

Controlador PID para equilibrar el balanceo de un dron

Enzo Bórquez
Ingeniería Civil Electrónica
Universidad Austral de Chile
Valdivia, Chile
Email:

Braulio Jeldres
Ingeniería Civil Electrónica
Universidad Austral de Chile
Valdivia, Chile
Email:

Ricardo Román
Ingeniería Civil Electrónica
Universidad Austral de Chile
Valdivia, Chile
ricardo.roman@alumnos.uach.cl

Resumen—Escribir Resumen al final.

Palabras Clave—Raspberry, Python, PID, Dron

I. INTRODUCCIÓN

Los drones, por su construcción y los elementos utilizados para ella, además de condiciones externas, no siempre se comportan como se requiere. De forma intrínseca, los drones son inestables al momento de volar, ya sea cuando se desplazan o cuando deben permanecer estáticos en un punto. Un ejemplo de elementos que pueden generar perturbaciones al momento del vuelo de un dron son sus motores, ya que estos, a pesar de que son fabricados en masa con las mismas especificaciones, no todos funcionan igual, por lo que, al tener un control de lazo abierto no siempre funcionarán como se espera, es por esto que al retroalimentar el sistema con sensores de aceleración y giroscopio se podrá predecir la forma más óptima para llegar al estado de equilibrio deseado.

II. IDENTIFICACIÓN DEL PROBLEMA

El movimiento de un dron se realiza en un espacio tridimensional, debido a esto, el control de dicho movimiento puede realizarse a lo largo de tres ejes. Para que el dron permanezca estático en un punto, es necesario que los motores en conjunto generen un empuje que neutralice la gravedad y el par motor en todos sus ejes debe ser cero. Las rotaciones de un dron son denominadas *Roll*, *Pitch* y *Yaw*, esto a lo largo de los ejes x , y y z respectivamente, tal como se muestra en la figura siguiente (1).

Como se menciono, por naturaleza, los drones no son estables al momento de volar o permanecer estáticos, es por ello que es necesario que estos cuenten con un controlador para desarrollar esta estabilidad, si bien los drones poseen rotaciones a lo largo de tres ejes, definir o implementar un control sobre la rotación de uno de ellos es suficiente en este caso, ya que este puede ser aplicado a los otros dos ejes restantes. Por lo anterior se planteo realizar con controlador PID para controlar la rotación y estabilidad de un dron a lo largo del eje x , es decir, un controlador PID para el control del *roll* de un dron.

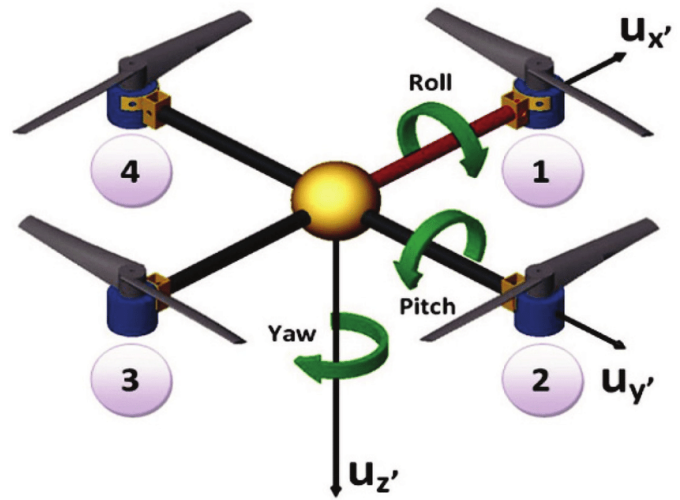


Figura 1: Rotaciones roll, pitch y yaw de un cuadricóptero



Figura 2: Ángulos de roll, pitch y yaw de un cuadricóptero

III. DEFINICIÓN DE LOS REQUISITOS

Para que la solución de este problema sea lo mas efectiva posible, es necesario establecer una serie de requisitos a cumplir, estos son tanto requisitos físicos (construcción del dron de prueba, elementos a utilizar, etc.) como también requisitos a nivel de software.

■ III-A. Construcción balanceada

Si bien lo que se plantea es realizar un controlador PID para obtener un balance lo mas estable posible, es necesario que la construcción del dron sea lo mas simétrica posible, esto solo por temas prácticos al momento de realizar las pruebas del controlador.

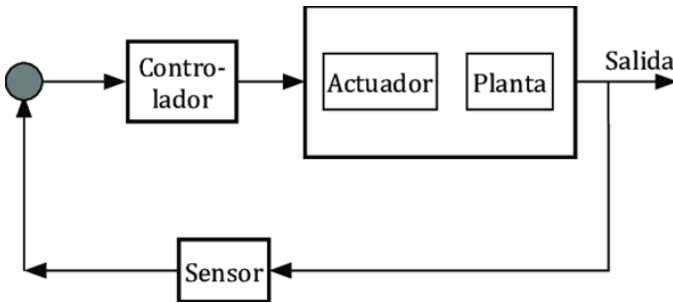


Figura 3: Control de planta por lazo cerrado.

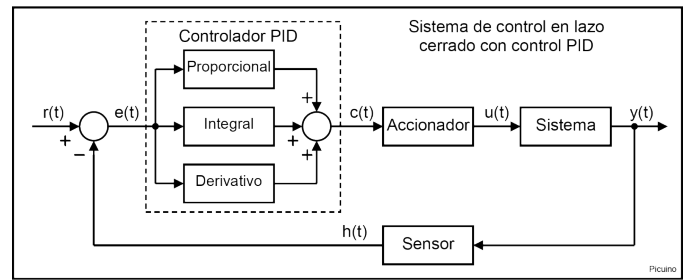


Figura 4: Control de planta por lazo cerrado con PID.

■ III-B. Flexibilidad del controlador

Es necesario generar un controlador que sea flexible, esto para su adaptabilidad a distintas condiciones de construcción del dron y condiciones externas a este.

■ III-C. Accesibilidad visual

En cuanto a la accesibilidad visual, esta está pensada para la visualización de los parámetros de ajuste del controlador, información que resulta relevante para el usuario. También, se requiere obtener información de las condiciones y estado del dron en cuanto a su posición y ángulo de rotación de este.

■ III-D. Interacción y configuración a nivel de usuario

En relación con el requisito anterior, se estima conveniente que el controlador pueda ser ajustado sin la necesidad de acceder al código de este, esto supone que el usuario pueda controlar los parámetros de ajuste desde la interfaz gráfica del controlador o de forma física (mediante pulsadores o encoder rotativos).

Si bien el controlador, en primera instancia, está pensado para que el dron permanezca estable con un ángulo 0 con respecto al eje horizontal, es posible que se requiera que este también permanezca estable con un ángulo diferente a 0 con respecto al mismo eje

IV. PLANTEAMIENTO DE ALGORITMO DE SOLUCIÓN

La solución propuesta al problema planteado es la implementación de un controlador PID (por su sigla en inglés de controlador proporcional, integral y derivativo). Este controlador permitirá generar la estabilidad deseada en la rotación del dron en su eje x .

IV-A. Controlador PID

El controlador PID, conceptualmente, calcula la diferencia entre el valor de una variable real contra el valor de la variable deseada, el cual permite el control de un sistema de lazo cerrado para que este alcance el estado de salida deseado. Como su nombre lo indica, está constituido por tres elementos, los cuales entregan una acción proporcional (P), integral (I) y derivativa (D). Este controlador será utilizado para disminuir la oscilación del dron cuando el balance de este

sea perturbado de forma externa y para que el dron regrese a su posición de equilibrio establecida de forma suave y precisa. La señal $r(t)$ del diagrama de la figura 4 corresponde a la señal de referencia del sistema, en este caso, esta señal $r(t)$ corresponde al ángulo ϕ (según la figura 2) del roll deseado mientras que la señal de salida $y(t)$ corresponde al valor del ángulo ϕ real, el cual variara con respecto al tiempo (producto de perturbaciones externas). La señal $e(t)$ de entrada al controlador PID corresponde al error del sistema, o sea, la diferencia entre el ángulo deseado dado por la señal $r(t)$ y el ángulo real, el cual es entregado por el sensor a utilizar mediante la señal $h(t)$. La magnitud del error define que tan cerca del estado deseado se encuentra el sistema, si el error es grande, el sistema se encuentra lejos del punto de equilibrio, mientras que si el error es pequeño, el sistema se encuentra cerca o en el punto de equilibrio.

IV-A1. Acción de control proporcional P: Tal como lo indica su nombre, esta acción de control es proporcional al error medido del sistema $e(t)$. El algoritmo del controlador PID multiplica el error del sistema por una constante definida como K_p , la cual determina la cantidad de acción proporcional que tendrá el controlador. La función de esta acción de control es intentar disminuir el error del sistema. Cuando el error es grande, la acción de control proporcional también lo es y tiende a minimizar este error.

IV-A2. Acción de control integral I: Esta acción de control, también como lo indica su nombre, es proporcional a la integral de la señal del error $e(t)$ del sistema multiplicada por una constante K_i . La integral, en este caso, se puede ver como la suma o acumulación de la señal de error.

IV-A3. Acción de control derivativa D: Esta acción de control es proporcional a la derivada de la señal de error multiplicada por un factor k_d . La derivada de la señal de error corresponde a la velocidad de cambio de dicha señal. La importancia de esta acción radica en la velocidad con que el error disminuye, y por lo tanto, la señal real se acerca al valor de referencia, ya que, debido a esta velocidad y a la inercia del sistema, este "pasará de largo" del valor de referencia, por lo cual se producirán oscilaciones en torno al valor de referencia. Para disminuir o evitar dichas oscilaciones, el sistema debe reconocer y frenar cuando el sistema se encuentre cerca de su valor de referencia.

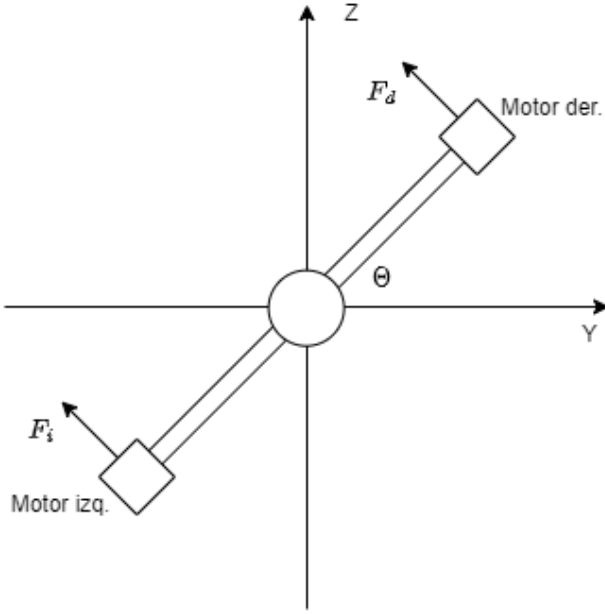


Figura 5: Diagrama de fuerzas y ángulo.

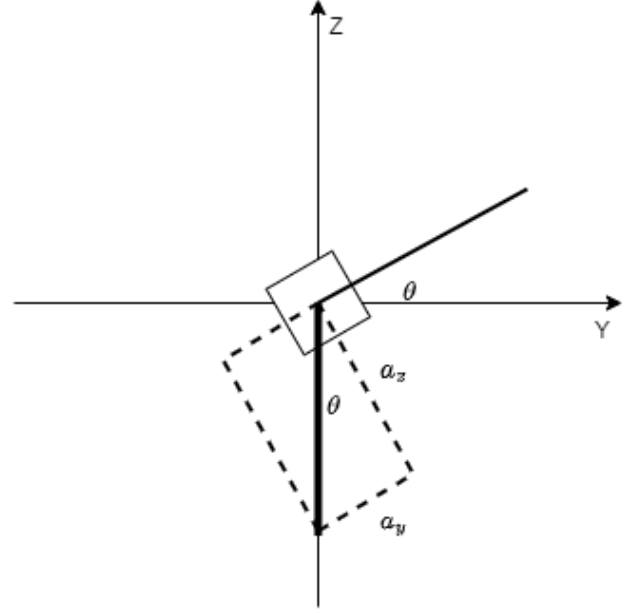


Figura 6: Diagrama de aceleraciones.

V. PLANTEAMIENTO DE LAS ECUACIONES

Tomando como referencia el diagrama de la figura 5, para que el sistema quede en equilibrio en torno al valor del ángulo de referencia, el cual, en este caso será $\theta_{ref} = 0$ es necesario que la fuerza ejercida por ambos motores sean iguales, o sea:

$$F_i = F_d \quad (1)$$

Donde F_i y F_d corresponden a las fuerzas ejercidas por los motores izquierdo y derecho respectivamente. Además, las fuerzas ejercidas por cada motor se definen como:

$$F_i = K_f * \omega_i^2 \quad (2)$$

$$F_d = K_f * \omega_d^2 \quad (3)$$

K_f corresponde a la constante de fuerza de cada motor, la cual es propia de cada uno, sin embargo, considerando que ambos motores son iguales en cuanto a modelo y características, se asume que ambas constantes son iguales. Por su parte, ω_i y ω_d corresponden a las tasas de rotación (medida en rad/s) de los motores izquierdo y derecho respectivamente. Para poder cambiar las fuerzas ejercidas por los motores, es necesario controlar la velocidad de cada uno, este control se realiza mediante el ángulo θ que se muestra en la figura 5, la velocidad de cada motor variará en relación a la diferencia entre el ángulo del sistema θ y el de referencia θ_{ref} . Además de lo anterior, otra condición para que el sistema esté en equilibrio en el punto deseado, es necesario que:

$$\theta - \theta_{ref} = 0 \quad (4)$$

Por lo tanto la condición de la ecuación 1 como la condición la ecuación 4 se deben cumplir para que el sistema permanezca en equilibrio en torno al punto deseado.

El sistema contará con una Unidad de Medición Inercial (IMU), el cual provee de los datos de aceleración en los ejes x , y , y z , dicha aceleración se mide en g por lo tanto, los valores de estas varían entre -1 y 1 en función del ángulo de inclinación en cada eje. En el caso del controlador a implementar, este solo controlará el giro en eje x , por lo tanto, es necesario medir el ángulo θ el cual se encuentra en el plano yz .

Como se menciona, la IMU entrega los datos de aceleración en cada eje de la unidad, y para poder determinar el ángulo θ del sistema, basta con conocer las aceleraciones en el eje y y z de esta, las cuales, en el diagrama de la figura 6 se denotan como a_y y a_z respectivamente. Entonces, el ángulo θ se puede determinar mediante la siguiente ecuación:

$$\theta = \arctan\left(\frac{a_y}{a_z}\right) \quad (5)$$

Otra forma de determinar el ángulo es mediante el valor de la velocidad angular entregada por el *Gyro* de la unidad IMU. Si se conoce el ángulo inicial del sistema, es posible determinar el ángulo en cada momento utilizando la velocidad angular como se mencionó, esto utilizando la siguiente ecuación:

$$\theta = \theta_i + \omega_x \cdot t \quad (6)$$

Las ecuaciones planteadas anteriormente son parte fundamental del controlador implementado, ya que estas son las que modelan al mismo, donde θ_i corresponde al ángulo inicial (conocido), ω_x es la velocidad angular en el eje x y la variable t es el tiempo entre mediciones de ω_x . Sin embargo la ecuación anterior calcula un nuevo θ con respecto al ángulo inicial, el cual no varía conforme se producen nuevas mediciones de ω_x , por lo tanto, es necesario definir dentro del código que se

implementa un ángulo θ en función de su valor anterior y de ω_x , lo cual implica agregar la siguiente igualdad luego de la ecuación 6

$$\theta_i = \theta \quad (7)$$

V-A. Filtro

Debido a condiciones inherentes a la IMU, a la tarjeta de desarrollo utilizada y a la electrónica en general, la señal de los datos de la IMU posee una cantidad de ruido considerable, por lo tanto, es necesario aplicar un filtrado a dicha señal, esto para prevenir variaciones en los datos obtenidos que no sean producto del movimiento de la IMU. Una forma sencilla de subsanar este problema de ruido presente en la señal es aplicando un filtro denominado complementario, el cual, en su forma mas sencilla para el calculo de θ puede expresarse de la siguiente forma:

$$\theta = a \cdot (\theta_\omega) + (1 - a) \cdot \theta_a \quad (8)$$

donde θ_ω y θ_a corresponden a los ángulos calculados utilizando la velocidad angular y la aceleración respectivamente, lo cual, de forma expandida corresponde a:

$$\theta = a \cdot (\theta_i + \omega_x \cdot t) + (1 - a) \cdot \arctan\left(\frac{a_y}{a_z}\right) \quad (9)$$

típicamente, $a = 0,98$.

Utilizando la ecuación 8 o la ecuación 9 produce una señal mas suave y con menos variaciones producto del ruido presente en las lineas.

VI. PLATAFORMA DE DESARROLLO Y SOFTWARES

Para este proyecto se utilizo la placa de desarrollo Raspberry pi 4 en su versión de 4Gb, esta plataforma, junto con el lenguaje de programación Python, ofrece una robusta solución en cuanto a la velocidad y potencia de procesamiento de los datos y señales a utilizar.

VI-A. Plataforma de desarrollo

Como se menciona, para el desarrollo del del controlador PID se utilizo la placa de desarrollo Raspberry Pi 4, en la cual fue instalado el sistema operativo Raspberry Pi OS (64 bits) mediante de la herramienta Raspberry Pi Imager v1.7.3

VI-B. Softwars

El código correspondiente fue escrito utilizando el lenguaje de programación "Python" en su versión 3.9.2 mediante la herramienta de programación "Thonny" incluida en el sistema operativo antes mencionado. Además, para el diseño físico del dispositivo fue utilizado el software "Inventor Professional 2023" para el diseño de las partes.

VII. DESARROLLO DEL SOFTWARE

Para diseñar el software se diseñó un diagrama de flujo figura 7 en donde en la primera parte se establece el ángulo de equilibrio que se quiere obtener como resultado, además de la velocidad inicial de los motores, luego se asignan los valores de las constantes proporcionales del PID, estas son modificadas para obtener el comportamiento deseado, después

de esto se establecen los valores iniciales para los bloques proporcional, integral y derivativo además del error, en este caso se establecen en 0 para que el software no envíe error, siguiendo con el diagrama se establecen valores temporales y una variable para poder parar el proceso. Iniciando el bucle, lo primero que aparece es la captura de los valores proporcionados por el acelerómetro para luego asignar el ángulo en el eje X (ecuación 5), luego se captura los datos del giroscopio, se establece la variación temporal para luego calcular la ecuación 9 con esto se determina el error respecto a la referencia, con este valor se determinan los valores P, I y D los cuales son sumados para obtener el resultado de la función PID, seguido de esto se obtienen los valores para los ciclos de trabajo de cada motor los cuales son entregados a la ESC, seguido de esto se verifica si se quiere seguir con el ciclo de trabajo, si es presionada la parada el valor cambia a True si se mantiene en False se guardan los valores del ciclo que acaba de pasar como el control integrativo, el error y el tiempo previo para luego capturar el nuevo tiempo y volver al ciclo de captura de datos del acelerómetro.

VIII. DISPOSITIVO FÍSICO

Las siguientes imágenes corresponden a las partes fabricadas para el dispositivo físico del proyecto, el cual esta construido utilizando aluminio e impresión 3d.

La figura 8 muestra la barra horizontal del dispositivo, en la cual, como se puede apreciar, se encuentran montados en los extremos los dos motores utilizados. Para la fijación de los motores a la barra fueron diseñados e impreso dos soportes utilizando impresion 3d.

En la figura 9 se puede ver la parte inferior de la barra horizontal, en la cual estan sujetos los controladores de velocidad electrónicos (ESC).

La figura 10 muestra uno de los dos soportes de la barra horizontal diseñados, estos cuentan en su interior con rodamientos para disminuir la fricción.

En la figura 11 se pueden ver ambos soportes de la barra horizontal.

La figura 12 corresponde al dispositivo ya montado, incluyendo los soportes verticales de la barra horizontal, los motores con sus respectivas helices y la IMU en el centro del dispositivo.

Finalmente, para mejorar la estabilidad del dispositivo, como también para aumentar el peso de este, fue agregada una base de MDF, la cual brinda un mayor soporte al dispositivo, lo cual se puede ver en la figura 13.

La IMU utilizada fue fijada a los soportes de la barra vertical, esto para asegurar una correcta medición del ángulo de dicha barra.

IX. COMPONENTES Y MATERIALES

Los motores utilizados fueron motores brushless modelo A2212/6T de 2200KV, como el que se puede ver en la imagen de la figura 15: Estos motores puede girar a 2200[RPM] por voltio suministrado, consumiendo una corriente de carga de 21,5[A] y generando un empuje de 732[g] (gramos) aprox.

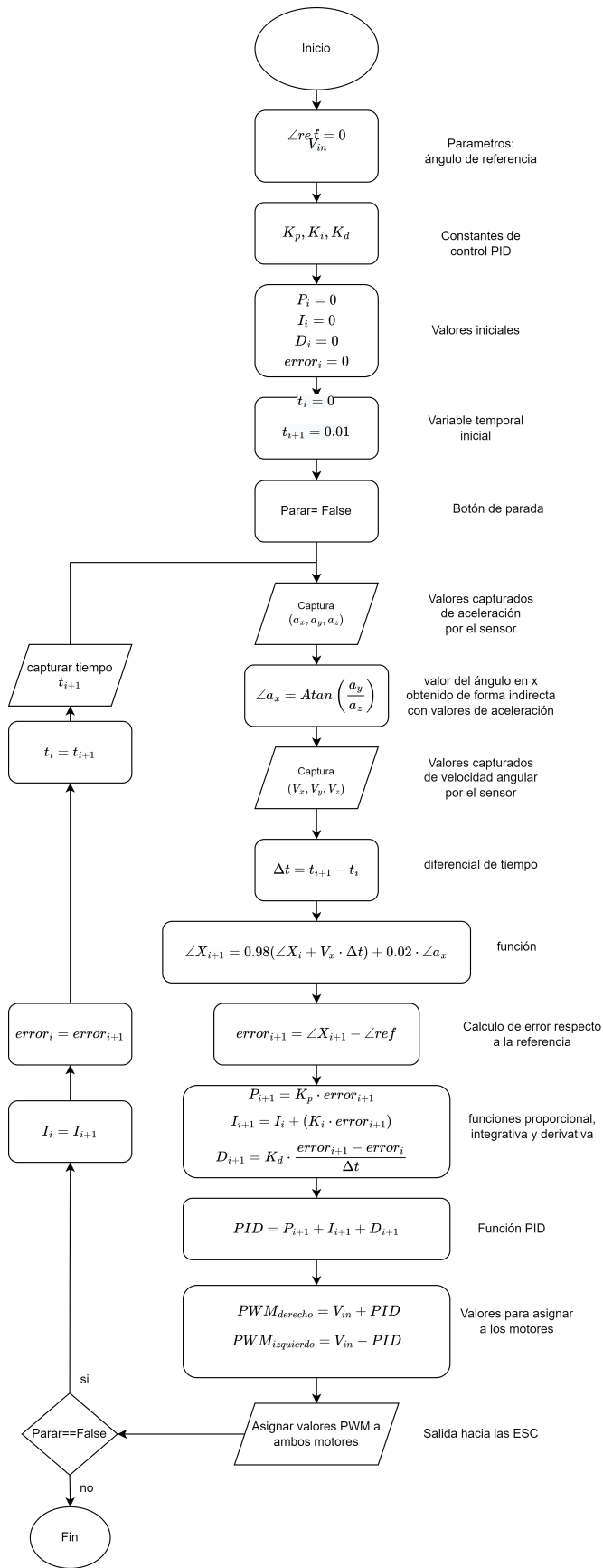


Figura 7: Diagrama de flujo.

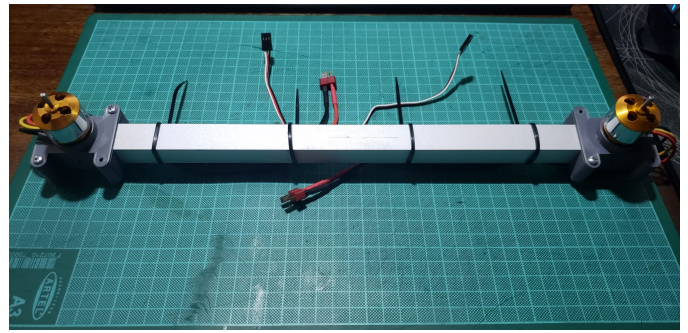


Figura 8: Barra horizontal y motores.

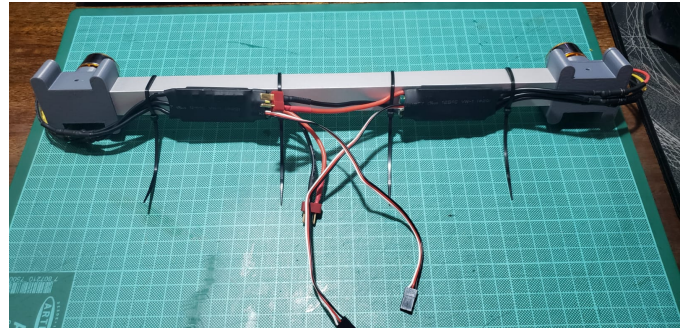


Figura 9: Barra horizontal y ESCs.



Figura 10: Soporte de barra horizontal.

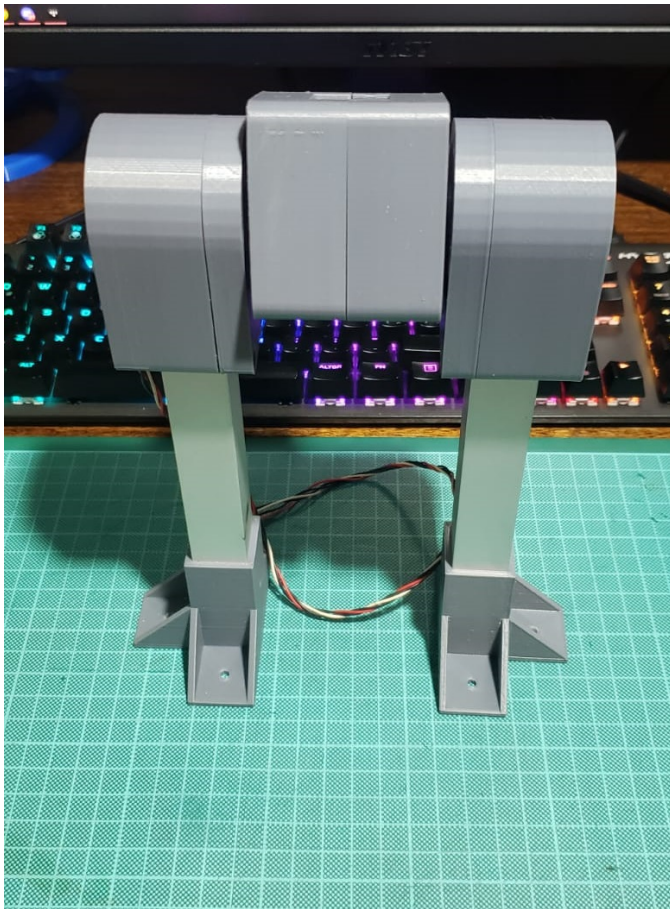


Figura 11: Soportes de barra horizontal.

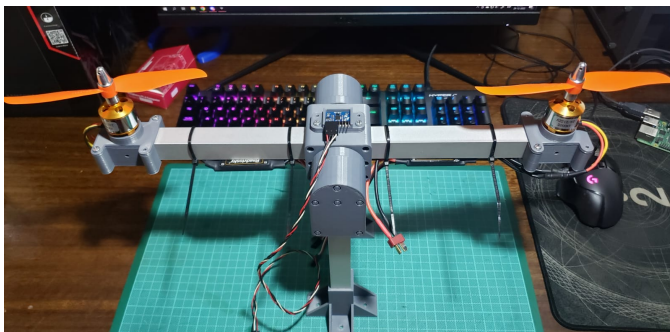


Figura 12: Dispositivo ensamblado.



Figura 13: Ensamble final.

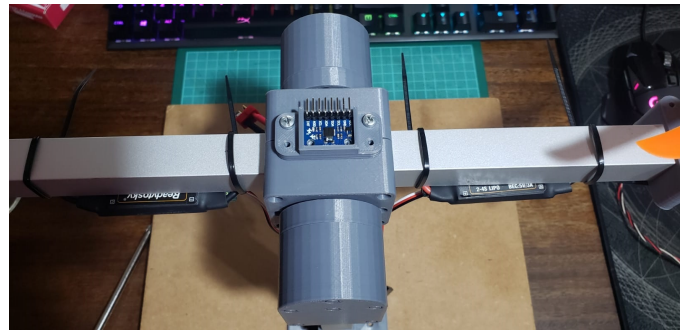


Figura 14: Unidad de medición inercial.



Figura 15: Motor A2212/6T.

Las hélices recomendadas y utilizadas para este modelo de motor son hélices 6030. En cuanto a los ESC utilizados para el control de los motores, estos corresponden a ESC 30A, y como su modelo lo indica, estos pueden manejar una corriente de hasta $30[A]$ ($40[A]$ máximos por 10 segundos). La IMU utilizada corresponde a una *MPU6050* de seis ejes, esta cuenta con un acelerómetro y giroscopio de 3 ejes cada uno. Además cuenta con un ADC de 16bits y puede ser manejada mediante I2C, posee conversores analógicos digitales por cada uno de los ejes, permitiendo así lecturas simultaneas de los 6 valores medidos por esta.

REFERENCIAS

- [1] Gomez, Victor y otros (2020). Pareto Optimal PID Tuning for Px4-Based Unmanned Aerial Vehicles by Using a Multi-Objective Particle Swarm Optimization Algorithm. Aerospace. 7. 10.3390/aerospace7060071.