

VALIDACIÓN EXPERIMENTAL DEL MODELO MATEMÁTICO DE UN
HELICÓPTERO RC EN VUELO ESTACIONARIO USANDO ARDUPILOT

WILSON DARÍO GÓMEZ SANTIESTEBAN

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECÁNICAS
ESCUELA DE INGENIERIAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2020

VALIDACIÓN EXPERIMENTAL DEL MODELO MATEMÁTICO DE UN
HELICÓPTERO RC EN VUELO ESTACIONARIO USANDO ARDUPILOT

WILSON DARÍO GÓMEZ SANTIESTEBAN

Trabajo de Grado para optar al título de
Ingeniero Electrónico

Director

DANIEL ALFONSO SIERRA BUENO

PhD en Ingeniería Biomédica

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECÁNICAS
ESCUELA DE INGENIERIAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2020

*Gracias a Dios por permitirme cumplir este logro.
Gracias a mis padres, hermanos y toda mi familia.
Gracias a la Universidad Industrial de Santander, a la E3T y al Dr. Daniel Sierra.*

TABLA DE CONTENIDO

<u>INTRODUCCIÓN.....</u>	<u>14</u>
<u>1. MARCO TEÓRICO.....</u>	<u>15</u>
<u>1.1. HELICÓPTEROS A RADIOCONTROL.....</u>	<u>15</u>
<u>1.1.1. Componentes del helicóptero a radiocontrol.....</u>	<u>16</u>
<u>1.1.2. Ángulos de navegación.....</u>	<u>19</u>
<u>1.1.3. Funcionamiento del helicóptero a radiocontrol.....</u>	<u>16</u>
<u>1.1.4. Señales de control.....</u>	<u>20</u>
<u>1.1.5. Vuelo estacionario.....</u>	<u>21</u>
<u>1.2. ARDUPILOT MEGA.....</u>	<u>22</u>
<u>2. MODELO MATEMÁTICO.....</u>	<u>24</u>
<u>2.1. MODELO TEÓRICO.....</u>	<u>24</u>
<u>2.2. MODELO EXPERIMENTAL.....</u>	<u>28</u>
<u>3. METODOLOGÍA.....</u>	<u>30</u>
<u>3.1. ESQUEMA GENERAL DEL PROCEDIMIENTO.....</u>	<u>30</u>
<u>3.2. MONTAJE EXPERIMENTAL.....</u>	<u>32</u>
<u>3.2.1. Estación en tierra.....</u>	<u>33</u>
<u>3.2.2. Estación a bordo.....</u>	<u>34</u>
<u>3.3. GENERACIÓN DATOS DE ENTRADA.....</u>	<u>36</u>
<u>3.4. REGISTRO DE DATOS DE SALIDA.....</u>	<u>39</u>
<u>3.5. BANCO DE PRUEBAS.....</u>	<u>40</u>
<u>3.5.1. Banco de pruebas 1-DOF.....</u>	<u>41</u>

<u>3.5.2. Banco de pruebas 3-DOF.....</u>	<u>41</u>
<u>4. RESULTADOS EXPERIMENTALES.....</u>	<u>43</u>
<u>4.1. MODELO DEL ÁNGULO DE GUIÑADA.....</u>	<u>43</u>
<u>4.1.1. Identificación y validación en banco 1-DoF.....</u>	<u>43</u>
<u>4.1.2. Identificación y validación en banco 3-DoF.....</u>	<u>46</u>
<u>5. CONCLUSIONES.....</u>	<u>49</u>
<u>REFERENCIAS BIBLIOGRÁFICAS.....</u>	<u>51</u>
<u>ANEXOS.....</u>	<u>56</u>

LISTA DE FIGURAS

Figura 1. Helicóptero HK-450.....	15
Figura 2. Componentes del helicóptero utilizado en este proyecto.....	17
Figura 3. Movimientos de un helicóptero.....	19
Figura 4. Conexiones en el helicóptero a radiocontrol.....	20
Figura 5.a) Comunicación emisora - helicóptero. b) Señal PWM c) Señal PPM.....	22
Figura 6. Placa Ardupilot Mega.....	23
Figura 7. Sistema MIMO.....	28
Figura 8. Sistema SISO del eje de guiñada.....	29
Figura 9. Esquema general.....	30
Figura 10. Esquema propuesto por Hazawa.....	32
Figura 11. Esquema de bloques.....	33
Figura 12. Estación de tierra.....	34
Figura 13. Estación a bordo.....	35
Figura 14. Acople de la señal de excitación al Radiocontrol.....	36
Figura 15. Método del doble receptor.....	37
Figura 16. Visualización de datos de entrada a los servomotores.....	38
Figura 17. Visualización datos de salida.....	40
Figura 18. Helicóptero en el banco de pruebas 1-DOF.....	41
Figura 19. Banco de pruebas 3-DOF.....	42
Figura 20. Señales de excitación y respuesta eje de guiñada, banco 1-DoF.....	44
Figura 21. Herramienta de identificación de sistemas de Matlab.....	44
Figura 22. Correlación de modelos, banco 1-DoF.....	45
Figura 23. Señales de excitación y respuesta eje de guiñada, banco 3-DoF.....	46
Figura 24. Correlación de modelos, banco 3-DoF.....	47

LISTA DE ECUACIONES

<u>Ecuación 1. Dinámica del rotor principal.....</u>	<u>25</u>
<u>Ecuación 2. Movimiento de alabeo.....</u>	<u>25</u>
<u>Ecuación 3. Movimiento de cabeceo.....</u>	<u>25</u>
<u>Ecuación 4. Movimiento de guiñada.....</u>	<u>26</u>
<u>Ecuación 5. Función de transferencia del movimiento de guiñada.....</u>	<u>27</u>
<u>Ecuación 6. Función de transferencia ángulo de guiñada banco 1-DOF.....</u>	<u>46</u>
<u>Ecuación 7. Función de transferencia ángulo de guiñada banco 3-DOF.....</u>	<u>48</u>

LISTA DE TABLAS

<u>Tabla 1: Especificaciones técnicas del helicóptero HK450.....</u>	<u>16</u>
<u>Tabla 2. Asignación de canales receptor.....</u>	<u>21</u>
<u>Tabla 3. Parámetros del modelo matemático.....</u>	<u>26</u>
<u>Tabla 4. Mejor ajuste para el conjunto de datos, banco 1-DoF.....</u>	<u>45</u>
<u>Tabla 5. Mejor ajuste para el conjunto de datos, banco 3-DoF.....</u>	<u>48</u>

LISTA DE ANEXOS

<u>Anexo A. Código generación señal de entrada (PPM).....</u>	<u>56</u>
<u>Anexo B. Código lectura señal PWM del receptor.....</u>	<u>58</u>
<u>Anexo C. Código de lectura de sensores en el Ardupilot.....</u>	<u>59</u>
<u>Anexo D. Código Phyton para lectura de datos.....</u>	<u>66</u>

RESUMEN

TÍTULO: VALIDACIÓN EXPERIMENTAL DEL MODELO MATEMÁTICO DE UN HELICOPTERO RC EN VUELO ESTACIONARIO USANDO ARDUPILOT*

AUTOR: WILSON DARÍO GÓMEZ SANTIESTEBAN**

PALABRAS CLAVE: Helicóptero, Ardupilot, modelo matemático, eje de guiñada.

DESCRIPCIÓN:

En el siguiente documento, se presenta la validación experimental del modelo matemático del eje de guiñada de un helicóptero radiocontrolado en vuelo estacionario. Inicialmente se realiza un análisis del comportamiento dinámico del helicóptero mediante un modelo matemático teórico del eje de guiñada, esta información está compuesta por diferentes ecuaciones que fueron adquiridas de trabajos relacionados. El modelo teórico se usa como guía para el procedimiento experimental de identificación del modelo usando datos de vuelo reales, este procedimiento consiste en enviar señales de excitación (pseudoaleatorias) al helicóptero en vuelo estacionario y capturar las señales de salida (ángulo de guiñada). La adquisición de datos se realiza por medio de la tarjeta Ardupilot, la cual es un sistema de control y navegación empleado en vehículos aéreos no tripulados, esta tarjeta posee diferentes sensores (acelerómetro, giróscopo y magnetómetro) que permiten conocer la posición relativa del vehículo al cual se encuentra acoplado. Debido al comportamiento inestable del helicóptero y previendo accidentes, las pruebas experimentales se realizan en dos bancos de pruebas, el primer banco emula un grado de libertad y el segundo banco tres grados de libertad. Finalmente, con los datos de entrada y salida hallados en el procedimiento experimental y mediante el uso de la herramienta Matlab se identifica y valida el modelo para el eje de guiñada del helicóptero en vuelo estacionario.

*Trabajo de grado

**Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: PhD. Daniel Alfonso Sierra Bueno

ABSTRACT

TITLE: EXPERIMENTAL VALIDATION OF THE MATHEMATICAL MODEL OF A RC HELICOPTER IN HOVER FLIGHT USING ARDUPILOT *

AUTHOR: WILSON DARÍO GÓMEZ SANTIESTEBAN**

KEY WORDS: Helicopter, Ardupilot, mathematical model, yaw axis.

DESCRIPTION:

The following document describes the experimental validation of yaw axis mathematical model of a radio-controlled helicopter in hover. Initially, analysis of the dynamic behavior of the helicopter is carried out using a theoretical mathematical model of yaw axis. The information used is made up of related equations that were obtained from other projects. The theoretical model is used as a guide for the experimental procedure of model identification using real flight data acquisition, this procedure consists of sending arousal (pseudorandom) signals to the helicopter in hover and obtain the output signals (yaw angle). Data acquisition is carried out by Ardupilot board, which is a control and navigation system used in unmanned aerial vehicles. Ardupilot board has different sensors (accelerometer, gyroscope and magnetometer) that allow knowing the relative position of the helicopter when it is coupled. Due to the unstable behavior of the helicopter and also to prevent accidents, the experimental procedures are done out on two testing workbenches, the first workbench emulates one degree of freedom and the second one, three degrees of freedom. At last, input and output data found in the experimental procedure and using the Matlab tool, the model for yaw axis of the helicopter in hover is identified and validated.

* Degree work

** Faculty of Physic Mechanical Engineering. School of Electrical, Electronic and Telecommunications Engineering. Director: PhD. Daniel Alfonso Sierra Bueno.

INTRODUCCIÓN

Los vehículos aéreos no tripulados, UAV (*Unmanned Aerial Vehicle*), se han convertido en una tecnología de gran potencial en diversas áreas, debido a que son adaptables a diferentes necesidades y condiciones de trabajo como terrenos irregulares o accidentados. Entre las áreas de mayor aplicación están seguridad y misiones militares, prevención y control de desastres, cartografía aérea, video y fotografía, etc. Entre estos vehículos cobran gran importancia los vehículos aéreos no tripulados de ala rotatoria, como es el caso del helicóptero, debido a las funcionalidades que presenta como vuelo estacionario, despegue y aterrizaje vertical.

Los helicópteros presentan una dinámica no lineal, por lo que presentan dificultad a la hora de ser manejados y controlados. Esto se puede observar en las ecuaciones que componen los modelos matemáticos de estas aeronaves. En diferentes aplicaciones es importante mantener al helicóptero en vuelo estacionario, el cual consiste en mantener a la aeronave totalmente suspendida en el aire a cierta altitud y con una dirección, en este estado estacionario puede hacerse el modelo matemático más lineal, debido a que varios ángulos son cercanos a cero.

Este informe contiene esencialmente cinco capítulos. El primer capítulo presenta el marco teórico relacionado a los helicópteros de radiocontrol y la placa Ardupilot. El segundo capítulo contiene un estudio del modelo matemático teórico y experimental propuesto. El tercer capítulo presenta la metodología experimental, equipos, instrumentos y el procedimiento de recolección de datos. El cuarto capítulo presenta los resultados experimentales, la identificación y validación del modelo. Finalmente en el capítulo quinto se describen las conclusiones.

1. MARCO TEÓRICO

1.1. HELICÓPTEROS DE RADIOCONTROL

Un helicóptero de radiocontrol presenta los mismos principios de funcionamiento que un helicóptero real. Se caracterizan por ser una aeronave versátil ya que tiene movimiento en todas las direcciones, hacia atrás, adelante, derecha, izquierda, arriba, abajo y rotar en su propio eje, además de mantenerse en sustentación totalmente inmóvil, maniobra que se conoce como vuelo estacionario (hovering) [24]. El helicóptero utilizado en este proyecto es un helicóptero a radiocontrol eléctrico HK-450 mostrado en la Figura 1.

Figura 1. Helicóptero HK-450.



Fuente: Autor.

Este modelo de helicóptero de radiocontrol fue elegido debido a su gran maniobrabilidad, disponibilidad comercial y piezas de repuesto de costo bajo.

Tabla 1. Especificaciones técnicas del helicóptero HK-450.

Helicóptero HK450	
Longitud	660mm
Altura	230mm
Diámetro rotor principal	700mm
Diámetro rotor de cola	150mm
Peso	760g
Motor	Brushless
Servos	4 servos
Batería	LiPo 11.1v 1300mA
Duración de vuelo	10 min

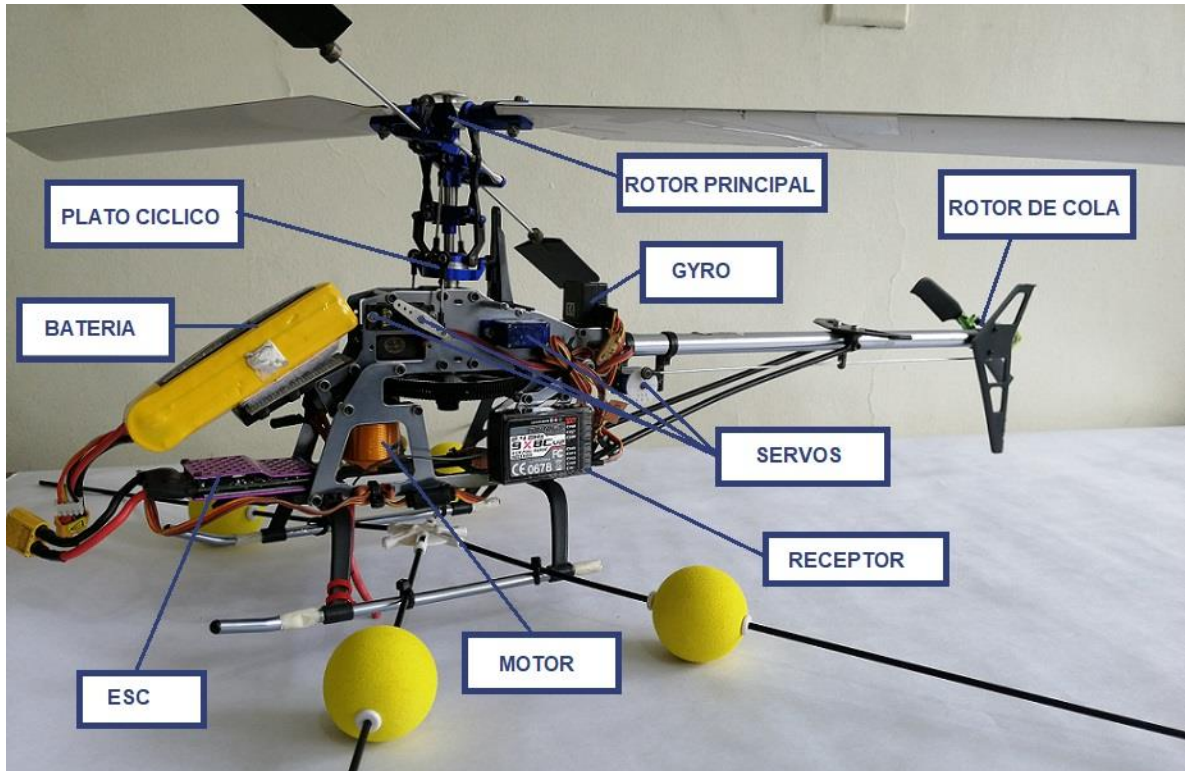
Fuente: [20]

1.1.1. Componentes del helicóptero a radiocontrol

En la figura 2 se presentan los principales componentes de un helicóptero de radiocontrol, los cuales se describen a continuación:

1. **Rotor principal:** este sistema es el encargado de mantener el helicóptero en el aire, así como para elevarlo o descender, sino también para impulsarlo hacia delante o hacia atrás, hacia los lados o en cualquier dirección. Esto se consigue variando el ángulo de incidencia (inclinación) de las palas dependiendo de su posición.

Figura 2. Componentes del helicóptero utilizado en este proyecto.



Fuente: Autor.

2. **Rotor de cola:** se encarga de anular el par reactivo generado por el rotor principal, evitando que el helicóptero gire en sentido contrario a la misma velocidad del rotor principal.
3. **Plato cíclico:** disco cuya posición vertical e inclinación permite variar el ángulo de ataque de las palas del rotor principal permitiendo la sustentación o movimiento del helicóptero.
4. **Motor:** motor eléctrico sin escobillas que está enlazado mecánicamente con el engranaje del rotor principal, dando movimiento a las palas principales y

mediante una correa dentada da movimiento al rotor de cola.

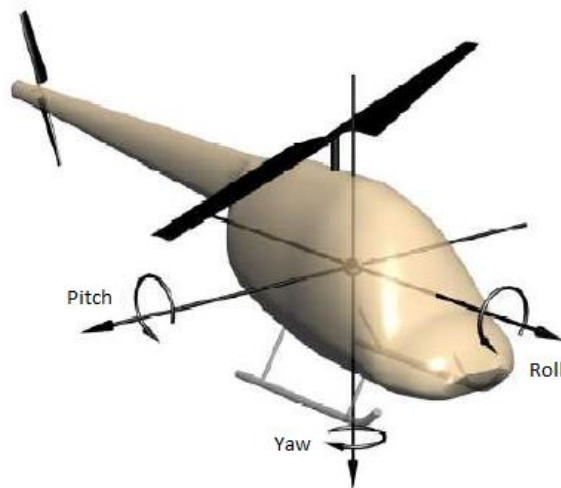
5. **Variador de velocidad (ESC):** dispositivo encargado de regular la velocidad de giro del motor y de transformar la corriente continua que llega de la batería en trifásica, que es la que solicita el motor. También incluyen un BEC (*battery eliminator circuit*), dispositivo que reduce el voltaje de la batería a 5V para alimentar al receptor.
6. **Servomotores:** dispositivos actuadores que tienen la capacidad de ubicarse en cierta posición dentro de un rango de acción, así como mantenerse estable en dicha posición. Se utilizan tres servomotores para manejar la posición del plato cíclico y un cuarto servomotor para controlar el ángulo de ataque de las palas del rotor de cola.
7. **Batería:** se utiliza una batería de polímero de litio (LiPo) para alimentar el variador, el motor y los servos.
8. **Giróscopo:** dispositivo electromecánico usado en el helicóptero para ayudar a semi-automatizar la respuesta del rotor de la cola. En un helicóptero RC se intercala eléctricamente entre el receptor y el servo que controla el paso de las palas del rotor de cola. Un sensor mide el cambio indeseado en el desvío del helicóptero y corrige la situación aumentando o disminuyendo el paso del rotor de cola para estabilizar el movimiento [27].
9. **Receptor:** dispositivo que establece la comunicación entre el radiocontrol y el helicóptero, recibe y decodifica la señal enviada por el radiocontrol y transmite esta información a los servos.

1.1.2. Ángulos de navegación

El helicóptero tiene gran libertad de movimientos, además de desplazarse verticalmente y avanzar a lo largo de la trayectoria longitudinal, el helicóptero puede rotar en torno a los tres ejes cartesianos (Figura 3), estas rotaciones se denominan de alabeo (roll), cabeceo (pitch) y guiñada (yaw) [6]:

- El *alabeo* define el movimiento de rotación alrededor del eje Y. Un movimiento de alabeo positivo se define cuando es realizado en sentido de las manecillas del reloj visto desde la parte izquierda del helicóptero.
- El *cabeceo* define el movimiento de rotación alrededor del eje X. Un movimiento de cabeceo positivo se define cuando es realizado en sentido de las manecillas del reloj visto desde la parte trasera del helicóptero.
- La *guiñada* define el movimiento de rotación alrededor del eje Z. Un movimiento de guiñada positivo se define cuando es realizado en sentido de las manecillas del reloj visto desde la parte superior del helicóptero.

Figura. 3. Movimientos de un helicóptero.



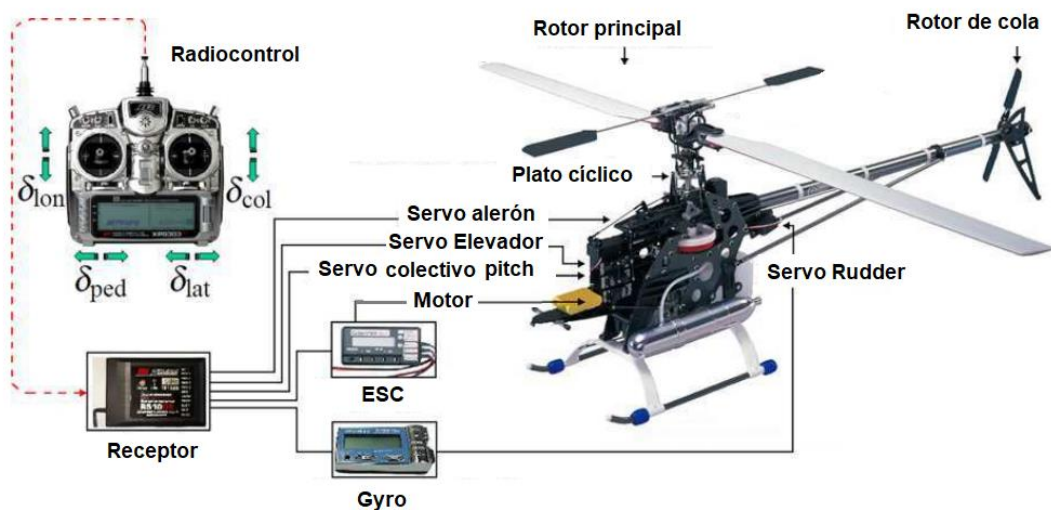
Fuente: [3]

1.1.3. Funcionamiento del helicóptero a radiocontrol

El rotor principal es controlado por tres entradas del control: *alerón*, *elevador* y *cabeceo*. El alerón permite mover el helicóptero lateralmente mediante movimientos laterales del plato cíclico; elevador, con un movimiento vertical del plato cíclico permite configurar el ángulo de ataque de las palas para elevar el helicóptero; cabeceo, permite al plato cíclico inclinarse de manera frontal, dando como resultado el movimiento frontal del helicóptero. Lo anterior se realiza mediante movimientos conjugados por tres servomotores. El rotor de cola es controlado por una entrada de control, *timón*, la cual mediante un servomotor varía el ángulo de ataque de las palas permitiendo el movimiento de guiñada.

El control de los servomotores se realiza por medio de un radiocontrol, que posee dos palancas o *sticks* de dos ejes cada uno. La palanca izquierda controla la aceleración del helicóptero (*throttle*) y la rotación del eje de guiñada (*yaw*). La palanca derecha maneja la inclinación en los ángulos de cabeceo (*pitch*) y alabeo (*roll*).

Figura 4. Conexiones en el helicóptero.



Fuente: Modificado de [15].

Para el manejo de estos servos se asignan cada uno de los canales del receptor, como se observa en la tabla 2.

Tabla 2. Asignación de canales en el receptor.

CANAL	CONEXION	INCIDENCIA DIRECTA	INCIDENCIA ADICIONAL	MOVIMIENTO DEL HELICOPTERO
CH1	Alerón (Aileron)	Servo 1	Servo 2	Derecha - izquierda
CH2	Elevador (Elevator)	Servo 2	Servo 1 Servo 3	Adelante- atrás
CH3	Acelerador (Throttle)	ESC	Servo 1 Servo 2 Servo 3	Arriba - abajo
CH4	Timón (Rudder)	Servo cola	No	Rotación
CH5	Gyro	No	No	No
CH6	Cabeceo (Pitch)	Servo 3	Servo1 Servo 2	Derecha - Izquierda

Fuente: Autor

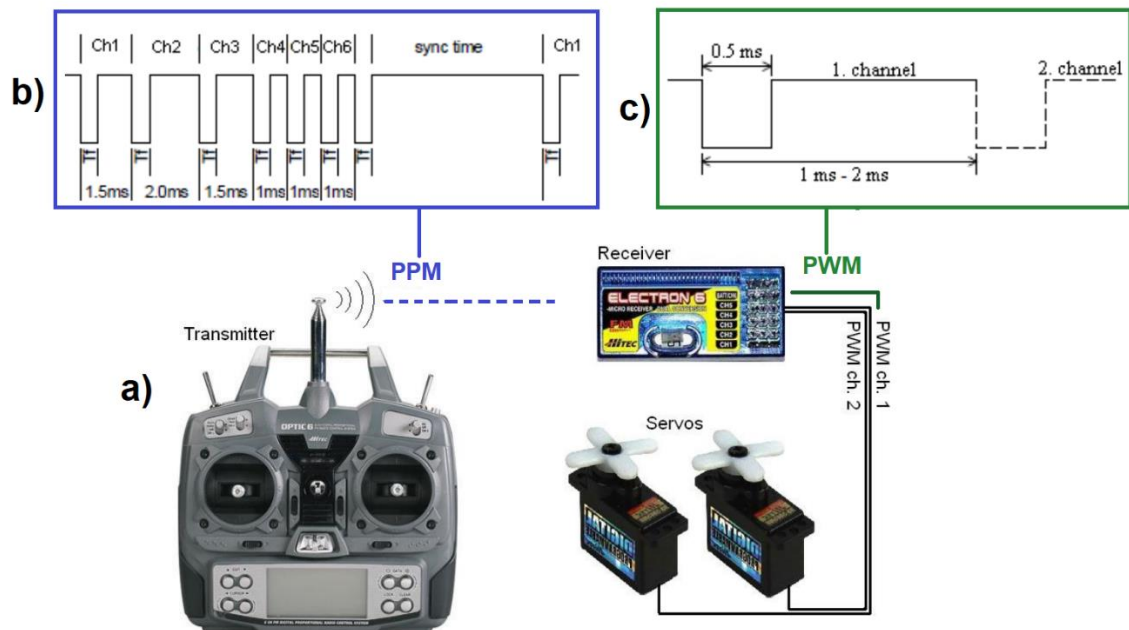
1.1.4. Señales de control

Una señal PPM (Modulación por Posición de Pulso) consiste en una trama de datos que contiene un pulso sincronizador seguido por un número de pulsos pequeños que son iguales al número de canales que use el radiocontrol, esta señal varía la posición del pulso a la vez que mantiene fijo el ancho de pulso. El radiocontrol lee cada potenciómetro e interruptor y construye la trama PPM que será enviada al receptor [25].

La señal PWM (Modulación de Ancho de Pulso) es la señal que envía el receptor a cada servomotor, las duraciones de cada pulso se interpretan como comandos de

posicionamiento del servo, mínimo de 1ms (posición de 0° en el servo) y máximo 2ms (posición de 180° en el servo). Este pulso se repite cada 20ms [26].

Figura 5. a) Comunicación emisora - helicóptero. b) Señal PWM. c) Señal PPM.



Fuente: Modificado de [16].

1.1.5. Vuelo Estacionario

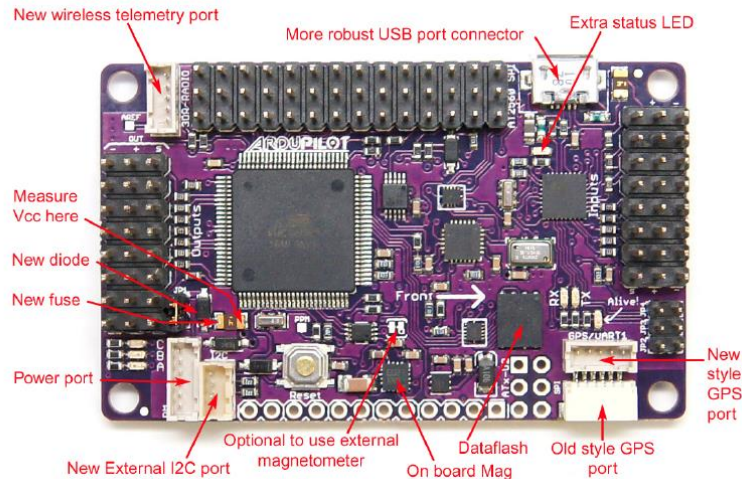
Una característica importante que poseen los helicópteros es el vuelo estacionario, es un modo de vuelo donde el helicóptero permanece suspendido en el aire sin variar la posición y la actitud, manteniendo los ángulos de cabeceo (pitch) y alabeo (roll) en valores muy cercanos a cero y el ángulo de guiñada (yaw) en un valor constante. Este modo de vuelo permite dejar a la aeronave flotando en el aire en un lugar fijo determinado.

1.2. ARDUPILOT MEGA

El Ardupilot Mega es un sistema de control y navegación empleado en vehículos aéreos no tripulados como aviones, helicópteros y drones. Este sistema está integrado con una unidad de medición inercial (IMU), esta unidad basada en el algoritmo DCM, nos permite obtener el valor de las fuerzas gravitacionales, la posición, la orientación y la velocidad para lo cual tiene incorporados ciertos sensores.

Entre los sensores de la tarjeta principal del Ardupilot Mega (Figura 6) se encuentran, un MPU-6000 de 6 ejes, el cual contiene un acelerómetro de 3 ejes y un giroscopio de 3 ejes; un HMC-5883 que contiene un compás digital que encuentra la desviación del punto móvil en grados con respecto al norte magnético, y un sensor de presión barométrica. También cuenta con una memoria interna tipo flash para el registro de datos, un microcontrolador Atmel, puertos de comunicación SPI, I2C, UART, entre otros, que permite conectar una antena de GPS que ubica el punto móvil con respecto a un punto global, un kit de telemetría y otras interfaces [5].

Figura 6. Placa Ardupilot Mega.



Fuente: [4]

2. MODELO MATEMÁTICO

“Un modelo matemático de un sistema dinámico se define como un conjunto de ecuaciones que representa la dinámica del sistema con precisión o al menos bastante bien” [23]. Para obtener este modelo se pueden adoptar dos enfoques [8]:

1. Partir de principios físicos (ecuaciones de sólido rígido, generación de pares y fuerzas de acuerdo a principios aerodinámicos,) para obtener un modelo en el que los parámetros tengan un sentido físico.
2. Utilizar un conjunto de expresiones sin sentido físico pero convenientemente ajustadas para reproducir el comportamiento entrada-salida del helicóptero.

2.1. MODELO TEÓRICO

El modelo matemático del helicóptero a radiocontrol se retoma de los trabajos [2] [21], [22] donde concluyeron que el modelado del helicóptero puede separarse en dos partes. La primera representa la dinámica del rotor principal y la segunda, la dinámica del rotor de cola.

Para realizar esta separación, se requiere las siguientes consideraciones [2]:

- El rotor principal está compuesto de dos palas sin movimiento de arrastre.
- El centro de masa del vehículo está localizado bajo el fuselaje del rotor.
- En vuelo estacionario la velocidad angular del rotor es constante.
- El rotor de cola está compuesto por dos palas y su centro de movimiento está localizado en el eje longitudinal del fuselaje.

A continuación, se presenta el modelo del helicóptero a radiocontrol basado en las consideraciones anteriores.

La dinámica del rotor principal se puede expresar a través de las siguientes ecuaciones:

Ecuación (1). Dinámica del rotor principal.

$$\begin{aligned} -\ddot{\Phi}-2\dot{\Omega}+\ddot{\beta}_c-2\dot{\beta}_s\Omega &= \frac{-2C_{LB}}{I_B} \left[\left(\frac{2}{3} B^3 R^3 \Omega \theta_0 - \frac{1}{2} B^2 R^2 v \right) (u_F - h\dot{\theta}) \right] + \frac{-2C_{LB}}{I_B} \left[\frac{1}{3} B^3 R^3 \Omega \beta_0 (v_F - h\dot{\Phi}) \right] \\ &+ \frac{-2C_{LB}}{I_B} \left[\left(\frac{1}{2} B^4 R^4 \Omega \theta_0 \beta_0 - \frac{1}{3} B^3 R^3 v \beta_0 \right) \dot{\theta} \right] + \frac{-2C_{LB}}{I_B} \left[\left(-\frac{1}{4} B^4 R^4 \Omega \dot{\Phi} + \frac{1}{4} B^4 R^4 \Omega^2 \theta_s \right) \right] \\ &+ \frac{-2C_{LB}}{I_B} \left[\frac{1}{4} B^4 R^4 \Omega (\beta_c + \beta_s \Omega) \right] - \frac{2K_B}{I_B} \beta_s \end{aligned}$$

$$\begin{aligned} -\ddot{\Phi}-2\dot{\Omega}+\ddot{\beta}_c-2\dot{\beta}_s\Omega &= \frac{-2C_{LB}}{I_B} \left[\left(\frac{2}{3} B^3 R^3 \Omega \theta_0 - \frac{1}{2} B^2 R^2 v \right) (u_F - h\dot{\Phi}) \right] + \frac{-2C_{LB}}{I_B} \left[\frac{1}{3} B^3 R^3 \Omega \beta_0 (v_F - h\dot{\theta}) \right] \\ &+ \frac{-2C_{LB}}{I_B} \left[\left(\frac{1}{2} B^4 R^4 \Omega \theta_0 \beta_0 - \frac{1}{3} B^3 R^3 v \beta_0 \right) \dot{\Phi} \right] + \frac{-2C_{LB}}{I_B} \left[\left(-\frac{1}{4} B^4 R^4 \Omega \dot{\theta} + \frac{1}{4} B^4 R^4 \Omega^2 \theta_c \right) \right] \\ &+ \frac{-2C_{LB}}{I_B} \left[\frac{1}{4} B^4 R^4 \Omega (\beta_c + \beta_s \Omega) \right] - \frac{2K_B}{I_B} \beta_c \end{aligned}$$

El movimiento de alabeo, que realiza los movimientos laterales a la izquierda y a la derecha, se representa por:

Ecuación (2). Movimiento de alabeo.

$$I_{\Phi} \ddot{\Phi} = -F_{Z0} \beta_s h - K_B \beta_s$$

El movimiento de cabeceo, que realiza los desplazamientos hacia delante y hacia atrás, se puede representar como:

Ecuación (3). Movimiento de cabeceo.

$$I_{\theta} \ddot{\theta} = -F_{Z0} \beta_c h - 2K_B \beta_c$$

Donde,

$$F_{z0} = -2C_{LB} \left(\frac{1}{3} B^3 R^3 \theta_0 - \frac{1}{2} B^2 R^2 \Omega v \right)$$

El movimiento de guiñada permite rotar el helicóptero y se representa de la siguiente forma:

Ecuación (4). Movimiento de guiñada.

$$I_{\Psi} \ddot{\Psi} = \left[-2C_L \left(\frac{c}{2a} B^4 R^4 \Omega - \frac{1}{3} B^3 R^3 v \theta_0 \right) - C_{LT} R_T^2 n \Omega h_T^2 \right] \dot{\Psi} + \left(\frac{2}{3} C_{LT} R_T^3 n^2 \Omega^2 \right) \theta_T$$

Donde:

$$C_{LT} = \frac{1}{2} \rho a_T c_T$$

$$C_L = \frac{1}{2} \rho a c$$

Tabla 3. Parámetros del modelo matemático.

Variable	Descripción	Variable	Descripción
Φ	Angulo de Alabeo	h	Distancia entre el centro de masa del helicóptero y el centro del rotor principal
Θ	Angulo de Cabeceo	h_T	Distancia entre el centro de masa del helicóptero y el centro del rotor de cola
Ψ	Angulo de guiñada	I_B	Momento de inercia del rotor principal
θ_c	Control de cabeceo	I_{ϕ}	Momento de inercia del cabeceo
θ_x	Control de alabeo	I_{θ}	Momento de inercia del alabeo
θ_T	Control de guiñada	I_{ψ}	Momento de inercia de la guiñada
θ_0	Control de paso colectivo	R	Radio del rotor principal
C_l	Coefficiente de Levantamiento rotor principal	R_T	Radio del rotor de cola
C_{lt}	Coefficiente de Levantamiento rotor de cola	Ω	Velocidad angular del rotor principal
β_c	Angulo de aleteo del Cabeceo	ρ	Densidad del aire
β_x	Angulo de aleteo del Alabeo	v	Velocidad inducida del motor principal
α	Coefficiente de inclinación del rotor principal	v_T	Velocidad inducida del motor de cola
α_T	Coefficiente de inclinación del rotor de cola	θ_0	Angulo de cabeceo de las palas principales. Promedio de cabeceo colectivo
B	Parte del rotor utilizado por la fuerza del despegue	m_F	Masa del vehículo
c	Anchura de las palas del rotor principal	β_0	Promedio del ángulo de aleteo
c_T	Anchura de las palas del rotor de cola		
n	Factor de reducción de la velocidad debido al efecto del suelo		

Fuente: [22]

Se observa que los modelos anteriores no son lineales, pero se pueden linealizar llevando el helicóptero a un punto de operación estable, como lo sería el vuelo estacionario, donde no hay movimiento lateral ni longitudinal, es decir ángulos de alabeo y cabeceo cercanos a cero, y el ángulo de guiñada en un valor constante. Para que no haya movimiento vertical se mantendrá la velocidad del rotor en un valor constante.

Para la ecuación del movimiento de guiñada (4) en vuelo estacionario conlleva a que varios parámetros como el ángulo de paso de la hélice del rotor principal (θ_0), el coeficiente de levantamiento (C_{LT}), la parte del rotor utilizado por la fuerza de despegue (B), la velocidad inducida (v) y el factor de reducción debido al efecto suelo (η), permanezcan constantes.

De acuerdo a lo anterior procedemos a calcular el equivalente en Laplace de la ecuación (4), tomando condiciones iniciales tanto de la velocidad como la posición de la cola en $t=0$ en cero, resultando la siguiente ecuación:

Ecuación 5. Función de transferencia del movimiento de guiñada.

$$\frac{\Psi(s)}{\theta_r(s)} = \frac{\left[\frac{\frac{2}{3} C_{LT} R_T^3 n^2 \Omega^2}{I_\Psi} \right]}{s^2 + \left[\frac{-2C_L \left(\frac{c}{2a} B^4 R^4 \Omega - \frac{1}{3} B^3 R^3 v \theta_0 \right) - C_{LT} R_T^2 n \Omega h_T^2}{I_\Psi} \right] s}$$

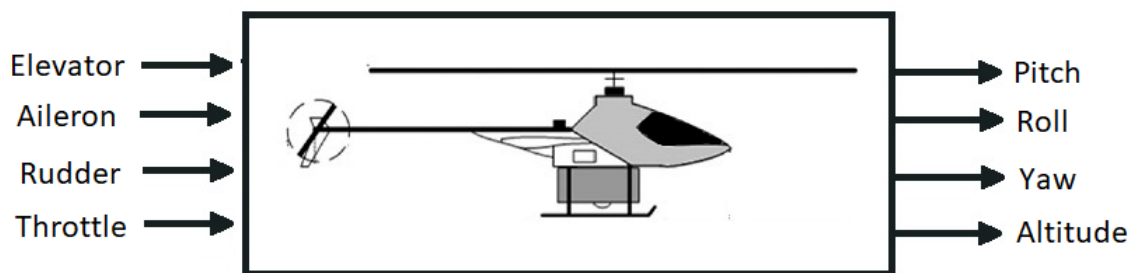
La función de transferencia anterior relaciona el ángulo de ataque de las palas del rotor de cola con el eje de desviación en el eje de guiñada. En este caso, el actuador (servomotor de cola) opera directamente sobre el ángulo de ataque de las palas del rotor de cola.

Hallar experimentalmente los parámetros de la función de transferencia descrita en la ecuación (5), puede tener cierta dificultad dado el hecho que estamos utilizando un helicóptero de bajo costo disponible comercialmente. En este caso un modelo experimental a partir de las relaciones entrada-salida sería una buena solución. La función de transferencia descrita en ecuación 5 resulta útil como una guía para el resultado del proceso de identificación.

2.2. MODELO EXPERIMENTAL

Los helicópteros son vehículos que pueden ser representados como sistemas complejos no lineales con múltiples entradas y múltiples salidas (MIMO) (Figura 7) y con fuertes modos acoplados lo que hace que su modelo matemático y control presenten un alto grado de dificultad [2].

Figura 7. Sistema MIMO.



Fuente: Autor.

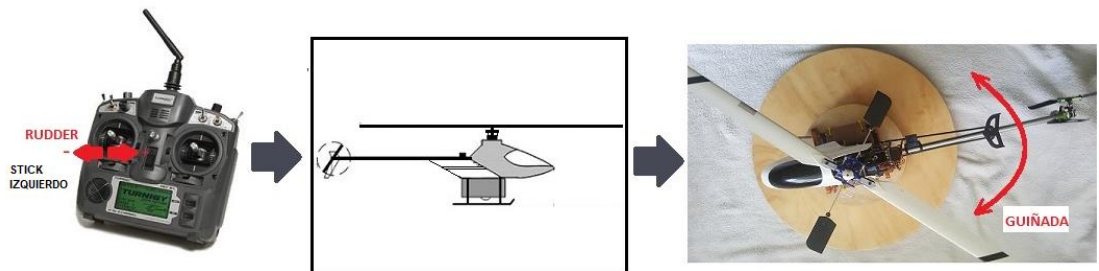
Con el fin de reducir la complejidad del modelo de un helicóptero, en los trabajos [6], [7], [8], [9], [12], [14] se toma en consideración un desacople del sistema anteriormente descrito asumiendo modelos simplificados de una entrada y una

salida (SISO), donde cada entrada de control afecta un solo parámetro del helicóptero a radiocontrol, esta suposición es válida especialmente para el movimiento de guiñada ya que no está influenciado por los otros tres movimientos (balanceo, cabeceo y altitud).

Esta consideración aplica para el vuelo estacionario donde el helicóptero se mantiene flotando en el aire con movimiento vertical y longitudinal nulos, y actitud constante.

Para el caso de este proyecto se identificó y validó el modelo del eje de guiñada en vuelo estacionario (Figura 8), que relaciona el movimiento de rotación sobre el eje Z del helicóptero. Si se define una relación entrada-salida se debe considerar como entrada la señal de control (ciclo útil de la señal PWM) al servomotor de cola y como salida el ángulo de guiñada, el giro del helicóptero sobre su eje vertical.

Figura 8. Sistema SISO del eje de guiñada.



Fuente: Autor.

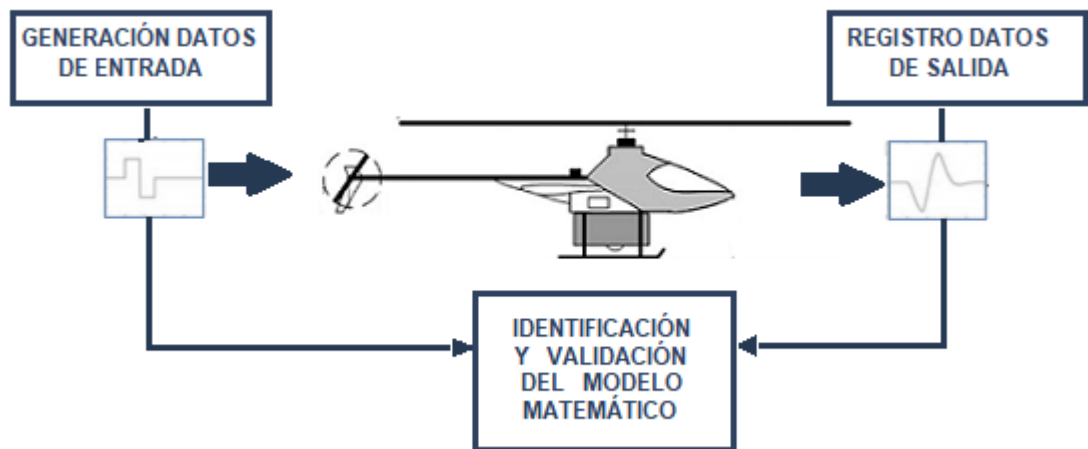
3. METODOLOGÍA

El presente capítulo documenta la metodología para realizar los experimentos de vuelo que llevan a la obtención y procesamiento de datos con la finalidad de identificar y validar un modelo del helicóptero radiocontrolado en vuelo estacionario.

3.1. ESQUEMA GENERAL DEL PROCEDIMIENTO

En la figura 9 se presenta el esquema general que resume el procedimiento experimental desarrollado para aproximar el modelo matemático de la planta. A continuación, se explican cada una de las etapas correspondientes.

Figura 9. Esquema general.



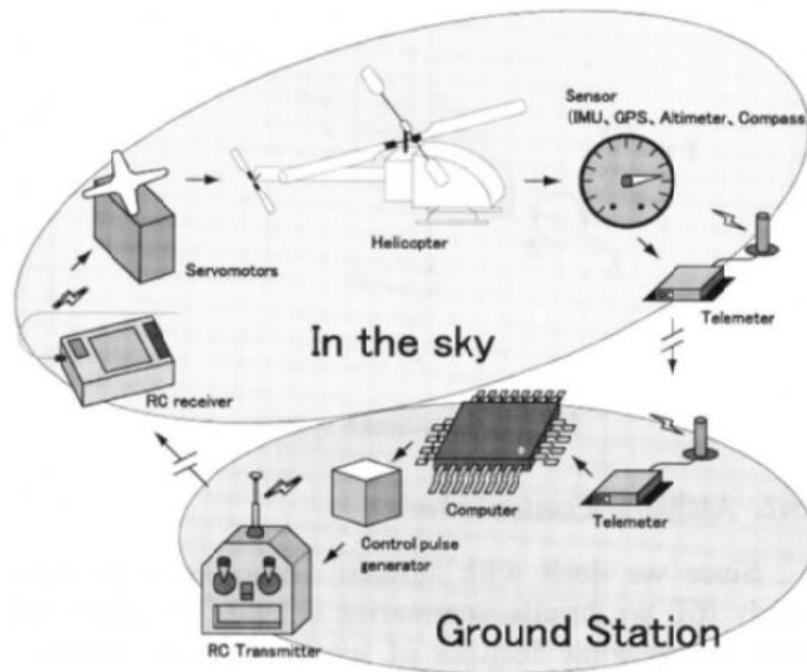
Fuente: Autor

- **Generación datos de entrada:** En esta etapa se construye una señal con la mayor cantidad de componentes frecuenciales para ser aplicada en la entrada, buscando excitar la mayor cantidad de dinámicas en el sistema y generar la mayor cantidad de información en la salida. Lo anterior corresponde a una señal de tipo binaria pseudoaleatoria programada en la plataforma de desarrollo Arduino Uno y aplicada al helicóptero mediante un módulo de comunicación de radiofrecuencia. En la sección 3.4 se explicará con más detalle.
- **Registro datos de salida:** En esta etapa se realiza el almacenamiento de datos de salida del sistema ante el estímulo de la señal de entrada, en términos prácticos la señal obtenida de los sensores del Ardupilot se envía a un PC mediante radios de telemetría. En la sección 3.5 se explicará con más detalle.
- **Identificación y validación del modelo matemático:** Empleando técnicas de análisis estadístico es posible inferir la relación matemática entre los vectores de entrada y salida obtenidos para el sistema en los dos procedimientos descritos anteriormente. Este proceso se desarrolló fuera de línea haciendo uso de la herramienta de identificación de Matlab® (IDENT – System identification toolbox [10], [11]).

3.2. MONTAJE EXPERIMENTAL

En este capítulo se presenta de manera detallada el montaje y los procedimientos para la puesta a punto del experimento, tomaremos como base el esquema presentado en el trabajo de Hazawa et al [1] que es presentado a continuación.

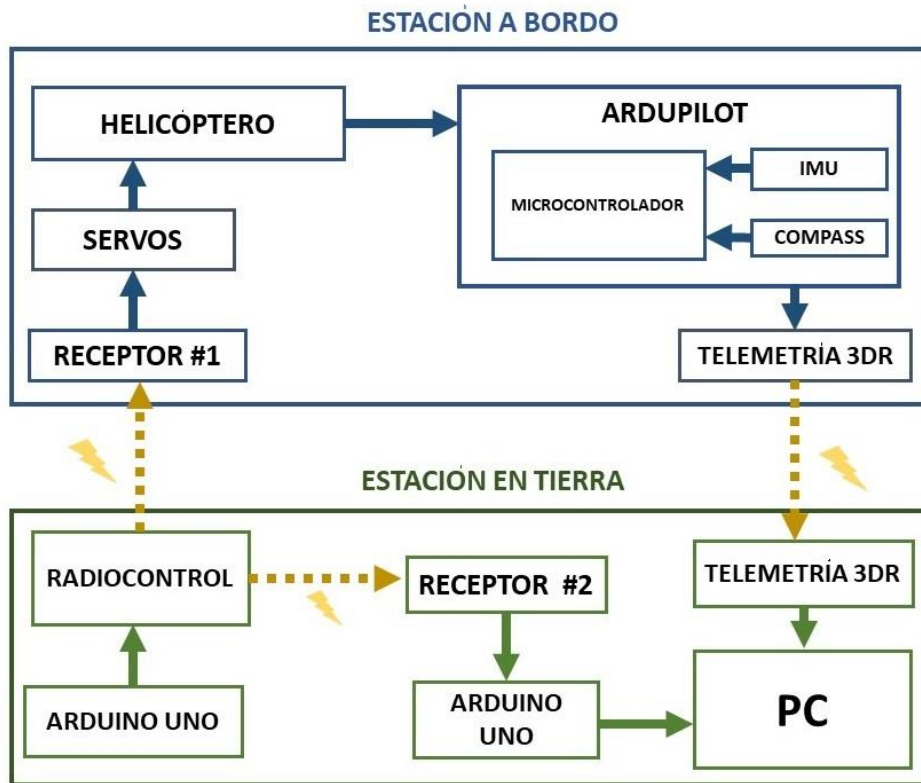
Figura 10: Esquema propuesto por Hazawa.



Fuente: [1]

Fue construido un diagrama de bloques (Figura 11) para la mejor interpretación del montaje, esta dividido en una estación de tierra y una estación a bordo, que serán explicados a continuación.

Figura 11. Esquema de bloques.



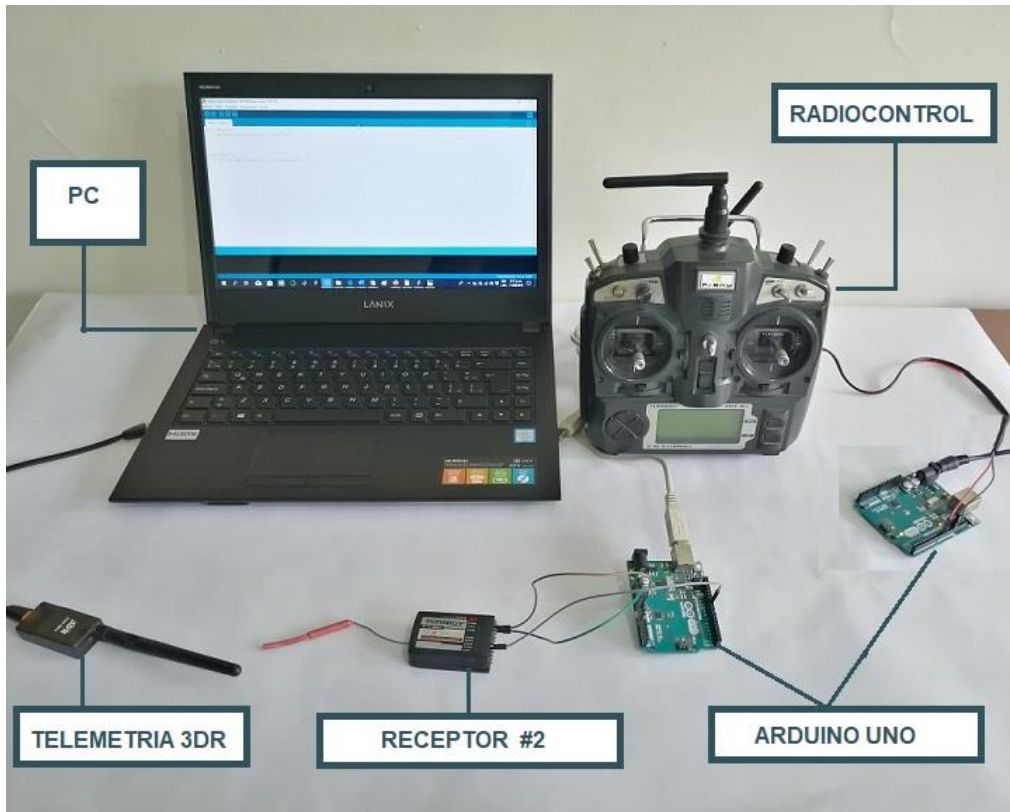
Fuente: Autor.

3.2.1. Estación de tierra

La estación de tierra (Figura 12) permite al piloto controlar el helicóptero, además se encarga de recibir la información que percibe la aeronave para su posterior análisis. Esta estación cuenta con un computador donde se almacenarán y visualizarán los datos. Dos placas de Arduino Uno, plataforma de desarrollo donde en una de ellas se diseñará la señal de excitación, mientras la otra placa se utiliza para leer los datos del receptor #2. Emisora de radiocontrol Turnigy 9x [17] necesaria para establecer la comunicación entre la aeronave y la estación de tierra, es la encargada de enviar la señal de excitación al helicóptero, además se utiliza

para que el piloto tome el control de la aeronave en caso que ocurra algo inesperado. Radio de telemetría 3DR, encargada de crear el enlace de datos en tiempo real entre la estación de tierra y el Helicóptero, la conexión se da por medio de micro-USB y su frecuencia de operación es de 433 MHz [13],

Figura 12. Estación de tierra



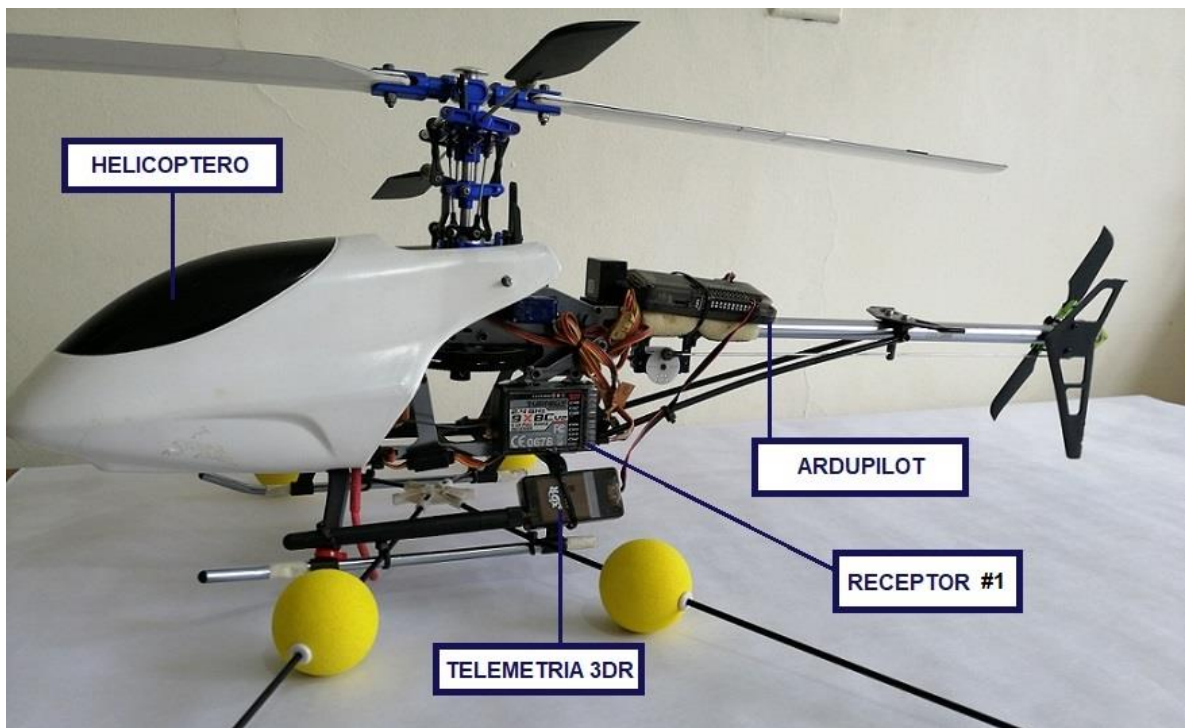
Fuente: Autor.

3.2.2. Estación a bordo

En términos de instrumentación el helicóptero cuenta con un receptor (receptor #1), dispositivo que establece la comunicación entre la estación de tierra y el helicóptero, recibe y decodifica una señal enviada por el radiocontrol y transmite esta

información a los servos. La placa Ardupilot, cuenta con los sensores necesarios para obtener información sobre la dinámica del helicóptero. El radio de telemetría 3DR, encargada de crear el enlace de datos en tiempo real entre el helicóptero y la estación tierra, la comunicación con el Ardupilot se realiza a través del protocolo UART y su frecuencia de operación es de 433 MHz [13].

Figura 13. Estación a bordo.



Fuente: Autor.

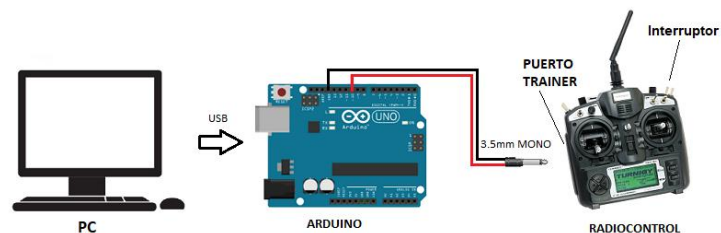
3.3. GENERACIÓN DE DATOS DE ENTRADA

La generación de datos de entrada al helicóptero es una tarea primordial en la identificación de sistemas, ya que dicha señal es la encargada de perturbar el sistema en estudio permitiendo excitar los modos dinámicos que estamos buscando y que aparecerán en los datos de salida.

Para aplicar la señal de excitación se creó una trama de datos compatible con el protocolo de comunicación del radiocontrol Turnigy 9X [17]. Se constituyó una señal PPM (Modulación por Posición de Pulso) empleando programación en la plataforma Arduino (Anexo 1). Esta secuencia binaria pseudoaleatoria garantiza cambios rápidos en la señal de entrada y niveles límites de actuación.

La transmisión de estos estímulos a los actuadores (servomotores) se realizó con el radiocontrol Turnigy 9X [17], empleando la función TRAINER (Figura 14) la cual permite elegir que canales (rudder, aileron, pitch, elevator, throttle) son controlados por la señal proveniente de la plataforma de desarrollo Arduino, este cambio se realiza mediante un interruptor, el cual al ser activado detiene las señales de los sticks y da paso a la señal del Arduino.

Figura 14. Acople de la señal de excitación al Radiocontrol.

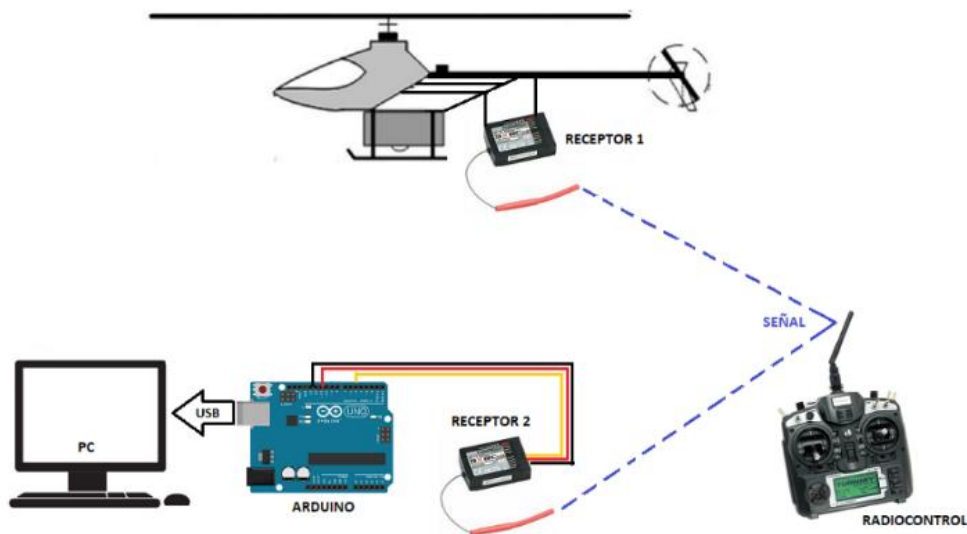


Fuente: Autor.

En el caso del modelo del eje de guiñada el canal Rudder será controlado por la señal del Arduino.

Para el proceso de identificación debemos registrar como datos de entrada el ciclo útil de la señal PWM que entrega el receptor a cada uno de los servomotores, para lograr lo anterior se aplica el método del doble receptor [18] donde se utilizan dos receptores idénticos que reciben una señal simultanea del radiocontrol. Un receptor (llamado receptor #1) se ubica en la estación a bordo y otro receptor (receptor #2) en la estación de tierra. El receptor #2 se conecta a una placa de desarrollo Arduino Uno y mediante programación (Anexo B) se registran los datos del canal requerido. La figura 15 ilustra este procedimiento.

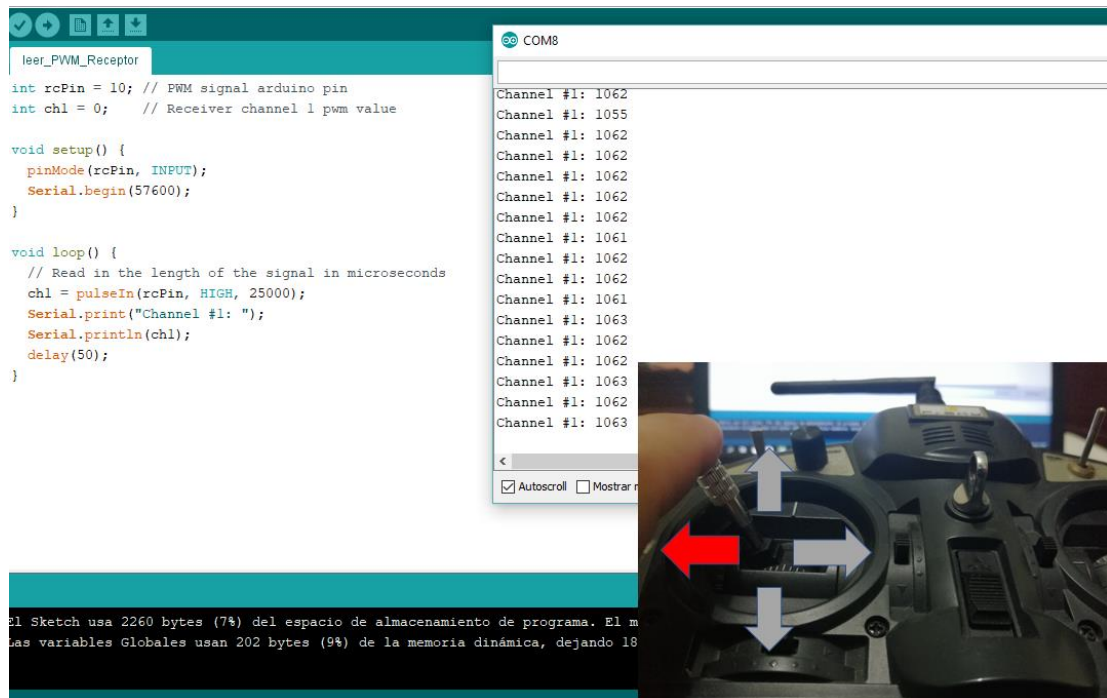
Figura 15. Método del doble receptor.



Fuente: Modificado de [18].

Los datos de entrada registrados (figura 16) consisten en el ciclo útil de la señal PWM que se aplica al servo, estos datos varía de 1.500 μ s (máxima posición del servo a la izquierda) a 2.000 μ s (máxima posición del servo a la derecha).

Figura 16. Visualización de datos de entrada a los servomotores



Fuente: Autor.

3.4. REGISTRO DE DATOS DE SALIDA

Para el seguimiento del comportamiento del helicóptero en vuelo estacionario, es necesario la medición de una serie de variables características del mismo, en el caso de este proyecto se requiere la medición de la posición angular en el eje de guiñada. Los sensores básicos para este fin son:

- Acelerómetro de 3 ejes: mide aceleraciones en los 3 ejes del helicóptero.
- Giróscopo de 3 ejes: sensor utilizado para medir velocidades angulares.
- Magnetómetro de 3 ejes: con este dispositivo se mide las direcciones del campo magnético de la Tierra.

Estos sensores vienen encapsulados en un solo dispositivo llamado Ardupilot, explicado en la sección 2.4, y que hace parte de este estudio. Para obtener datos de salida requeridos para la identificación del modelo se requiere información de la dinámica del helicóptero, en este caso el ángulo de guiñada, para esto se retoma el trabajo presentado en [19] donde se implementó una unidad de medición inercial (IMU) en el Ardupilot con información proveniente de los sensores de la tarjeta (giróscopo, acelerómetro, magnetómetro) para ello, se utilizará en primer lugar el entorno de desarrollo de Arduino IDE, que permitirá cargar el programa (Anexo C) en el microcontrolador del Ardupilot.

Para registrar los datos de salida provenientes del helicóptero cuando está en vuelo, es necesario establecer una comunicación inalámbrica entre este y la estación tierra, para ello emplea un sistema de transmisión / recepción por medio de la telemetría 3DR [13].

Figura 17. Visualización datos de salida del Ardupilot.

```
IMU_apm_v3
//-----//
//----- IMU Ardupilot 2.6 -----//
//-----//

// ARDUPILOT 2.5 / MPU6000 / HMC5883L
// los valores de los angulos medidos
// estan en funcion de la posicion
// el ardupilot 2.5

// roll => eje y
// pitch => eje x
// yaw => eje z

// librerias
#include <SPI.h>
#include "Wire.h"
#include "I2Cdev.h"
#include "HMC5883L.h"
#include <math.h>

// valores offset - ver primero calibracion
// offset magnetometro
#define offsetmx -36.5

//-----//
// ANGULOS MEDIDOS //
ROLL: -30.261871
PITCH: 23.922363
YAW: 31.945923

//-----//
// MEDIDOR DE VARIABLES - IMU -//
// MPU6000 / HMC5883L -//
//-----//

El Sketch usa 10074 bytes (3%) del espacio de memoria disponible.
Las variables Globales usan 607 bytes de memoria, dejando 3933 bytes libres.
Guardando sketch en "IMU_apm_v3.ino".
```

Fuente: Autor

3.5. BANCO DE PRUEBAS

El vuelo de un helicóptero de radiocontrol proporciona un problema de control desafiante con maniobras agresivas que pueden ocasionar accidentes en el piloto y daños en el vehículo aéreo. Para solventar las dificultades anteriores se propone un esquema con dos bancos de pruebas que limitan los grados de libertad del helicóptero y conseguir maniobras de vuelo estacionario más estables, facilitando una configuración segura para realizar experimentos en un laboratorio. Estos bancos de pruebas que restringen el movimiento del helicóptero a los grados de libertad de interés se explican a continuación.

3.5.1. BANCO DE PRUEBAS 1-DOF

Este banco de pruebas (Figura 18) es una plataforma circular giratoria de madera, tiene un grado de libertad en la que sólo es posible el movimiento de guiñada, permite al helicóptero un movimiento de rotación de 360°.

Figura 18. Helicóptero en el banco de pruebas 1-DOF



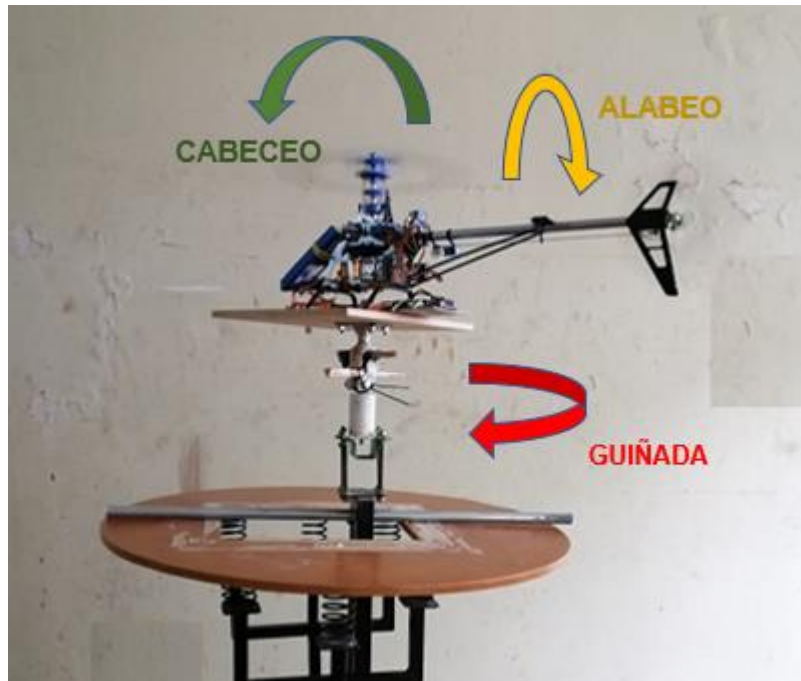
Fuente: Autor

3.5.2. BANCO DE PRUEBAS 3-DOF

Este banco de pruebas (Figura 19) permite un movimiento para 3 grados de libertad (alabeo, cabeceo y guiñada) del helicóptero de radiocontrol, esto se logra por medio

de un mecanismo formado por una bandeja de sujeción, un cardán y dos rodamientos. La bandeja tiene la función de sujetar al helicóptero, el cardán permite movimiento en dos ejes (alabeo y cabeceo) y los rodamientos dan movimiento en el eje de guiñada, completando así los 3 grados de libertad.

Figura 19. Banco de pruebas 3-DOF



Fuente: Autor.

4. RESULTADOS EXPERIMENTALES

En este capítulo se presentan los resultados obtenidos tras aplicar la metodología descrita en el capítulo 3, para obtener el modelo de la dinámica de un helicóptero a radiocontrol en vuelo estacionario. Dado que los datos de entrada y los datos de salida se miden con diferentes sistemas (datos de entrada provenientes del receptor #1 y datos de salida provenientes del Ardupilot), se desarrolla un esquema de sincronización para construir los conjuntos de datos utilizados para identificar la dinámica del helicóptero. Este esquema de sincronización está basado en el lenguaje Python (Anexo D) estos datos se almacenan en un archivo de texto Excel que serán exportados a Matlab y mediante el Toolbox de Identificación [10],[11] se realiza la identificación y validación del sistema.

4.1. MODELO DEL ANGULO DE GUIÑADA.

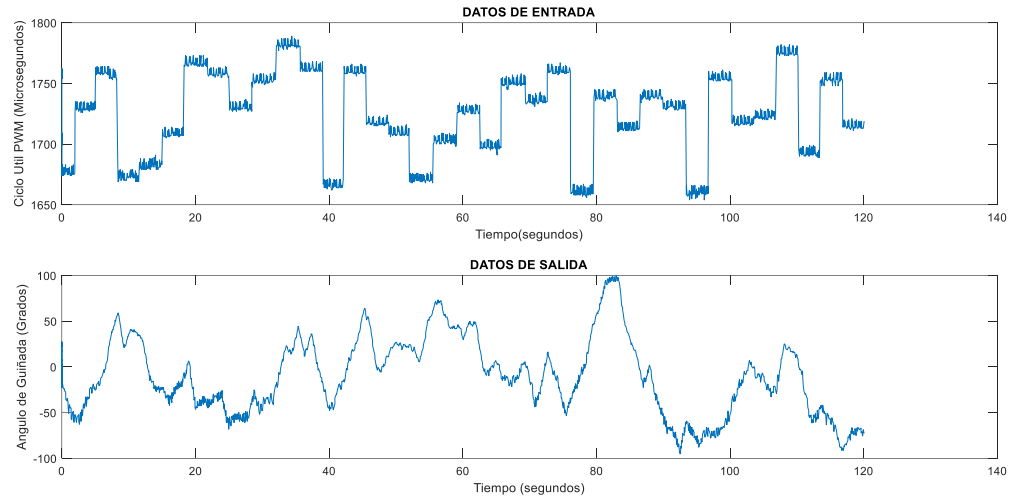
4.1.1. Identificación y validación en banco 1-DoF

Aplicando los pasos presentados en el esquema de la figura 9, la primera etapa consiste en la generación de datos de entrada y salida. Para el modelo del ángulo de guiñada corresponden a:

1. Datos de entrada: ciclo útil se la señal PWM del servo de cola.
2. Datos de salida: ángulo de guiñada (yaw) proveniente de los sensores del Ardupilot.

Estas señales se presentan en la figura 20.

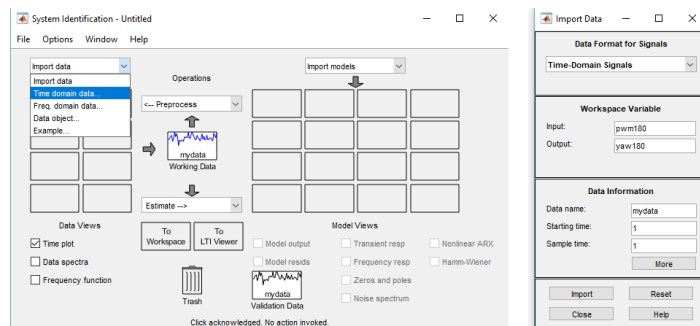
Figura 20. Señales de excitación y respuesta eje de guiñada, banco 1-DoF.



Fuente: Autor.

Para el procedimiento de identificación y validación del modelo que relaciona el ciclo útil de la señal PWM de entrada al servo de cola y el ángulo de guiñada, datos mostrados en la figura 20, se utiliza la herramienta de identificación de sistemas de Matlab (System Identification Toolbox – IDENT) (figura 21). Donde se selecciona la mitad inicial de los datos como conjunto de estimación y la segunda mitad como conjunto de validación.

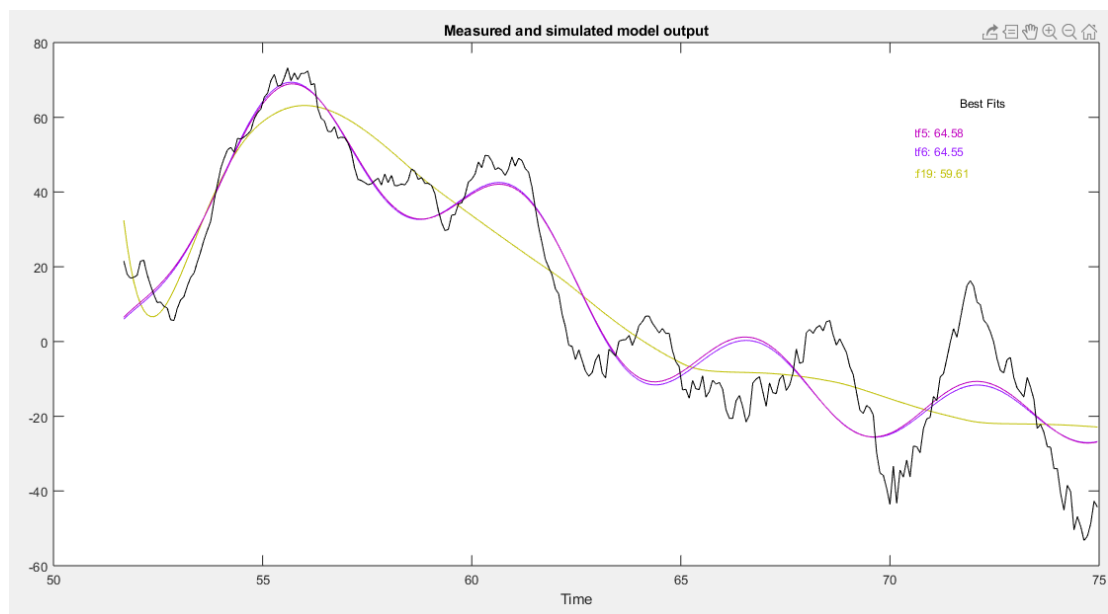
Figura 21. Herramienta de identificación de sistemas de Matlab.



Fuente: Autor.

Con la información de cada una de las pruebas anteriores y por medio del 'Toolbox System Identification' de Matlab se procesaron los datos para encontrar la mejor aproximación al comportamiento del sistema, para ello se realizaron varias estimaciones para cada señal del experimento encontrando la mejor para cada señal, a continuación se presentará los mejores ajustes encontrados para la señal.

Figura 22. Correlación de modelos, banco 1-DoF



Fuente: Autor.

Tabla 4. Mejor ajuste para el conjunto de datos, banco 1-DoF.

Función de Transferencia	Best fit
Poles: 4 zeros:1	64,58
Poles: 4 zeros:2	64,55
Poles: 3 zeros:1	59,61

Fuente: Autor.

La elección del modelo se basó en el alto índice de correlación (best fit), como se observa en la Tabla 4, la correlación más alta (64,58) se obtuvo para la función de transferencia con 4 Polos y 1 Cero. Por tanto, el modelo matemático como mejor estimación de la relación entrada – salida es:

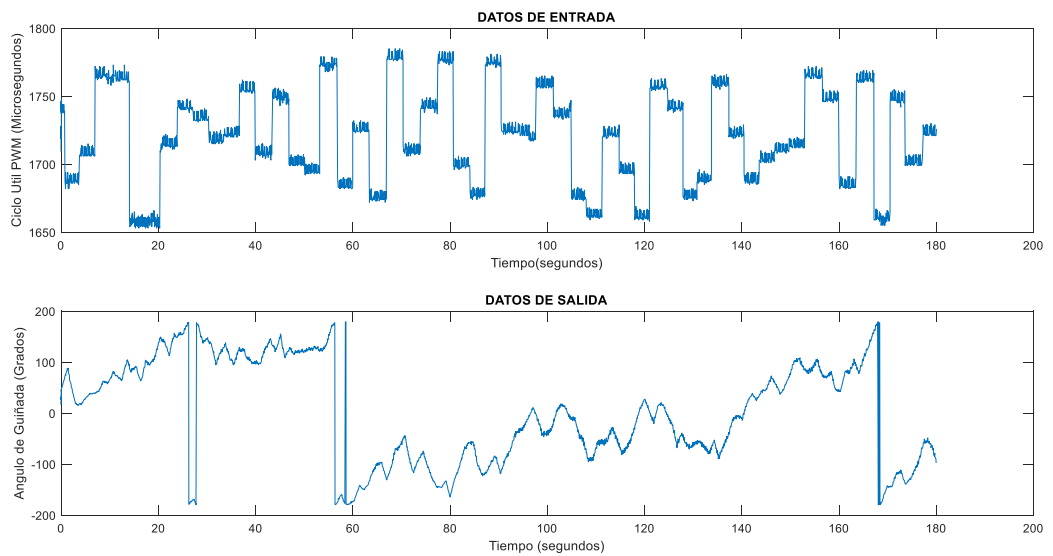
Ecuación 6. Función de transferencia para el ángulo de guiñada en el banco 1-DOF.

$$H1(s) = \frac{0.07641 * s - 0.0007705}{s^4 + 0.3071 * s^3 + 1.403 * s^2 + 0.4131 * s + 0.07791}$$

4.1.2. Identificación y validación en banco 3-DoF

Para el caso del vuelo experimental en el banco 3-Dof, se obtienen las siguientes señales de excitación y respuesta:

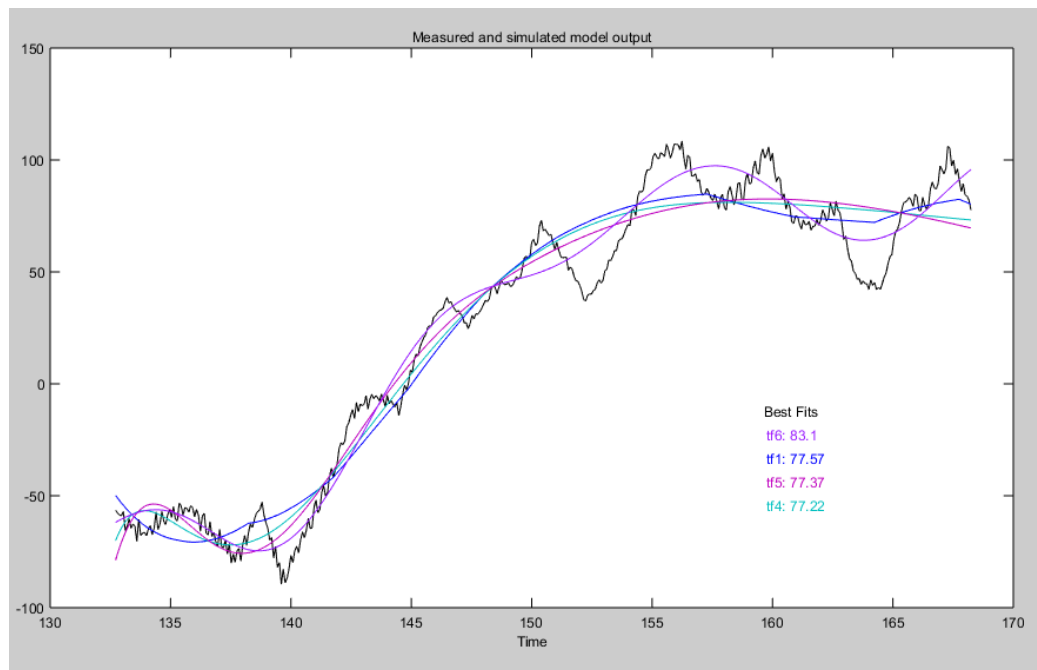
Figura 23. Señales de excitación y respuesta eje de guiñada, banco 3-DoF.



Fuente: Autor.

Aplicando el mismo procedimiento que el numeral anterior, se obtiene la relación de transferencia entre el ángulo de guiñada y el ciclo útil de entrada al servomotor de posicionamiento del eje de guiñada, obteniéndose la función de transferencia de la ecuación 7. Este modelo matemático fue obtenido por considerarse el mejor ajuste, como se muestra en la tabla 4.

Figura 24. Correlación de modelos, Banco 3-DoF.



Fuente: Autor.

Tabla 5. Mejor ajuste para el conjunto de datos, banco 3-DoF.

Función de transferencia	Best Fits
Poles: 4 zeros:1	83,1
Poles: 2 zeros:1	77,57
Poles: 4 zeros:0	77,37
Poles: 3 zeros:1	77,22

Fuente: Autor

Ecuación 7. Función de transferencia ángulo de guiñada banco 3-DOF.

$$H2 = \frac{-0.01729 * s + 0.0002444}{s^4 + 0.1569 * s^3 + 0.3163 * s^2 + 0.04647 * s + 0.005969}$$

5. CONCLUSIONES

- Se realizó la selección de un modelo matemático para el helicóptero a radiocontrol, para el caso de este proyecto fue la dinámica del eje de guiñada, el cual determina la orientación del helicóptero en vuelo estacionario. Además el estudio este modelo es importante para el desarrollo de sistemas de control que permitan un comportamiento más estable de la aeronave.
- Se estudió un modelo teórico de la dinámica del eje de guiñada del helicóptero a radiocontrol y se usó como guía para el proceso de identificación.
- Se evaluó e implementó un sistema de adquisición de datos aprovechando los diferentes sensores que posee la tarjeta Ardupilot.
- Se realizó un procedimiento experimental donde se identificó y validó el modelo matemático adecuado para la dinámica del eje de guiñada de un helicóptero a radiocontrol basándose en datos de vuelos reales.
- El helicóptero es un sistema con múltiples entradas y múltiples salidas, pero en ciertos puntos de operación como el vuelo estacionario es posible separarlo en subsistemas menos complejos como es el caso del modelo dinámico del eje de guiñada.
- Los resultados (Ecuaciones 6 y 7) en el banco de pruebas 1-DOF y el banco de pruebas 3-DOF permitió encontrar un modelo matemático para el eje de guiñada muy similar. En el banco de pruebas 1-DOF se muestra el porcentaje de aproximación (64,58%) del modelo de la función de transferencia obtenido con un número de polos igual a 4 y un número de ceros igual a 1. En el banco

de pruebas 3-DOF un porcentaje de aproximación de 83,1% y una función de transferencia con 4 polos y 1 cero.

- La dinámica inestable del helicóptero a radiocontrol impone limitantes en la longitud y tipo de entradas durante la recolección de datos. Para registros largos, las acciones de estabilización provenientes del piloto son necesarios para prevenir que la aeronave pierda el control y ocurran accidentes.
- Las características dinámicas del helicóptero a radiocontrol cambian significativamente de una condición de operación a otra. La recolección de datos se desarrolló en condiciones de operación específicas como lo es el vuelo estacionario, razón por la cual se utilizaron dos bancos de pruebas, lo que garantizó un vuelo estacionario más cercano al real.

REFERENCIAS BIBLIOGRÁFICAS

[1] Hazawa, K., J. Shin, D. Fujiwara, K. Igarashi, D. Fernando and K. Nonami (2004b). Autonomous flight control of hobby-class small unmanned helicopter (modeling based on experimental identification and autonomous flight control experiments). Nippon Kikai Gakkai Ronbunshu, C Hen/Transactions of the Japan Society of Mechanical Engineers, Part C 70(3), 720–727.

[2] Aguirre-Gil, I., Barrientos, A. y Del Cerro, J. (2006). Attitude control of a minihelicopter in hover using different types of control. Revista técnica de la facultad de ingeniería Universidad del Zulia. Pag 209.

[3] Michael Svenstrup and Kasper Winther. Autonomous hover of an electric helicopter. Master's thesis, Aalborg University: Intelligent Autonomous Systems, 2007.

[4] PEÑALOZA BORRÁS, Iván Darío. JAIMES NIÑO, Iván Mauricio. Diseño e implementación de un algoritmo de rutas para el vuelo de un quadrotor. Tesis de pregrado de ingeniería electrónica. Bucaramanga: Universidad Industrial de Santander. Escuela de ingeniería eléctrica, electrónica y telecomunicaciones, 2013.

[5] JAIMES SANCHEZ, Miguel. SEPULVEDA PEÑA, Cesar Javier. Evaluación de la solución Ardupilot Mega 2.5 para la implementación de algoritmos de control en aeroplanos a escala no tripulados (UAV). Tesis de pregrado de ingeniería

electrónica. Bucaramanga: Universidad Industrial de Santander. Escuela de ingeniería eléctrica, electrónica y telecomunicaciones, 2013.

[6] GUERRERO NOBOA, Freddy Roberto. MENENDEZ GRANIZO, Oswaldo Aníbal. Modelación, simulación y control de sistemas aéreos no tripulados utilizando inteligencia artificial. Tesis de pregrado de ingeniería electrónica y control. Quito: Escuela Politécnica Nacional. Facultad de Ingeniería Eléctrica y Electrónica, 2013, 181p.

[7] LEYTON, Hernando. Desarrollo, implementación y prueba de un filtro de Kalman del tipo UKF para un vehículo aéreo no tripulado. Tesis de Maestría en Matemáticas Aplicadas. Medellín: Universidad EAFIT. Departamento de Ciencias Básicas, 2009, 67p.

[8] Béjar, M. Ollero, A. "Modelado y control de helicópteros autónomos. Revisión del estado de la técnica". (2008).

[9] Sendoya-Losada, D. Quintero-Polanco, J. "PID controller applied to an unmanned aerial vehicle". (2018).

[10] L. Ljung. System Identification Toolbox™ Getting Started Guide. Disponible en: https://la.mathworks.com/help/pdf_doc/ident/ident_gs.pdf

[11] L. Ljung. System Identification Toolbox™ User's Guide. Disponible en: https://la.mathworks.com/help/pdf_doc/ident/ident.pdf

[12] Kim, S; & Tilbury, D. "Mathematical modeling and experimental identification of a model helicopter". (1998).

[13] Telemetría 3DR. Disponible en: <https://3dr.com/wp-content/uploads/2013/10/3DR-Radio-V2-doc1.pdf>

[14] Daigo FUJIWARA, Jinok SHIN, Kensaku HAZAWA, Kenzo NONAMI, H^∞ Hovering and Guidance Control for Autonomous Small-Scale Unmanned Helicopter, Transactions of the Japan Society of Mechanical Engineers Series C, 2004, Volume 70, Issue 694, Pages 1708-1714, Released March 04, 2011.

[15] Cai G, Chen BM, Lee TH (2011) Unmanned rotorcraft systems. Springer, Berlín. Pag 56.

[16] Pestún, I., Miroslav, H., & Kurčík, P. "Control of an RC helicopter model through USB interface". (2011).

[17] Radiocontrol Turnigy 9x. Disponible en: <https://hobbyking.com/media/file/725056143X2037269X20.pdf>

[18] Ramaswamy, Nikhil et al. "Development of an Avionics System for Flight Data Collection of an UAV Helicopter" (2011).

[19] BERMUDEZ MARQUEZ, Gabriela. NIÑO DIAZ, Carlos Augusto. Implementación de una unidad de medición inercial (IMU) en Ardupilot 2.5. Tesis de pregrado de ingeniería electrónica. Bucaramanga: Universidad Industrial de Santander. Escuela de ingeniería eléctrica, electrónica y telecomunicaciones, 2014.

[20] Helicóptero HK450. Disponible en:

<https://hobbyking.com/es-es/hk450-ccpm-3d-helicopter-kit-align-t-rex-compat-ver-2.html>

[21] PINZON GOMEZ, Sergio Andrés. TORRES JEREZ, Ricardo. Implementación del sistema de control de cola de un helicóptero RC. Tesis de pregrado de ingeniería electrónica. Bucaramanga: Universidad Industrial de Santander. Escuela de ingeniería eléctrica, electrónica y telecomunicaciones, 2010.

[22] D´CROZ BARON, David Felipe. VILLAMIZAR DELGADO, Sergio Iván. Diseño y construcción de un sistema de control de posición de cola de un helicóptero de aeromodelismo. Tesis de pregrado de ingeniería electrónica. Bucaramanga: Universidad Industrial de Santander. Escuela de ingeniería eléctrica, electrónica y telecomunicaciones, 2009.

[23] Ogata, Katsuhico. Ingeniería de control moderno, Universidad de Minnesota, Prentice Hall, quinta edición, 1996.

[24] RETANA PEREZ, David. VARGAS MORENO, Aldo Enrique. Estabilización de un helicóptero a escala mediante sistemas neuro-difusos. Tesis de pregrado de ingeniería mecatrónica. Ciudad de México: Universidad Nacional Autónoma de México. Facultad de ingeniería, 2010, 137p.

[25] BUENO ALMEIDA, Julián Alberto. ULLOA FIGUEROA, Mario Andrés. Diseño y construcción de un hexarotor no tripulado de navegación autónoma. Tesis de pregrado de ingeniería mecánica. Bucaramanga: Universidad Industrial de Santander. Escuela de ingeniería mecánica, 2015.

[26] ¿Cómo funciona un servo?. Disponible en:
<https://www.mecatronicalatam.com/es/tutoriales/motor/motores-electricos/motor-de-corriente-continua/servomotor/>

[27] FERNÁNDEZ GÓMEZ, Wilmar Darío. Estudio de la factibilidad de la construcción de un mini helicóptero como aplicación de la fotogrametría en soluciones de ingeniería civil. Tesis de maestría en ingeniería civil. Bogotá: Universidad de los Andes. Departamento de ingeniería civil y ambiental, 2003, 81p.

ANEXOS

ANEXO A. CÓDIGO GENERACIÓN SEÑAL DE ENTRADA (PPM)

```
//int inPin = 2;
//int buttonVal = 0;
int outPinPPM = 10;

int pitchAnalog = 0;
uint16_t pitch_uS = 0;
int rollAnalog = 0;
uint16_t roll_uS = 0;
int buttonButton = 0;
uint16_t button_uS = 0;

uint16_t FIXED_uS_LOW = 300;
uint16_t PULSE_uS_MIN = 650;
uint16_t PULSE_uS_MAX = 1700;

ISR(TIMER1_COMPA_vect) {
  ppmoutput ();
}

void setup () {
  Serial.begin(57600);
  randomSeed (analogRead (0));
  // pinMode (inPin, INPUT);

  pinMode (outPinPPM, OUTPUT);

  //Setup Timer
  TCCR1A = B00110001;
  TCCR1B = B00010010;
  TCCR1C = B00000000;
  TIMSK1 = B00000010;
  TIFR1  = B00000010;
  OCR1A  = 22000;
  OCR1B  = 1000;
}

void loop() {

  pitchAnalog = random (400,650);
  rollAnalog = random (50,800);
  //550 es el punto central
  //90 sentido contrario manecillas del reloj
  //800 sentido manecillas del reloj
```

```

pitch_uS = pitchAnalog + PULSE_uS_MIN;
Serial.println(pitch_uS);
roll_uS = rollAnalog + PULSE_uS_MIN;

pitch_uS = constrain (pitch_uS, PULSE_uS_MIN, PULSE_uS_MAX);
roll_uS = constrain (roll_uS, PULSE_uS_MIN, PULSE_uS_MAX);
button_uS = constrain (button_uS, PULSE_uS_MIN, PULSE_uS_MAX);
delay (500);
}

void ppmoutput (){
  // Channel 1 - Alerón
  digitalWrite(outPinPPM, LOW);
  delayMicroseconds (FIXED_uS_LOW);
  digitalWrite (outPinPPM, HIGH);
  delayMicroseconds(roll_uS);

  // Channel 2 - Elevador
  digitalWrite (outPinPPM, LOW);
  delayMicroseconds (FIXED_uS_LOW);
  digitalWrite (outPinPPM, HIGH);
  delayMicroseconds(roll_uS);

  // Channel 3 - Throttle
  digitalWrite (outPinPPM, LOW);
  delayMicroseconds (FIXED_uS_LOW);
  digitalWrite (outPinPPM, HIGH);
  delayMicroseconds(pitch_uS);

  // Channel 4 - Rudder
  digitalWrite (outPinPPM, LOW);
  delayMicroseconds (FIXED_uS_LOW);
  digitalWrite (outPinPPM, HIGH);
  delayMicroseconds(pitch_uS);

  // Channel 5
  digitalWrite (outPinPPM, LOW);
  delayMicroseconds (FIXED_uS_LOW);
  digitalWrite (outPinPPM, HIGH);
  delayMicroseconds(pitch_uS);

  // Channel 6 - Button
  digitalWrite (outPinPPM, LOW);
  delayMicroseconds (FIXED_uS_LOW);
  digitalWrite (outPinPPM, HIGH);
  delayMicroseconds(button_uS);

  // Channel 7
  digitalWrite (outPinPPM, LOW);
  delayMicroseconds (FIXED_uS_LOW);
  digitalWrite (outPinPPM, HIGH);
  delayMicroseconds(roll_uS);

  // Channel 8

```

```

digitalWrite (outPinPPM, LOW);
delayMicroseconds (FIXED_uS_LOW);
digitalWrite (outPinPPM, HIGH);
delayMicroseconds (pitch_uS);

//Synchro pulse
digitalWrite (outPinPPM, LOW);
delayMicroseconds (FIXED_uS_LOW);
digitalWrite (outPinPPM, HIGH);
}

```

ANEXO B. CÓDIGO LECTURA SEÑAL PWM DEL RECEPTOR.

```

int rcPin = 10; // PWM signal arduino pin
int ch1 = 0;    // Receiver Chanel 1 pwm value

void setup () {
  pinMode (rcPin, INPUT);
  Serial.begin(57600);
}

void loop () {
  // Read in the length of the signal in microseconds
  ch1 = pulseIn (rcPin, HIGH, 25000);
  Serial.print("Channel #1: ");
  Serial.println(ch1);
  delay (50);
}

```

ANEXO C. CÓDIGO DE LECTURA DE SENSORES EN EL ARDUPILOT.

```
//-----//
//----- IMU Ardupilot -----//
//-----//

// ARDUPILOT / MPU6000 / HMC5883L
// los valores de los ángulos medidos mediante la imu
// están en función de la posición de los sensores en
// el ardupilot

// roll => eje y
// pitch => eje x
// yaw => eje z

// librerias
#include <SPI.h>
#include "Wire.h"
#include "I2Cdev.h"
#include "HMC5883L.h"
#include <math.h>

// valores offset - ver primero calibracion
// offset magnetometro
#define offsetmx -36.5
#define offsetmy -59
#define offsetmz -60.5
// offset acelerometro
#define offsetax 80.0
#define offsetay 93.0
#define offsetaz -273.0
// offset giroscopio
#define offsetgx -68.0
#define offsetgy -1.0
#define offsetgz 4.0

#define escalamx 0.0024
#define escalamy 0.0025
#define escalamz 0.0032

// señales y variables
HMC5883L magnetometro;

double offset[6];
char str[100];

double t_now = 0.0;
double alpha = 0.97;
const int cs = 53;
double giro_x = 0;
double giro_y = 0;
```

```

double giro_z = 0;
double ace_x = 0;
double ace_y = 0;
double ace_z = 0;
double dt = 0.0;
double fx, fy, fz, gx, gy, gz;
double accel_angley, accel_anglex;
double angle_giro_x, angle_giro_y;
double pitch, roll, yaw;
double mx, my, mz;
int16_t mag_x1, mag_y1, mag_z1;
double mag_x, mag_y, mag_z;
double cal1, cal2;
double mag_xn, mag_yn, mag_zn;
uint16_t roll_m, pitch_m, yaw_m;

// delay mecanico
unsigned long previousMillis = 0;
const long interval = 80;

//-----//
//----- ENTIDAD -----//
void setup()
{
    // puerto serial
    Serial.begin(57600);

    // pines
    pinMode(40, OUTPUT);
    digitalWrite(40, HIGH);

    // configuration SPI
    SPI.begin();
    SPI.setClockDivider(SPI_CLOCK_DIV16);
    SPI.setBitOrder(MSBFIRST);
    SPI.setDataMode(SPI_MODE0);
    delay(100);

    pinMode(cs, OUTPUT);

    // configuration MPU6000
    configuracion_MPU6000();

    // inicio magnetometro
    Wire.begin();
    magnetometro.initialize();

    // calibracion MPU6000 / HMC5883L
    // imu_calibracion();
    // Serial.println("listo");
    // delay(3000);
    // HMC_calibracion();

```

```

}

//-----//
//----- ARQUITECTURA -----//
void loop()
{
  // LECTURA DE DATOS GIR/ACCEL/MAGN
  giro_x = (GyroX() - offsetgx) / 16.4;
  giro_y = (GyroY() - offsetgy) / 16.4;
  giro_z = ((GyroZ() - offsetgz) / 16.4);

  ace_x = (AcceX() - offsetax) / 4096.0;
  ace_y = (AcceY() - offsetay) / 4096.0;
  ace_z = ((AcceZ() - offsetaz) / 4096.0);

  magnetometro.getHeading(&mag_x1, &mag_y1, &mag_z1);
  mag_x = ((double)(mag_x1) - (offsetmx)) * escalamx;
  mag_y = ((double)(mag_y1) - (offsetmy)) * escalamy;
  mag_z = ((double)(mag_z1) - (offsetmz)) * escalamz;
  mag_xn = -1 * mag_x;
  mag_yn = -1 * mag_y;
  mag_zn = -1 * mag_z;

  // filtro complementario
  fx = ace_x * alpha + (fx * (1 - alpha));
  fy = ace_y * alpha + (fy * (1 - alpha));
  fz = ace_z * alpha + (fz * (1 - alpha));

  gx = giro_x * alpha + (gx * (1 - alpha));
  gy = giro_y * alpha + (gy * (1 - alpha));
  gz = giro_z * alpha + (gz * (1 - alpha));

  mx = mag_xn * alpha + (mx * (1 - alpha));
  my = mag_yn * alpha + (my * (1 - alpha));
  mz = mag_zn * alpha + (mz * (1 - alpha));

  // calculos de valores ROLL - PITCH - YAW
  accel_angley = (atan2(-fx, fz) * 180.0) / M_PI;
  accel_anglex = (atan2(-fy, sqrt(pow(fx, 2) + pow(fz, 2))) * 180.0) /
M_PI;

  dt = (micros() - t_now) / 1000000.0;
  t_now = micros();

  roll = alpha * (roll + gy * dt) + (1 - alpha) * accel_angley; //
roll [-180,180]
  pitch = alpha * (pitch + gx * dt) + (1 - alpha) * accel_anglex; //
pitch [-90,90]

  call = ((-mx * cos(roll * M_PI / 180.0)) + (mz * sin(roll * M_PI /
180.0)));
  cal2 = ((my * cos(-pitch * M_PI / 180.0)) + (mx * sin(roll * M_PI /
180.0) * sin(-pitch * M_PI / 180.0)) + (mz * cos(roll * M_PI / 180.0) *
sin(-pitch * M_PI / 180.0)));

```

```

yaw = (-1 * atan2(-cal1, cal2)) * 180 / M_PI;

// muestreo de datos para envio por serial
roll_m = (roll + 360.0) * (pow (2, 14) - 1) / 550.0;
pitch_m = (pitch + 360.0) * (pow (2, 14) - 1) / 550.0;
yaw_m = ( yaw + 360.0 ) * (pow(2, 14) - 1) / 550.0;

// envio de datos
// delay minimo

unsigned long currentMillis = millis();

if ((currentMillis - previousMillis) >= interval)
{
    previousMillis = currentMillis;
    //Serial.print("A"); Serial.print(roll); Serial.print(",B");
Serial.print(pitch); Serial.print(",C"); Serial.println(yaw);
    sprintf(str, "A%dB%dC%dD", roll_m, pitch_m, yaw_m);
    Serial.println(str);
}
//delay(80);

}

//-----//
//----- CONFIGURACION MPU6000 -----//
void configuracion_MPU6000()
{
    // configuración de los parámetros del integrado
    // MPU6000

    // reset device
    SPIwrite (0x6B, 0x80); // reset
    delay (200);

    // activar, seleccionar reloj giroz
    SPIwrite(0x6B, 0x03); // reloj pllgiroz
    delay(200);

    // desactivar I2C
    SPIwrite(0x6A, 0x03); // desactivar I2C
    delay(200);

    // muestreo a 1kHz/(19+1)=50Hz
    SPIwrite(0x19, 19); // frecuencia muestreo a 50Hz
    delay(200);

    // filtro digital pasabaja de 20Hz

```

```

SPIwrite(0x1A, 0x04);
delay(200);

// escala de giro a 2000dps (°/s)
SPIwrite(0x1B, 0x18);
delay(200);

// escala de aceleración a 8G (4096 lsb/g)
SPIwrite (0x1C, 0x10);
delay (200);
}

//-----//
//----- CALIBRACION MPU6000 -----//
void imu_calibracion ()
{
    signed int offgx = 0;
    signed int offgy = 0;
    signed int offgz = 0;
    signed int offax = 0;
    signed int offay = 0;
    signed int offaz = 0;

    delay(50);

    // calculo de los valores offset de la IMU
    GyroX (); GyroY (); GyroZ ();
    AcceX (); AcceY (); AcceZ ();

    for (int i = 0; i < 100; i++)
    {
        offgx += GyroX ();
        offgy += GyroY ();
        offgz += GyroZ ();

        offax += AcceX ();
        offay += AcceY ();
        offaz += AcceZ ();
        delay (10);
    }

    // promedios:
    offset[0] = offax / 100;
    offset[1] = offay / 100;
    offset[2] = (4096 - offaz) / 100;

    offset[3] = offgx / 100;
    offset[4] = offgy / 100;
    offset[5] = offgz / 100;

    Serial.println(" VALORES OFFSET CALCULADOS: ");
    Serial.print(" offax: "); Serial.print(offset[0]);
    Serial.print(" offay: "); Serial.print(offset[1]);

```

```

Serial.print(" offaz: "); Serial.println(offset[2]);

Serial.print(" offgx: "); Serial.print(offset[3]);
Serial.print(" offgy: "); Serial.print(offset[4]);
Serial.print(" offgz: "); Serial.println(offset[5]);
Serial.println("listo");
while (1)
{
}
}

//-----//
//---- CALIBRACION HMC5883L ----//
void HMC_calibracion()
{
  int i = 0;
  do
  {
    magnetometro.getHeading(&mag_x1, &mag_y1, &mag_z1);
    mag_x = ((double)(mag_x1) - (-36.5)) * 0.0024;
    mag_y = ((double)(mag_y1) - (-59)) * 0.0025;
    mag_z = ((double)(mag_z1) - (-60.5)) * 0.0032;
    Serial.print(mag_x); Serial.print(","); Serial.print(mag_y);
Serial.print(","); Serial.println(mag_z);
    delay(100);
    i++;
  } while (i < 300);
  Serial.println("listo");
  // while (1)
  // {
  // }

}

//-----//
//---- ESCRIBIR SPI ----//
void SPIwrite(byte reg, byte data)
{
  uint8_t dat;
  digitalWrite(cs, LOW);
  dat = SPI.transfer(reg);
  dat = SPI.transfer(data);
  digitalWrite(cs, HIGH);

}

//-----//
//---- LEER SPI ----//
uint8_t SPIread(byte reg)
{
  uint8_t dat;
  uint8_t return_value;
  uint8_t addr = reg | 0x80;

```

```

    digitalWrite(cs, LOW);
    dat = SPI.transfer(addr);
    return_value = SPI.transfer(0x00);
    digitalWrite(cs, HIGH);
    return (return_value);
}

// GIROX
int GyroX()
{
    uint8_t GyroX_H = SPIread(0x43);
    uint8_t GyroX_L = SPIread(0x44);
    int16_t GGyroX = GyroX_H << 8 | GyroX_L;
    return (GGyroX);
}

// GIROY
int GyroY()
{
    uint8_t GyroY_H = SPIread(0x45);
    uint8_t GyroY_L = SPIread(0x46);
    int16_t GGyroY = GyroY_H << 8 | GyroY_L;
    return (GGyroY);
}

// GIROZ
int GyroZ()
{
    uint8_t GyroZ_H = SPIread(0x47);
    uint8_t GyroZ_L = SPIread(0x48);
    int16_t GGyroZ = GyroZ_H << 8 | GyroZ_L;
    return (GGyroZ);
}

// ACELX
int AcceX()
{
    uint8_t AcceX_H = SPIread(0x3B);
    uint8_t AcceX_L = SPIread(0x3C);
    int16_t AAcceX = AcceX_H << 8 | AcceX_L;
    return (AAcceX);
}

// ACELY
int AcceY()
{
    uint8_t AcceY_H = SPIread(0x3D);
    uint8_t AcceY_L = SPIread(0x3E);
    int16_t AAcceY = AcceY_H << 8 | AcceY_L;
    return (AAcceY);
}

// ACELZ
int AcceZ()

```

```

{
  uint8_t AcceZ_H = SPIread(0x3F);
  uint8_t AcceZ_L = SPIread(0x40);
  int16_t AAcceZ = AcceZ_H << 8 | AcceZ_L;
  return (AAcceZ);
}

```

ANEXO D. CODIGO PHYTON LECTURA DE DATOS

Editor de Spyder

Este es un archivo temporal.

"""

```

import serial
import time
import msvcrt
import csv

```

```

#arduino.close()
#ardupilot.close()

```

```

print("Abriendo puerto serie...")

```

```

arduino = serial.Serial('COM8',57600, timeout=1)
ardupilot = serial.Serial('COM6',57600, timeout=1)

```

```

print("Puerto Inicializado Arduino")
print("Puerto Inicializado Ardupilot")

```

```

archivo_PWM = open("yaw.csv", "a") #nombre del archivo de salida

```

```

dato_PWM = arduino.readline()
dato_yaw = ardupilot.readline()
time.sleep(5)

```

```

print("YA")

```

```

start=time.time()

```

```

e_time=0

```

```

time_f=30 #duracion de la toma de datos

```

```

while e_time<=time_f:

```

```

dato_PWM = arduino.readline()
dato_yaw = ardupilot.readline()

dato_yaw=str(dato_yaw)
dato_PWM=str(dato_PWM)

dato_PWM=dato_PWM.split("'")
PWM=dato_PWM[1].split("\\")

dato_yaw = dato_yaw.rstrip("\r\n")
dato_yaw=dato_yaw.replace("B","A").replace("C","A").replace("D","A")
dato_yaw = dato_yaw.rstrip("A").lstrip("A")
angulos=dato_yaw.split("A")

roll=int(angulos[1])
pitch=int(angulos[2])
yaw=int(angulos[3])

roll = -360 + 550*(int(roll))/16383
pitch = -360 + 550*(int(pitch))/16383
yaw = -360 + 550*(int(yaw))/16383

"""
print("roll: ",roll)
print("pitch: ",pitch)
print("yaw: ",yaw)
"""

e_time=time.time()-start

#archivo_PWM.write("%s; %s; %s\n"% (roll, PWM[0],e_time))
#archivo_PWM.write("%s; %s; %s\n"% (pitch, PWM[0],e_time))
archivo_PWM.write("%s; %s; %s\n"% (yaw, PWM[0],e_time))

#time.sleep(1.3)

print("Listo")
arduino.close()
ardupilot.close()
archivo_PWM.close()

```