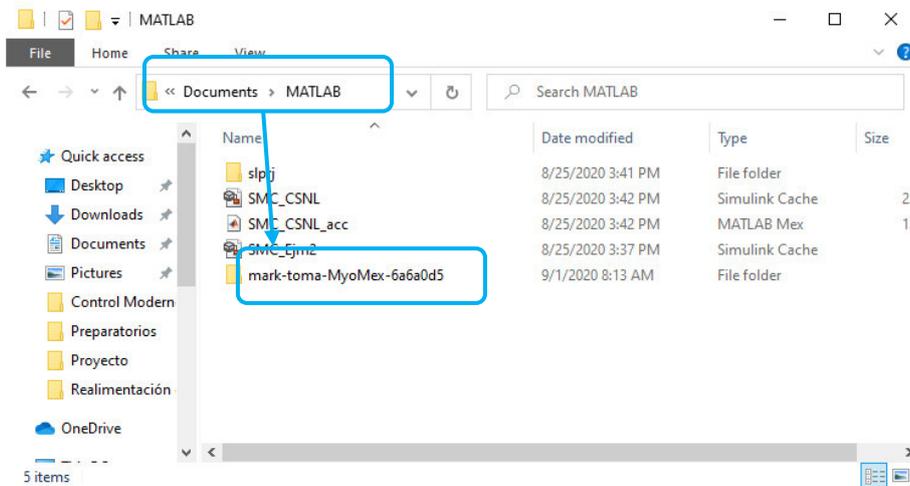
- Se busca la carpeta correspondiente que debe estar en la dirección "C:\myo-sdk-win-0.9.0". Luego dar click en Select Folder.



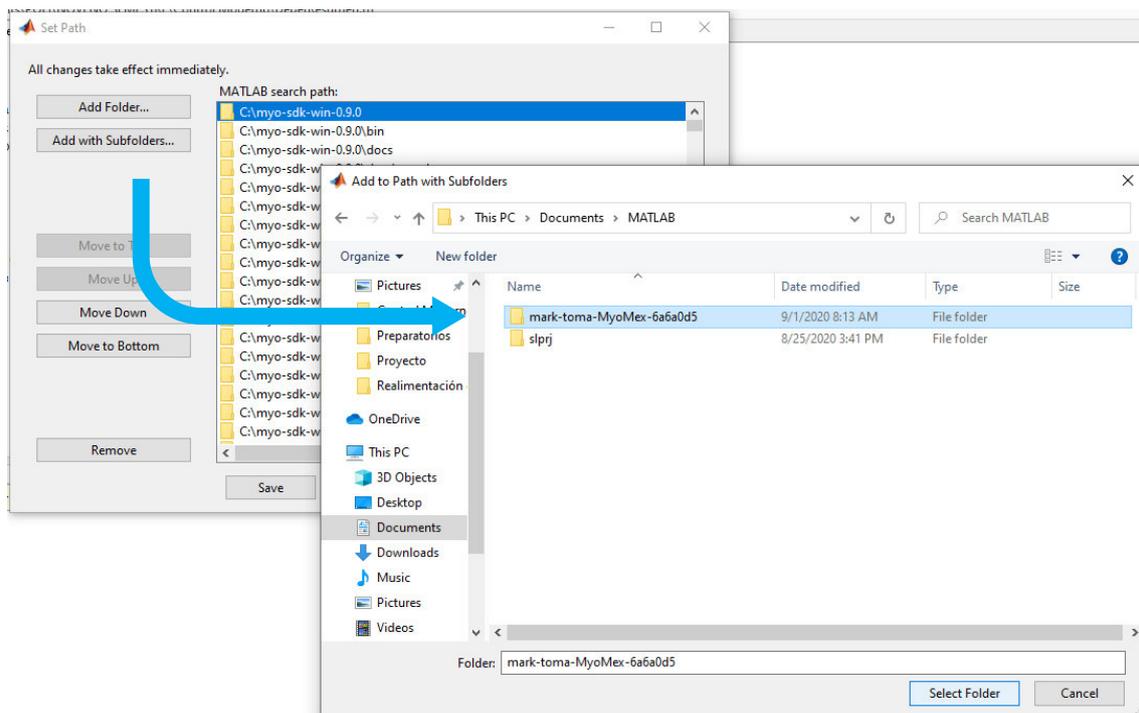
- Dar click en los botones **Save** y luego **Close**.

### 2.3. Carpeta Mark-toma-MyoMex

Esta carpeta debe ser añadida a la carpeta raíz de Matlab, que se encuentra en la dirección "C:\Users\username\Documents\MATLAB"



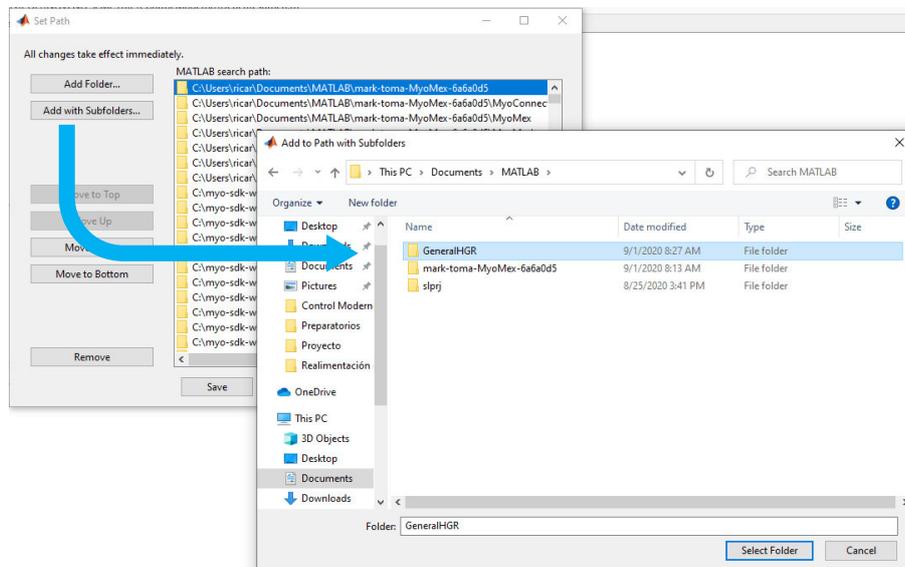
- De nuevo, esta carpeta se debe añadir al PATH de Matlab. Para ello se deberá realizar el procedimiento indicado anteriormente, pero en este caso se debe añadir el directorio de "C:\Users\username\Documents\MATLAB\mark-toma-MyoMex-6a6a0d5"



- Dar click en los botones **Save** y luego **Close**.

## 2.4. Carpeta GeneralHGR

Esta carpeta deberá ser añadida al PATH de Matlab al igual que los casos anteriores (de preferencia también copiar esta carpeta en el directorio de Matlab). Para esto se debe seguir el procedimiento indicado antes.



- Dar click en los botones **Save** y luego **Close**.

## 2.5. Añadir Myo SDK al PATH de la variable de entorno de Windows

Al correr los programas es posible encontrarse con el siguiente error:

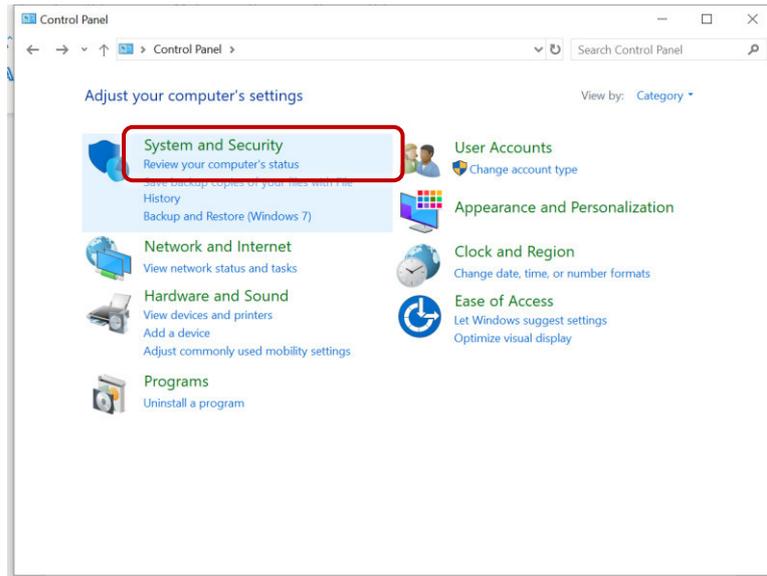
```
Error using MyoMex (line XXX)
```

```
MEX-file 'myo_mex' failed to initialize with error:
```

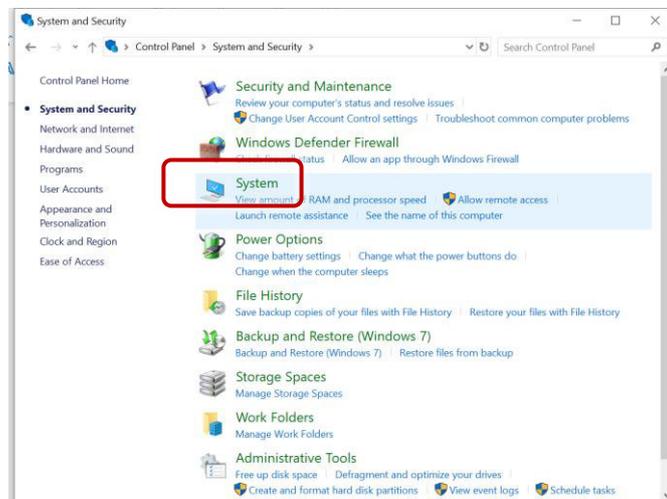
```
'Invalid MEX-file '<absolute path to  
MyoMex>\MyoMex\myo_mex\myo_mex.mexw64': Can't find specified module.'
```

Para solucionar este error se deben agregar el directorio bin del SDK de Myo al PATH de la variable de entorno en Windows (PATH Environment Variable). Para lo cual se realiza lo siguiente:

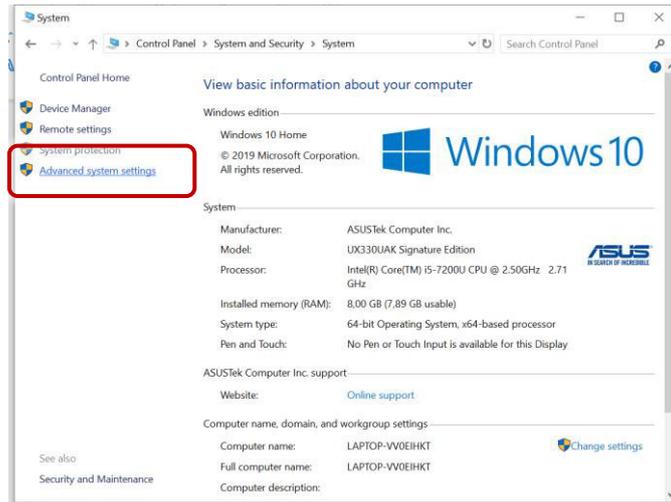
1. Abrir el panel de control y seleccionar la opción Sistema y Seguridad



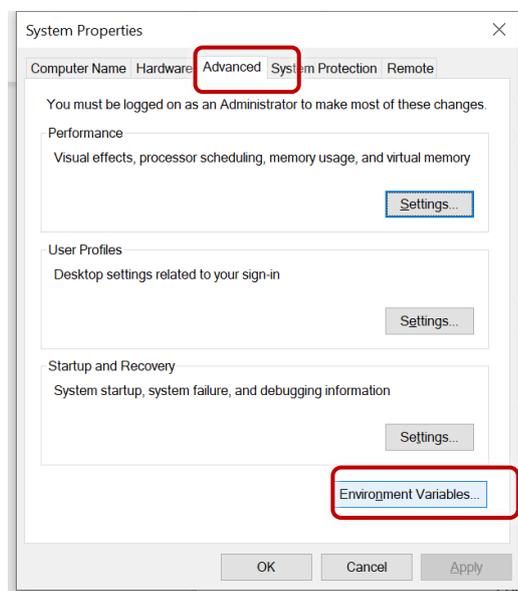
2. Una vez ahí seleccionamos la opción Sistema



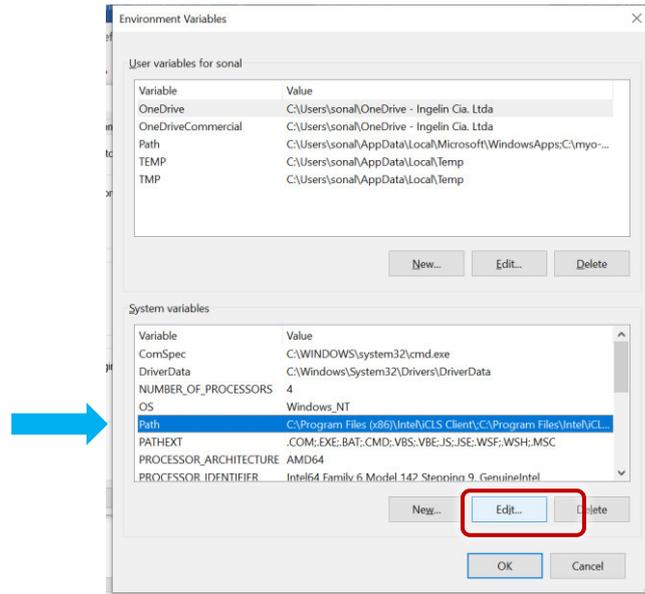
3. Una vez aquí se ingresa a la parte de Ajustes avanzados de sistema:



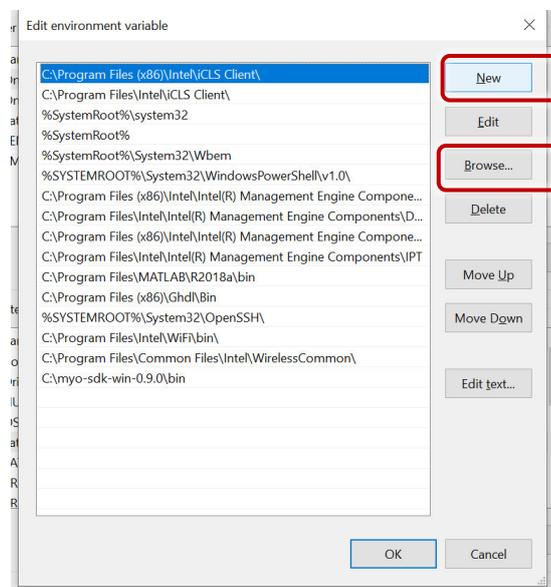
4. Se abrirá una nueva ventana y bajo la pestaña de ajustes Avanzados se debe dar click en el botón Variables de Entorno (Environment Variables)



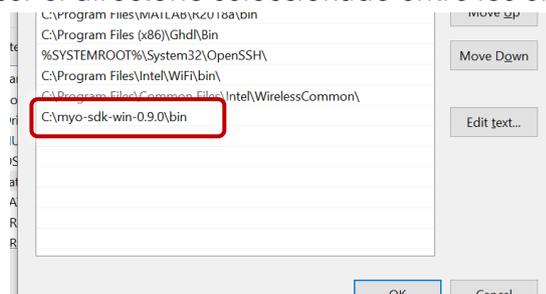
5. Una vez aquí seleccionamos la variable "Path" y luego el botón de editar



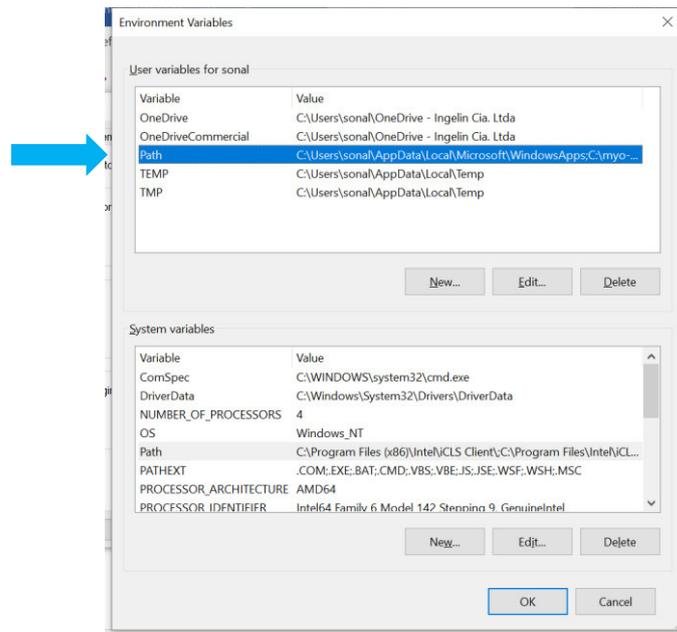
6. Dar click en el botón "Nuevo" y luego Navegar o Browse



7. Navegar en el buscador hasta encontrar el directorio **bin** del SDK de Myo que se debe encontrar en la dirección "**C:\myo-sdk-win-0.9.0\bin**". Una vez realizado esto deberá aparecer el directorio seleccionado entre los elementos del Path:



8. Repetir los pasos desde el 5 hasta el 7 para la variable Path de usuario (en la parte superior de la ventana de Variables de Entorno):



## 2.6. Instalación Myo Mex

Para instalar el software de la carpeta Myo Mex se debe ingresar a Matlab e ingresar el siguiente comando en el Command Window:

```
>> install_myo_mex
```

Este proceso será muy rápido y no aparecerá ningún aviso. De todos modos, con eso es suficiente para instalar el software.

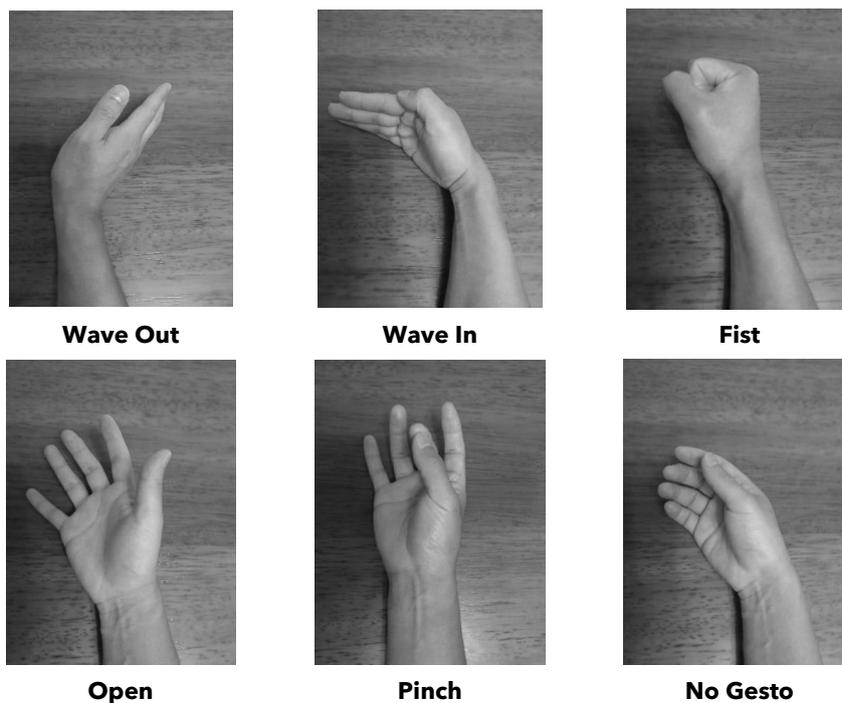
## 2.7. Instalación Myo Connect

- Abrir el instalador de Myo Connect.
- Aceptar las licencias, seleccionar el directorio de destino y seleccionar la opción de "Install".
- Con esto se finalizará la instalación y posteriormente se abre una guía de uso del dispositivo.
- Con la guía de uso ofrecida por el fabricante se podrá inicializar adecuadamente al Myo.

Luego de realizar todos estos pasos será necesario **reiniciar Matlab** para que el sistema funcione adecuadamente.

## 3. Uso del Software

El software desarrollado permite la detección de 6 gestos de la mano que se muestran a continuación:

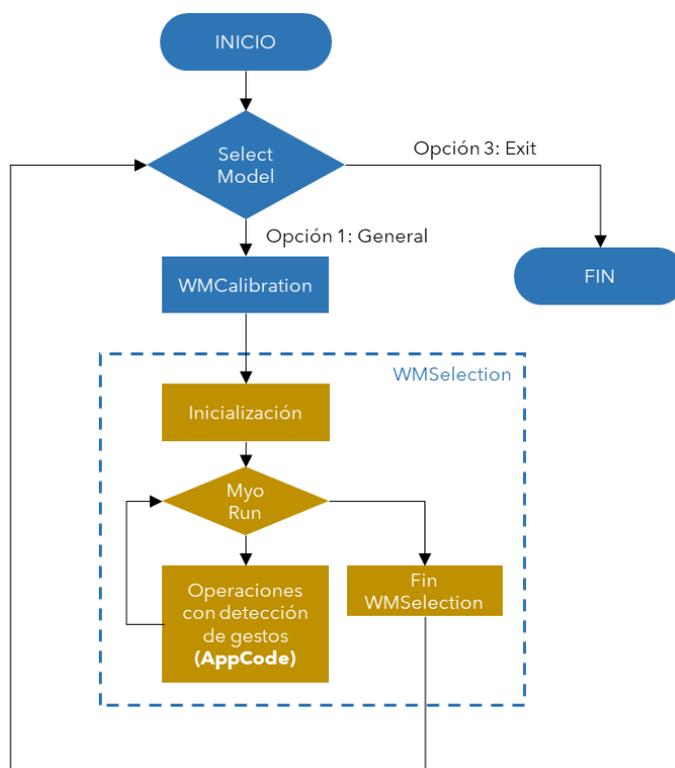


### 3.1. Estructura del software de Reconocimiento de gestos de la mano

La aplicación de reconocimiento contiene dos archivos '.m' principales, que son:

- Main.m
- AppCode.m

El programa Main.m tiene la siguiente estructura:



Para poder realizar alguna aplicación mediante el uso del sistema de reconocimiento se debe escribir el programa correspondiente en el archivo AppCode.m. Sin embargo, como se puede observar en el gráfico, este archivo corresponde a una función de Matlab que se encuentra dentro de un lazo while de WMSelection. En caso de que la aplicación realizada requiera un código de inicialización (por ejemplo, establecer comunicación con un puerto serial o con otro programa como V-REP), se deberá modificar el archivo WMSelection según sea necesario.

Para la identificación de gestos se tienen dos variables que son **ges** y **newGesture**. Estas variables pueden tomar valores que se muestran en la siguiente tabla donde se muestra la correspondencia con cada gesto:

Gesto						
Variable						
<b>newGesture</b>	waveOut	waveIn	fist	open	pinch	noGesture
<b>ges</b>	1	2	3	4	5	6

Entonces, por ejemplo, si se desea realizar una acción mediante la identificación del gesto fist se tiene las siguientes posibilidades para la sintaxis:

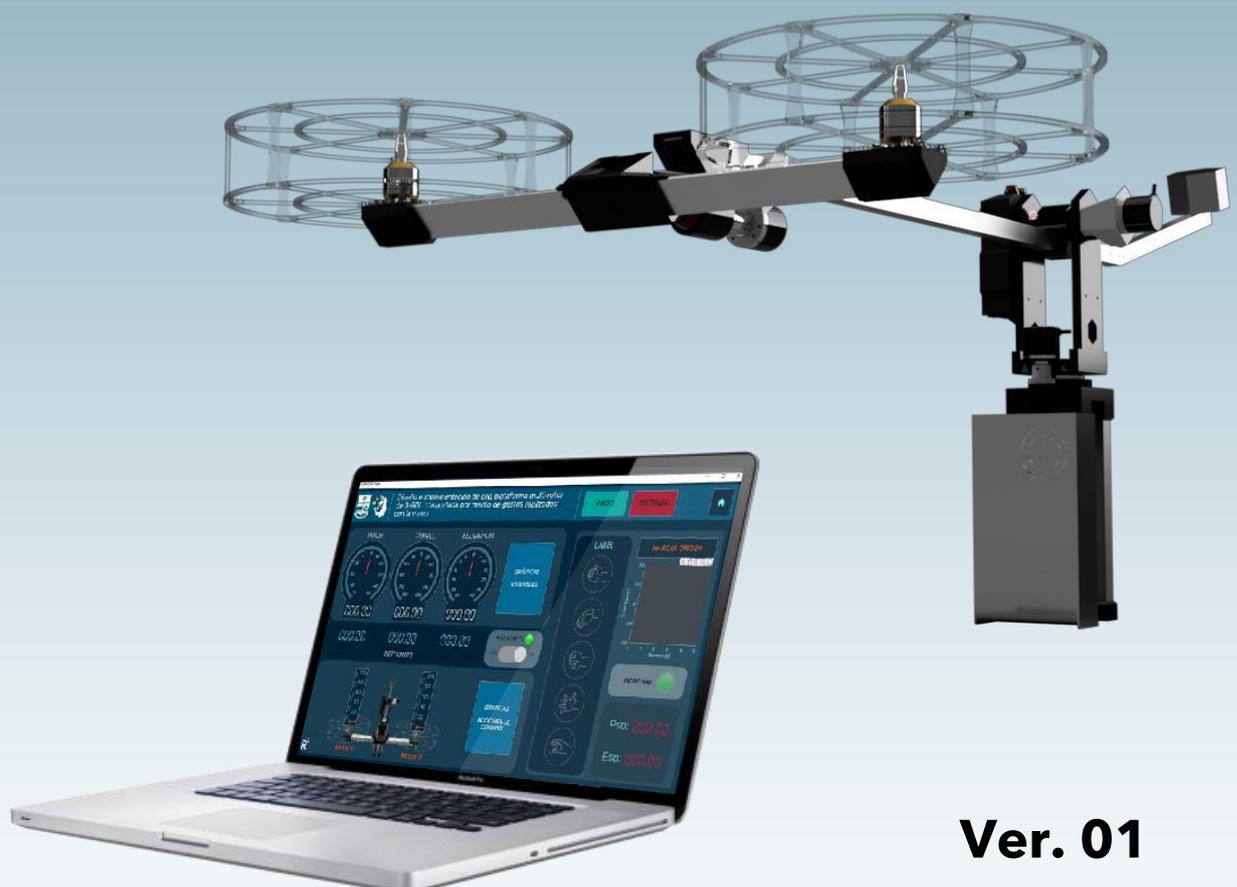
Usando <b>newGesture</b>	Usando <b>ges</b>
<pre> if newGesture == 'fist'     %accion end </pre>	<pre> if ges == 3     %accion end </pre>

## ANEXO D



# Manual de Usuario

Información y uso del prototipo y la interfaz gráfica.



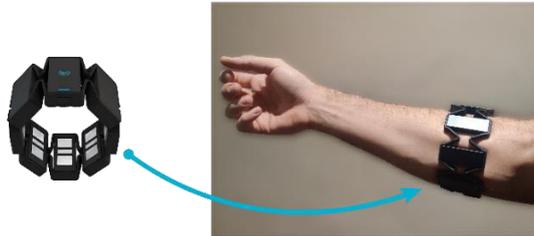
**Ver. 01**

Ricardo Romero

Octubre 2021

# 1. Activación del sistema de HGR

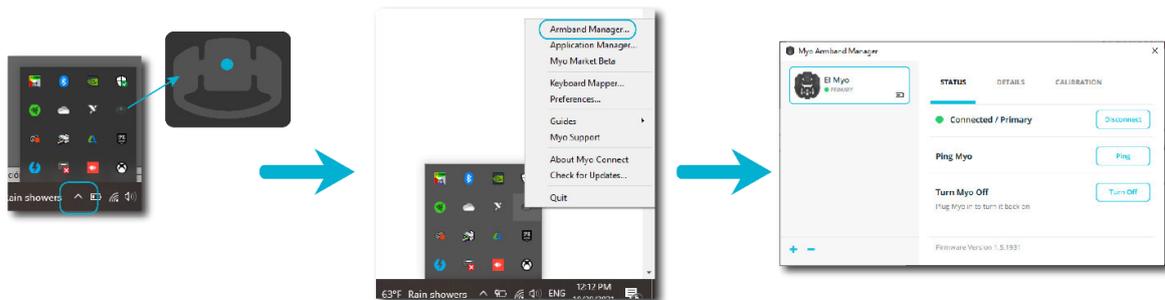
1.1. Colocar el sensor en el antebrazo en una posición cómoda: no se debería sentir muy apretado ni muy flojo.



1.2. Conectar el USB del sensor Myo Armband a la computadora. Se puede verificar el funcionamiento del mismo mediante el software del fabricante (Myo Connect).



- El estado de la conexión se puede comprobar en la aplicación de Armband Manager a la cual se puede acceder mediante la barra de tareas de Windows en los íconos ocultos.



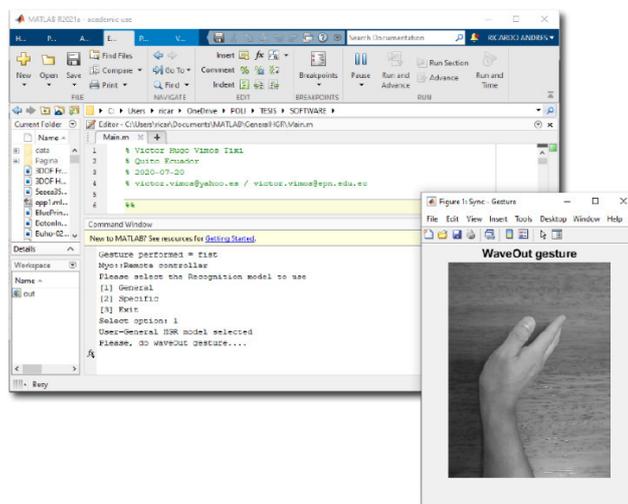
- Para verificar que el sensor se encuentra correctamente conectado y funcionando se puede ingresar al panel de información disponible en

<http://diagnostics.myo.com/>. En este enlace se desplegará la siguiente ventana:

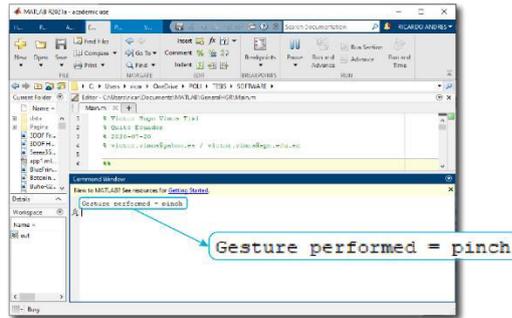


1.3. Abrir una nueva ventana de Matlab y ejecutar el archivo llamado "Main.m". Al correr este programa se solicitará escoger una opción para el funcionamiento. Se selecciona la opción 1 para Usuario General y se pide realizar el gesto de sincronización.

1.4. En este punto se debe realizar el gesto **Wave Out** y mantenerlo durante unos segundos para calibrar el sistema. Luego de esto, el reconocimiento de gestos se encuentra listo para su uso.



1.5. Se puede verificar el funcionamiento del sistema de HGR mediante la etiqueta del gesto realizado que se muestra en la ventana de comandos de Matlab.



## 2. Encendido del Helicóptero de 3-GDL

2.1. Para encender el sistema se debe girar el botón de emergencia. Verificar que el jumper de alimentación esté conectado y el cable USB desconectado. Se debe encender el LED amarillo (motores apagados) y el LED rojo (alimentación desde la fuente).



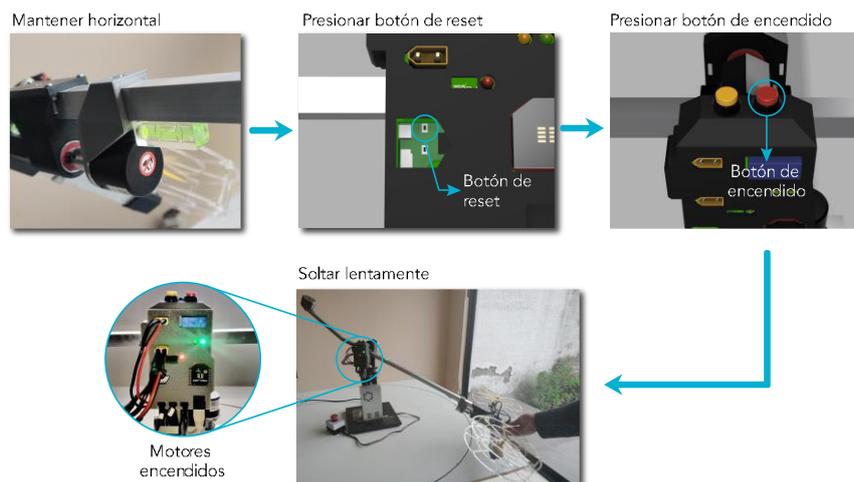
- La caja de control posee 3 indicadores luminosos que permiten conocer el estado de la alimentación de los componentes. Los indicadores amarillo y verde muestran el estado de la alimentación de la fuente y los motores. El LED amarillo indica que la fuente se ha encendido pero los motores se encuentran apagados. El LED verde indica que la fuente y los motores se encuentran encendidos. Por otro lado, el LED de color rojo muestra la alimentación de la placa Pyboard, cuando este está encendido significa que el jumper se encuentra conectado y la placa es alimentada desde la fuente. Cuando está apagado significa que el jumper está desconectado y

la placa se debe alimentar desde USB. Esto se hace en el modo de programación.

2.2. Mantener el brazo de soporte en posición horizontal al igual que el cuerpo del helicóptero. Para eso se mantienen las burbujas de nivel en el centro.

2.3. Sosteniendo al helicóptero en esta posición se debe presionar el botón de reset de la placa Pyboard. Con esto se determina la posición 0° del sistema. Inmediatamente después se debe presionar el botón rojo para encender los motores, en este momento se encenderá el LED de color verde.

2.4. Luego de 5s se escuchará una serie de pitidos que indican que los motores se han encendido. Luego de otros 5s las hélices empezarán a moverse. En este punto se puede soltar la plataforma lentamente y el sistema se encuentra ya encendido.

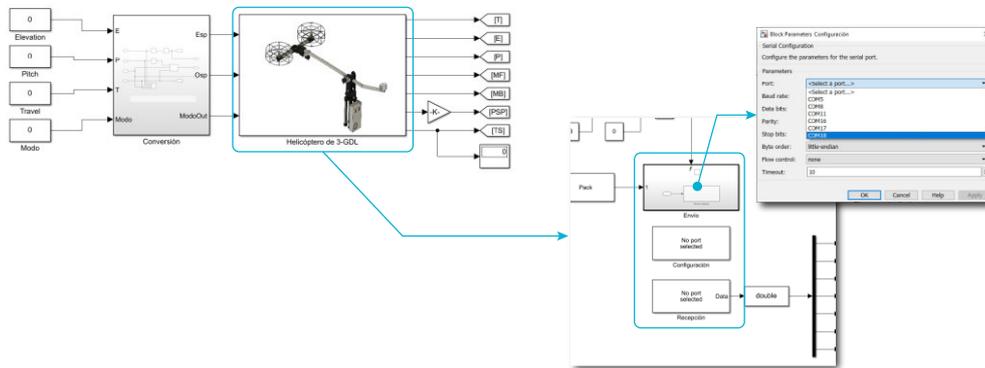


### 3. Activación de la conexión inalámbrica (Simulink)

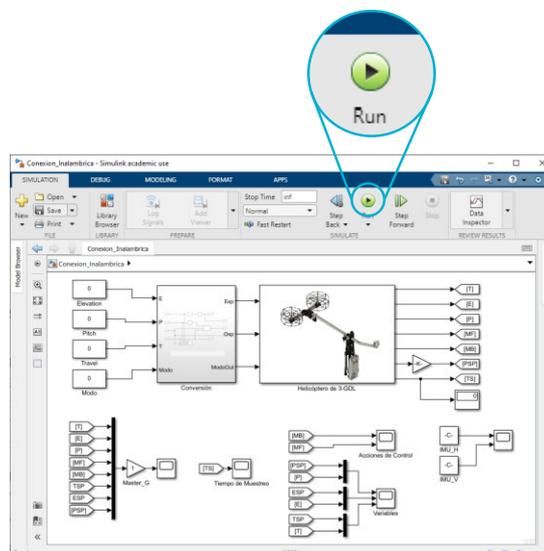
3.1. Para establecer la conexión inalámbrica con el helicóptero de 3-GDL lo primero que se debe hacer es conectar el adaptador de USB para el módulo

XBee a la computadora y verificar el puerto COM que se utiliza, lo cual se puede realizar en el administrador de dispositivos. Para este caso se usa el COM18.

3.2. Luego de conectar el USB se deberá abrir una nueva ventana de Matlab y correr el modelo en Simulink de nombre "Conexion\_Inalambrica.slx". Al abrir este modelo se podrá visualizar un bloque principal denominado "Helicóptero de 3-GDL", al cual se accede mediante doble clic. Dentro de este sistema se tienen otros 3 bloques necesarios para la comunicación serial: Envío, Recepción y Configuración. Para poder establecer la comunicación se deberá ingresar a cada uno mediante doble clic y cambiar el parámetro Port al puerto COM correspondiente; en este caso el COM18.



3.3. Una vez configurado el puerto serial se puede arrancar la simulación. Se puede verificar la comunicación con el helicóptero mediante los indicadores LED que se encenderán en los módulos XBee de la PC y la plataforma.



## 4. Uso de la interfaz

4.1. Una vez que se tienen funcionando el sistema de HGR y la conexión inalámbrica en Simulink se puede ejecutar la ventana de inicio donde se podrá observar el estado de ambas partes. Solo cuando ambas partes se encuentren funcionando se habilita el botón de "Arrancar Aplicación". Caso contrario solo se tiene activo el botón para ingresar a esta ventana de información y el botón para cerrar.

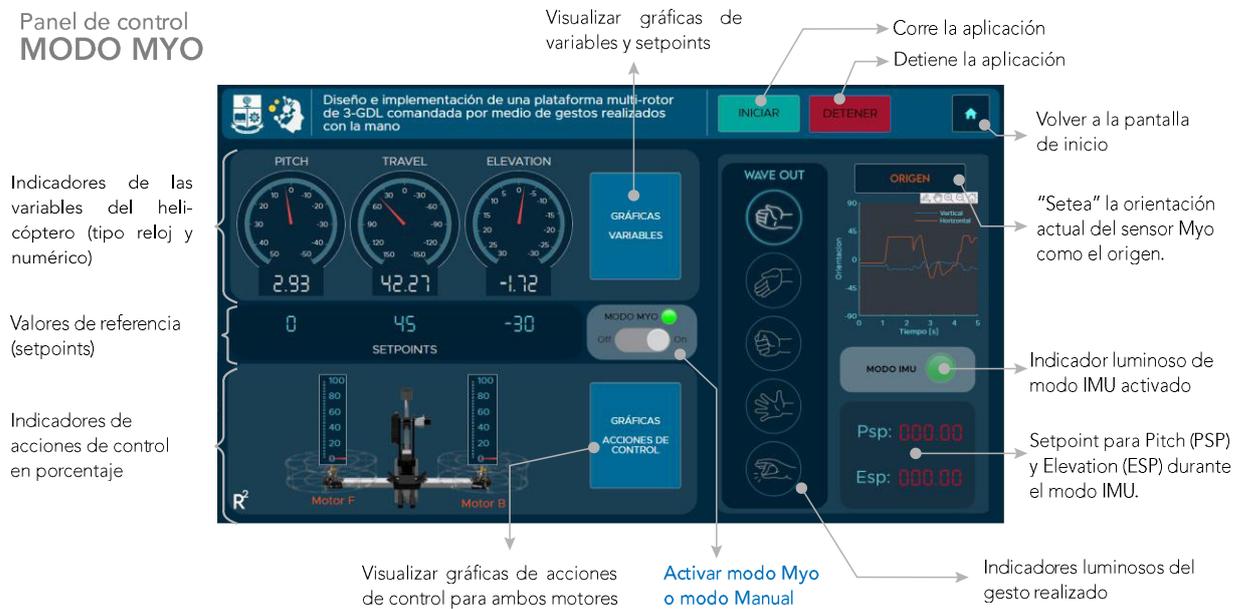


4.2. En caso de tener ambas partes funcionando (Simulink y HGR) se presiona el botón de "Arrancar Aplicación" y se podrá ingresar al panel de control. En este punto la simulación se pausa automáticamente y se arranca nuevamente al presionar el botón de "INICIAR".

4.3. Luego de presionar el botón de "Arrancar Aplicación" se abre automáticamente el panel de control. En este panel se tiene acceso a todos los controles que se muestran en la imagen a continuación. En esencia, el panel posee dos partes: una de visualización a la izquierda y otro de comandos en la parte derecha. La sección de comandos se modifica al cambiar el modo de control de Manual a Myo.

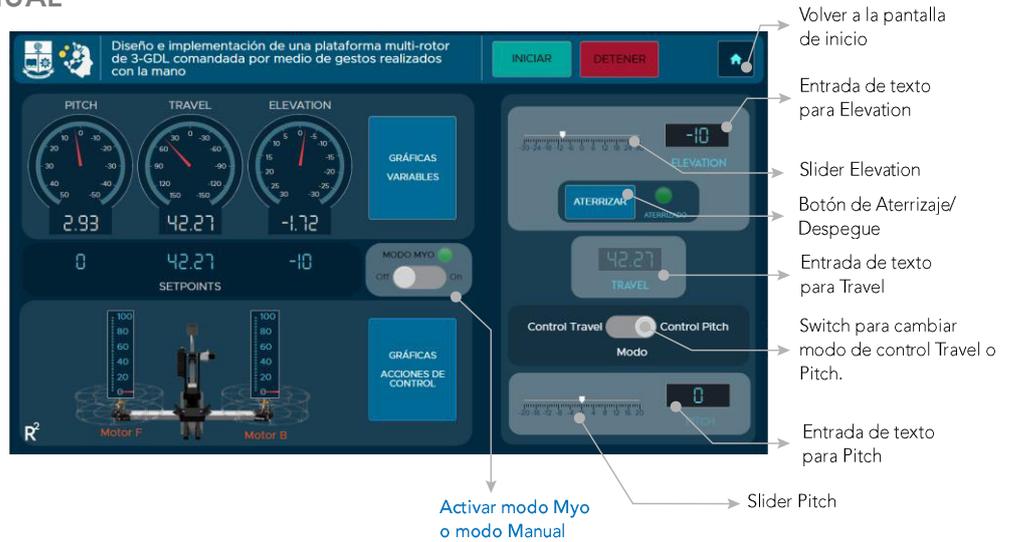
- En modo Myo la sección de comandos (parte derecha de la pantalla) muestra visualizadores de los gestos realizados y una gráfica para observar la orientación vertical y horizontal del sensor Myo.

### Panel de control MODO MYO



- En modo Manual se tienen las entradas de texto y los sliders para para manipular las variables pitch, elevation y travel. Dado que las variables Travel y Pitch se encuentran acopladas, solo se puede controlar una variable a la vez, por lo que si se escoge control tipo Travel se inhabilita la entrada para Pitch y viceversa. La razón por la cual en la variable Travel no se tiene un slider es que el rango de movimiento es demasiado amplio para acotarlo entre los límites del slider, por lo que solo se puede manipular mediante la entrada numérica.

Panel de control  
**MODO MANUAL**



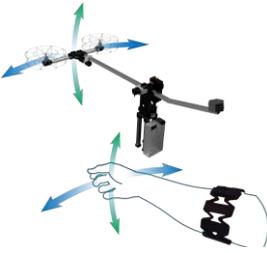
4.4. Al pulsar el botón de "INICIAR" los indicadores empiezan a mostrar los valores del helicóptero y de las entradas y gestos realizados. El botón de detener pausa el modelo en Simulink y detiene el funcionamiento del panel de control. Se puede regresar a la pantalla de inicio mediante el botón de la esquina superior derecha.

4.5. Finalmente, la pantalla de información presenta el manual de usuario de una forma interactiva y posee una barra de navegación independiente que permite acceder a cada una de las secciones que contiene. El ícono de la esquina superior izquierda permite regresar al inicio del manual.



## 5. Comandos de movimiento

Para controlar el movimiento de la plataforma durante el modo Myo se tienen los diferentes comandos que se muestran a continuación. Luego de seguir los pasos descritos para encender el prototipo y arrancar la interfaz se pueden ejecutar estos comandos utilizando el sensor durante el modo Myo.

Wave Out 		Girar a la derecha 45°
Wave In 		Girar a la izquierda 45°
Fist 		Cambiar modo gesto/imu
Pinch 		Aterrizar
Open 		Despegar

## ANEXO E

Resumen de costos de elaboración del prototipo de helicóptero de 3-GDL.

**Tabla E.1.** Costo de materiales.

	<b>Elementos</b>	<b>Cantidad</b>	<b>Costo Unitario</b>	<b>Costo total</b>
Componentes eléctricos/ electrónicos	Motores Brushless 1000KV	2	\$15.99	\$31.98
	ESC 30A	2	\$16.49	\$32.98
	Helices (paquete de 4 pares)	1	\$12.99	\$12.99
	Anillo deslizante 30A	1	\$29.89	\$29.89
	Pyboard	1	\$44.95	\$44.95
	Encoders 1000PPR	3	\$18.99	\$56.97
	Fuente 12V / 30A	1	\$18.95	\$18.95
	Relé 30A	1	\$7.50	\$7.50
	Relé 5A	1	\$1.00	\$1.00
	Pulsadores (NO y NC)	2	\$0.45	\$0.90
	Conector XT60 (par)	5	\$2.00	\$10.00
	XBee 3 antena PCB	2	\$17.95	\$35.90
	Adaptador XBee USB	1	\$25.95	\$25.95
	Adaptador regulado XBee 4 pines	1	\$10.95	\$10.95
	Cable 14 AWG	3	\$1.00	\$3.00
	Cable UTP	1.5	\$1.00	\$1.50
	Conector 4 pines	3	\$0.20	\$0.60
	Conector RJ45 hembra	2	\$1.00	\$2.00
	Conector RJ45 macho	2	\$0.05	\$0.10
	LED	3	\$0.10	\$0.30
	Resistencias 1kOhm	3	\$0.04	\$0.12
	Fusible automotriz 30A	1	\$0.20	\$0.20
	Botón de emergencia y caja	1	\$5.00	\$5.00
PCB	Placa principal (5 unidades)	1	\$7.40	\$7.40
	Placa auxiliar (5 unidades)	1	\$2.00	\$2.00
	Envío	1	\$20.00	\$20.00
Impresiones 3D	Soporte motores	2	\$3.00	\$6.00
	Soporte central fondo	1	\$2.50	\$2.50
	Soporte central tapa	1	\$3.00	\$3.00
	Soportes eje pitch	2	\$2.50	\$5.00
	Tapa soportes eje pitch	1	\$0.50	\$0.50
	Soporte de encoder pitch	1	\$2.00	\$2.00
	Soporte eje elevation	1	\$3.00	\$3.00
	Base elevation	2	\$3.50	\$7.00
	Caja de control fondo	1	\$2.50	\$2.50
	Caja de control tapa	1	\$1.50	\$1.50
	Soporte de encoder elevation	1	\$2.00	\$2.00

	Base travel superior	1	\$3.00	\$3.00
	Base travel inferior	1	\$4.00	\$4.00
	Acople engranaje	1	\$1.00	\$1.00
	Bases pequeñas	8	\$0.75	\$6.00
Acrílicos	Protector inferior	2	\$6.50	\$13.00
	Protector superior	2	\$6.50	\$13.00
	Columna	16	\$0.35	\$5.60
Metales	Perfil "L" de aluminio 30x450mm	1	\$1.50	\$1.50
	Tubo de aluminio 20x850mm	1	\$1.50	\$1.50
	Tubo de aluminio 20x300mm	1	\$0.55	\$0.55
	Contrapeso de acero (transmisión hexagonal 2 1/2" x 50mm)	1	\$6.10	\$6.10
	Acoples flexibles de 6.35 a 8 mm	2	\$4.00	\$8.00
	Eje travel (torneado)	1	\$7.50	\$7.50
	Eje pitch (torneado)	1	\$7.50	\$7.50
	Rodamiento (6000-2RS)	4	\$1.00	\$4.00
	Rodamiento grande (6205-RS)	1	\$2.50	\$2.50
Tornillos, tuercas y seguros	Tornillo M5 largo (0.8x30mm)	20	\$0.09	\$1.80
	Tornillo M5 corto (0.8x10mm)	5	\$0.04	\$0.20
	Tuerca M5	25	\$0.04	\$1.00
	Tornillo M3 largo (0.5x20mm)	20	\$0.04	\$0.80
	Tuerca M3	20	\$0.05	\$1.00
	Tornillo M3 corto (0.5x10mm)	10	\$0.04	\$0.40
	Tornillo M4 (0.7x15mm)	5	\$0.04	\$0.20
	Tuerca M4	5	\$0.03	\$0.15
	Seguro para rodamiento	5	\$0.27	\$1.35
	Tornillo M5 grande (0.8x100mm)	1	\$0.36	\$0.36
	Tuerca mariposa M5	1	\$0.13	\$0.13
Otros	Repisa de madera 30x60cm	1	\$8.00	\$8.00
	Plataformas de caucho (4 unidades)	1	\$3.00	\$3.00
			<b>TOTAL</b>	<b>\$501.27</b>

**Tabla E.2.** Costo de ingeniería.

Categoría	Número de horas	Costo por hora	Total
Estudio previo e investigación	50	\$5.00	\$250.00
Diseño electrónico	125	\$10.00	\$1,250.00
Programación	150	\$10.00	\$1,500.00
Elaboración de piezas y ensamblaje	75	\$10.00	\$750.00
<b>TOTAL</b>	<b>400</b>	<b>-</b>	<b>\$3,750.00</b>

**Tabla E.3.** Costo total.

<b>Concepto</b>	<b>Total</b>
Materiales	\$501.27
Ingeniería	\$3,750.00
<b>TOTAL</b>	<b>\$4,251.27</b>

Con base en los costos presentados en estas tablas el costo total estimado para el prototipo del helicóptero de 3-GDL elaborado es de \$4,251.27.

Comparado con el valor del prototipo comercial de la marca Quanser, encontrado en [59] (alrededor de \$13,029.75), el costo del modelo realizado en este proyecto representa menos del 33% del valor del modelo comercial, es decir, hay una reducción de más del 67% en el costo.