Facultad de Ingeniería

Trabajo de Investigación

“Diseño de un sistema de estabilización basado en Arduino para un avión no tripulado con motor eléctrico”

Autor: Quispe Curi, Pedro - 1523018

Para obtener el Grado de Bachiller en:

Ingeniería Aeronáutica

Lima, Diciembre del 2019

ANEXO 6

Declaración de Autenticidad y No Plagio
(Grado Académico de Bachiller)

Por el presente documento, yo Pedro Quispe Cury,
identificado/a con DNI N° 72685095, egresado de la carrera de Ingeniería Aeronautica,
informo que he elaborado el Trabajo de Investigación denominado "Diseño de un sistema de estabilización basado en Arduino para un avión no Tripulado con motor eléctrico",
para optar por el Grado Académico de Bachiller en la carrera de Ingeniería Aeronautica,
declaro que este trabajo ha sido desarrollado íntegramente por el/los autor/es que lo suscribe/n y afirmo que no existe plagio de ninguna naturaleza. Así mismo, dejo constancia de que las citas de otros autores han sido debidamente identificadas en el trabajo, por lo que no se ha asumido como propias las ideas vertidas por terceros, ya sea de fuentes encontradas en medios escritos como en Internet.

Así mismo, afirmo que soy responsable solidario de todo su contenido y asumo, como autor, las consecuencias ante cualquier falta, error u omisión de referencias en el documento. Sé que este compromiso de autenticidad y no plagio puede tener connotaciones éticas y legales. Por ello, en caso de incumplimiento de esta declaración, me someto a lo dispuesto en las normas académicas que dictamine la Universidad Tecnológica del Perú y a lo estipulado en el Reglamento de SUNEDU.

Miercoles, 13 de Noviembre de 2019

(firma)

RESUMEN

El diseño de un sistema de estabilización para un avión no tripulado con motor eléctrico desarrollado en este trabajo, consistió en cinco capítulos, en el primer capítulo se recolectó información relacionada al tema de investigación. Posteriormente, en el segundo capítulo se estudió los principios físicos de los UAV, se investigó sobre los sistemas de control electrónicos, las propiedades y funcionalidades de las plataformas electrónicas Arduino, sensores y servomotores.

El tercer capítulo es la metodología de esta investigación, cuya primera parte consiste en identificar los parámetros de diseño, y según estos definir los componentes que serán parte del dispositivo electrónico. Conocidos e identificados los elementos, se selecciona mediante una tabla de comparaciones los componentes cuyas propiedades se ajustan mejor a las necesidades de diseño planteadas. La segunda parte de la metodología se enfoca en el diseño e implementación del código general de programación que amalgama la parte mecánica con la parte electrónica del dispositivo. Este código general de programación es la combinación de ecuaciones matemáticas y algoritmos de control, que según el tipo de UAV varía la estructura de este.

Posterior a la implementación del código en el microprocesador de la plataforma Arduino, se procedió con la afinación de las variables del controlador PID, lo cual se hizo de manera experimental, evaluando el tiempo y la velocidad de reacción de los servomotores a determinadas variaciones de medidas del IMU. En el cuarto capítulo se exponen los resultados obtenidos, donde se compara las gráficas para distintos códigos incluido el

control PID. En las conclusiones, se logró materializar el dispositivo a un costo muy accesible, donde la principal funcionalidad es corregir las desviaciones en el eje longitudinal y transversal del UAV, aunque con una penalidad de masa y volumen.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABLAS

INTRODUCCIÓN

Los sistemas de control y estabilización son dispositivos que están ganando importancia en el campo de la automatización de vuelos de UAV's (Unmanned Aerial Vehicle), debido a que se encargan de corregir desviaciones de trayectoria generadas por efectos de perturbaciones aerodinámicas como las ráfagas de viento. La presencia de estas perturbaciones se da en la mayor parte del vuelo de los UAV's (generalmente vuelo en crucero y "hover" para multirotores). Hoy en día, numerosas investigaciones nacionales e internacionales concentran su atención en la optimización de estos dispositivos, mediante la aplicación de modelamientos matemáticos de las leyes de la Mecánica de Vuelo en los UAV's, y la automatización y control de estos sistemas mediante dispositivos electrónicos.

Un considerable número de trabajos de investigación se enfocan en el desarrollo de sistemas de vuelo autónomo que permiten interconectar a dos o más UAV. Para realizar vuelos en distintas formaciones y en diferentes escenarios, mediante la recolección de datos de posición entre cada vehículo y manteniendo su estabilidad por medio de un controlador PID (proporcional, integral y derivativo). Otra investigación se centró en el diseño e implementación de un sistema de control de bajo costo económico para un UAV tipo cuadrotor, mediante el uso de un controlador PID para cada rotación del UAV (alabeo, cabeceo y guiñada) y lo complementó con el filtro de Kalman para suavizar la reacción de los actuadores frente a las perturbaciones en el vuelo. Por otro lado, dada la desventaja que supone la utilización del controlador PID en sistemas no lineales como son los UAV's, otros investigadores desarrollan pilotos automáticos, mediante el control de lógica difusa,

que consiste en comparar variables ideales con variables experimentales registradas por sensores.

Otro método de control usado en los sistemas autopilotos es el PID-difuso, que combina el control PID y el control de lógica difusa, y que aprovecha las principales ventajas como la capacidad de predicción del primero y la compatibilidad para implementarse en sistemas no lineales del segundo control.

En base a los antecedentes mencionados, este trabajo de investigación enfoca su atención en el uso de la plataforma electrónica Arduino UNO como parte fundamental del sistema de estabilización, con algoritmos de programación que incluirán el control PID que ejecutarán el accionamiento de los servos según los datos de giro y aceleración registrados por la unidad de medida inercial. Además, la estructura del hardware del sistema está compuesta por una unidad de medición inercial MPU 6050 y servomotores.

Se plantearon los siguientes objetivos de esta investigación:

Objetivo general

Diseñar un sistema de estabilización basado en Arduino para un avión no tripulado con motor eléctrico.

Objetivos específicos

- Implementar un sistema de estabilización basado en Arduino para un avión no tripulado con motor eléctrico, que cumpla con las mismas funciones que los sistemas de estabilización convencionales.
- Determinar el hardware del sistema de estabilización basado en Arduino con componentes presentes en el mercado nacional para remplazar a los sistemas de control convencionales.

- Desarrollar software del sistema de estabilización con menor presupuesto al costo de adquisición de un sistema convencional.

Finalmente, el alcance de esta investigación se alinea en el desarrollo de un sistema de estabilización, con componentes electrónicos disponibles en el mercado peruano y mediante el uso de algoritmos PID para un UAV de alas fijas. La implementación y desarrollo de un sistema de comunicación no son considerados en el desarrollo de este trabajo. Debido a que, en el proceso de investigación del proyecto, se descubrió que la plataforma Arduino UNO es un microcontrolador ATMEGA 328 de ejecución secuencial de algoritmos lo cual limita el procesamiento de datos de dos o más algoritmos, mientras que los autopilotos cuentan con procesadores más potentes y con mayor memoria. Asimismo, la falta de laboratorios de computo debidamente implementados es otra limitante, ya que la dificultad de la programación en Arduino se incrementa cuando cuándo las horas de practica en el tema son reducidas.

CAPÍTULO 1

ANTECEDENTES DE LA INVESTIGACIÓN

Debido al constante avance tecnológico e integración de los diferentes campos de la ingeniería, como la electrónica, robótica, mecánica y aeronáutica, actualmente, se diseña y construye máquinas capaces de realizar vuelos remotamente controlados o cada vez más autónomos [1]. Considerando prescindibles en ambos casos, la intervención de una tripulación encargada del control del UAV (Unmanned Aerial Vehicle).

Para el vuelo autónomo o remotamente controlado de los UAV, los sistemas de control están compuestos por elementos eléctricos y electrónicos, que permiten vuelos estables durante el desarrollo de la misión. Los componentes principales son sensores, actuadores (motores y servomotores), controladores electrónicos de velocidad, autopilotos, GPS y baterías.

Con el transcurrir de los años, el desarrollo de nuevos dispositivos electrónicos que permitan la ejecución autónoma de actividades aéreas en la industria militar y civil ha ido evolucionando hasta el grado de desplazar casi por completo la intervención humana, salvo en momentos donde la toma de decisiones sea de vital importancia. Para ilustrar esto, en el año 2006 se publicó una patente para el desarrollo de un sistema de autopiloto programable, que permite la conexión inalámbrica entre la estación en tierra y el UAV. Asimismo, la patente incluye métodos de procesamiento, procedimientos de comando para el procesamiento de comunicación y un proceso de estimación de la altura de vuelo [2]. El proceso de estimación de la altura de vuelo se considera una función muy importante para

efectos de esta investigación, ya que con esta función el autopiloto puede ajustar adecuadamente la altura de vuelo al nivel deseado. Por otro lado, la comunicación inalámbrica en tiempo real es una función que se ha ido extendiendo en los últimos años y que es más relacionada al sistema de comunicación lo cual se puede prescindir de esta característica en la investigación, aunque su implementación aportaría mayor precisión en el vuelo de los UAV.

Los autopilotos son dispositivos electrónicos encargados de controlar el vuelo de los UAV, mediante algoritmos matemáticos como el control “Proporcional, Integral y Derivativo”, la lógica difusa, entre otros. Donde cada tipo de controlador implementado tiene un funcionamiento especifico. Para validar lo mencionado, en el año 2018 se comparó diferentes tipos de controladores, con la finalidad de seleccionar el de mejores prestaciones para un cuadricóptero, siendo los parámetros de comparación el tiempo de respuesta y el exceso a la perturbación. Los controladores evaluados fueron el control PID, un control PID modificado y el control de lógica difusa, donde según los resultados el PID modificado tuvo el mejor desempeño [3]. Se considera que la modificación de un control convencional se realizó específicamente para un UAV tipo cuadricóptero, siendo necesario reestructurar el diagrama del control PID para implementarlo a un UAV de alas fijas.

Igualmente, aprovechando las ventajas del control PID en el año 2012, se desarrolló diseñó e implementó un sistema de control para un cuadricóptero, mediante 4 controladores PID (proporcional, integral y derivativo) para los movimientos de cabeceo, alabeo, guiñada y elevación vertical. El método de ajuste de las 12 constantes del PID, fue la evolución diferencial, que consiste en la optimización de una población de soluciones mediante su combinación o mutación con generaciones anteriores. En la investigación se usó una placa de control Gumstix, una placa de expansión Summit (que brinda la posibilidad de implementar nuevos dispositivos e instrumentos, sin cambiar la estructura actual), sensor

IMU CHR-6d y un sensor ultrasónico MB310. El diseño del software se hizo con la implementación de 3 aplicaciones; lenguaje C (para el control y configuración del multicóptero), lenguaje Python y framework QT (compatible con Linux y usado para el ajuste de las constantes PID) y el ultimo lenguaje fue Java. Además, gracias a la memoria RAM de la placa de control Gumstix fue posible la ejecución simultanea de varias operaciones. Por otra parte, la implementación de la evolución diferencial, para ajustar las constantes PID, ayudo a ajustar la población de valores ideales a pocas generaciones de haberse inicializado el algoritmo [4]. El uso de componentes electrónicos como la memoria RAM, el sensor IMU y entre otros son elementos considerablemente costosos, a pesar de incrementar la eficiencia en el dispositivo.

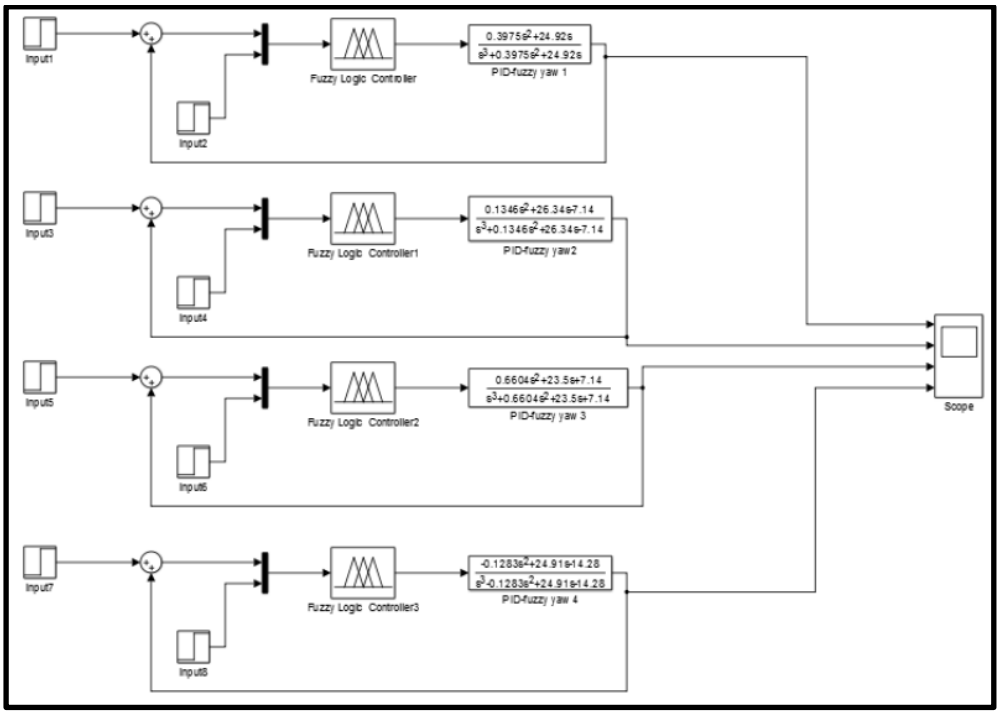
En el año 2012, Paredes [5] desarrolló una investigación basado en el control dinámico de un UAV mediante el uso de sensores combinados a un sistema de control de lógica difusa. Como parte de la implementación el proyecto tenía un GPS (sistema de posicionamiento global) para medir la trayectoria ejecutada durante el vuelo y compararla con la trayectoria programada antes del vuelo. Esto es debido a que el control de lógica difusa requiere de ciertos parámetros de entrada para aproximar los niveles de membresía de los parámetros de salida según un rango de valores que van de 0-1. El método de desarrollo de la investigación consistió en la simulación en Matlab, de las ecuaciones diferenciales linealizadas y no linealizadas de la dinámica de vuelo del UAV. Para el desplazamiento del UAV en el eje Z (ascenso o descenso mediante el timón de profundidad) la máxima deflexión de la superficie de control fue de -10° a 10°, mientras que para las guiñadas se consideró -8° a 8°. Posteriormente se continuó con el diseño del software y elección de hardware, donde se usó un módulo DSP TMS320F2812, servomotor, GPS, giróscopos, compás y sensores de aceleración. Finalmente, las pruebas experimentales se realizaron en un avión Cessna 182 en escala 1:5 (Ver figura 1), donde el sistema de control funcionó bien en el vuelo de autónomo de una pequeña ruta pre-programada [5]. La implementación

de la lógica difusa ayudara a controlar el comportamiento de la aeronave e incrementa la precisión del mismo, ya que el algoritmo acerca el valor medido a un valor deseado. Una desventaja de este controlador es su nivel de complejidad, que dificulta aún más el código de programación.

Figura 1: Modelo a escala 1:5 de un Cessna 182 [4].

En el año 2015 en la ciudad de Bogotá (Colombia), se diseñó e implementó un controlador PID-difuso para un cuadricóptero. Donde se combinó el control PID (que sigue principios lineales) y el control de lógica difusa. El análisis de las ecuaciones de la dinámica de vuelo del cuadricóptero se realizó en Matlab (Simulink) con la implementación de la transformada de Laplace de la expresión del control PID, para determinadas condiciones de vuelo. Posteriormente, se combinó el control PID y el control de lógica difusa en un mismo sistema (Ver figura 2). Según los resultados obtenidos en la simulación, se identificó que el sistema era críticamente amortiguado, teniendo un tiempo de 2s y una amplitud de 0.5m [6].

Figura 2: Modelo de control PI-Difuso para la guiñada a distintas condiciones [5].



En el año 2004, Lancaster [7] desarrolló un sistema de control de vuelo autopiloto aplicado para formaciones de UCAVs (Unmanned Combat Aerial Vehicle) y posteriormente se analizó mediante simulaciones no lineales en Matlab (Simulink). En el sistema de control de vuelo se utilizó el controlador PID. En la simulación y análisis de los modelos analizados en Matlab, se identificó deficiencias para algunas condiciones de vuelo, donde los resultados no eran muy confiables [7]. La variación de resultados de simulación y experimentales se explica debido a que en un entorno real surgen repentinamente factores que pueden afectar directamente en el desempeño del UAV, por ejemplo, los vientos cruzados no siempre son siempre tiene la misma velocidad y sentido.

Actualmente, existen muchas opciones para la implementación de un sistema de control para un UAV, pero las plataformas electrónicas Arduino son una buena opción dado que son dispositivos para prototipos de código abierto que permite la interacción entre computadoras y actuadores electrónicos, mediante la recepción de datos de entrada por sensores. Su funcionamiento se basa en la interacción de componentes de entrada y salida

que son de señal analógica y digital. El código de programación utilizado es el lenguaje “Processing” [8] y [9].

Las ventajas más importantes de la plataforma Arduino son su fácil utilización, programación intuitiva, bajo coste, compatibilidad con distintos sistemas operativos (Windows, Mac, Linux, etc.), lenguaje de programación extensible a “C++” y con conocimientos más profundos, el hardware puede ser extendido y mejorado [8] y [9].

Aprovechando las ventajas antes citadas, Nadales [9] desarrolló una investigación para el control de un cuadricóptero mediante una plataforma Arduino combinado a un sistema de comunicación de XBee. El objetivo del citado proyecto fue implementar un sistema de control inicial simple, dejando la posibilidad de extender el software y hardware a nuevas funcionalidades. En igual manera, Legasa [10] construyó un UAV basado en la plataforma Arduino y remotamente controlado desde un teléfono inteligente con sistema operativo Android. En el hardware se usó una plataforma Arduino UNO, una IMU y un módulo bluetooth (este último para comunicar el teléfono celular con el UAV). El software fue desarrollado mediante un controlador PID y un filtro de Kalman para suavizar la reacción de los actuadores a las medidas leídas por la IMU [10]. La aplicación de un filtro de Kalman en la programación definitivamente soluciona el problema de la reacción de los sensores frente a repentinas variaciones de ángulo de la IMU, pero es necesario tener en cuenta que el microprocesador del Arduino posee una limitada memoria de almacenamiento (32 kilobytes). Por otro lado, el control del UAV mediante un teléfono celular supone la creación de una interface de control (aplicación móvil) en el celular, además la comunicación “bluetooth” posee ciertas limitaciones de rango de operación y comunicación.

Finalmente, en el diseño del sistema se debe de considerar la configuración del UAV, siendo el caso de este proyecto un UAV de alas fijas cuya envergadura es de 1.8 m, con

un peso total de 3 kg, propulsado por un motor eléctrico sin escobillas tipo Brushless de 1100 kV (1.3 kgf de empuje) y controlado por radio frecuencia. Por otro lado, este trabajo de investigación solo se enfoca en el diseño de un sistema de estabilización y control del UAV, más no contempla el diseño del sistema de comunicación entre el UAV en vuelo y el operador en tierra. Pero se considera este tema como motivo de investigación para futuros proyectos donde la integración general de todos los sistemas en investigación determinara la viabilidad de la remplazar todo el sistema electrónico de un UAV por uno basado en la plataforma Arduino.

CAPÍTULO 2

MARCO TEÓRICO

2.1. Vehículo Aéreo No Tripulado (UAV)

El acrónimo UAV o conocido también en español como VANT hace referencia a aeronaves capaces de realizar vuelos sin la presencia de una tripulación a bordo. Para desarrollar este tipo de actividades aéreas, la aeronave está implementada con un piloto computarizado a bordo (autónomo) o un operador que la controlará remotamente desde una base en tierra (remotamente controlado).

Los elementos que permitirán que una aeronave sea autónoma o remotamente controlada, son una combinación de sensores, actuadores y procesadores; conectados en tiempo real a la base de operaciones en tierra, para controlar el vuelo y determinar el estado cinético del UAV [11]. Y al igual que el UAV, la base en tierra debe estar implementada para posibilitar la operación del UAV en condición de vuelo autónomo o remotamente controlado [12]. Siendo la plataforma Arduino una elección muy ventajosa desde la visión de nuestro proyecto, ya que este producto puede realizar las mismas funciones que los sistemas de control y comunicación ya integrados en los UAV. Además, el costo de este producto y la facilidad de uso, hace viable la implementación de un sistema de control y estabilización por cualquier persona (sin experiencia en programación y con limitaciones económicas).

Así mismo, debido al significado literal del acrónimo UAV, se infiere que todos los objetos capaces de realizar vuelos sin tripulación son considerados UAV, pero existen ciertos artefactos como globos aerostáticos meteorológicos y misiles autónomos o pilotados a

control remoto que no se consideran en este grupo. Una característica para definir un UAV es el control que se tiene sobre ellos [12] así como también los múltiples usos que se le da en distintas misiones y escenarios de vuelo [13] y [14]. Siendo, los globos aerostáticos y misiles bélicos dispositivos sin control del mismo o de un solo uso [14].

Por otro lado, actualmente existen confusiones a la hora de nombrar a los vehículos aéreos no tripulados, debido al tipo de operación de vuelo de estos. Al primer grupo se le denomina como SAA (sistema aéreo autónomo), refiriéndose a aeronaves autónomas que no dependen de la interacción con algún piloto o controlador en tierra, pero que actualmente no existen sistemas completamente autónomos [14], debido a que los actuales drones categorizados como SAA aún requieren de intervención humana en ciertos segmentos de operación considerados como críticos, donde es necesario tomar decisiones. Mientras que el grupo de drones catalogados como UAV o VANT son remotamente controlados por un operador en tierra.

Otro acrónimo utilizado para denominar a los drones es UAS (Unmanned Aircraft System), que es utilizado por las autoridades aeronáuticas internacionales FAA (Federal Aviation Administration) y EASA (European Aviation Safety Agency). Este acrónimo fue adoptado por la FAA y EASA para denominar de manera técnica a los drones en las regulaciones que limitan y controlan su utilización. Los requerimientos para designar a los UAV's como UAS son: garantizar la aeronavegabilidad inmanente a este y desplazarse a través de tres ejes de coordenadas [13].

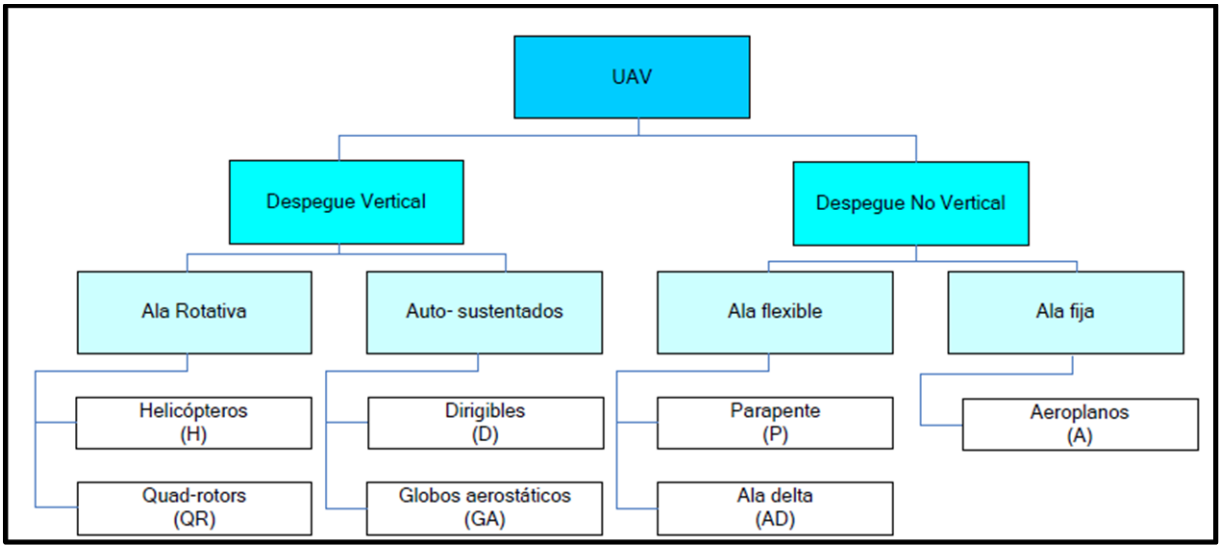
2.2. Tipos de Vehículos Aéreos No Tripulados

Los UAV pueden ser clasificados de acuerdo a las características y configuraciones que posean [15]. La primera forma de clasificarlos es según el tipo de operación: UAV (remotamente controlados) y SAA (aeronave autónoma) [15]. Otra forma de clasificarlos es en dos grupos; en el primer grupo corresponden los avionico-aerodinámico, que son

identificados por sus características de diseño como alcance, carga útil, tipo de propulsión y entre otros, mientras en el segundo grupo se basan en la misión o función que también es una característica de diseño debido a un requerimiento fundamental para la concepción del UAV [14].

Asimismo, otra manera de clasificarlos se presenta en la figura 3, donde el parámetro de diferencia es el tipo de despegue del UAV. Este parámetro caracteriza el tipo de despegue depende principalmente de la geometría del UAV y del tipo de alas que este posea (ala fija o ala giratoria).

Figura 3: Clasificación de los UAV según el tipo de despegue [11].



2.2.1. Despegue Vertical.

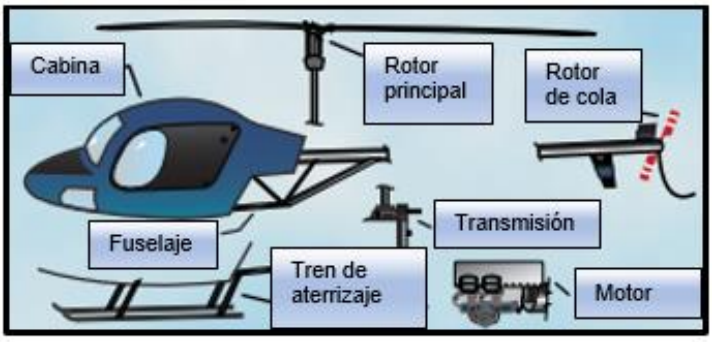
A la mayoría de UAV que realizan este tipo de despegue se les denomina multicópteros (multirotores) si poseen más de tres hélices o helicópteros, siendo los subtipos más comunes los cuadricópteros y hexacópteros. A este tipo de UAV se le denomina también de ala giratoria, ya que la fuerza de sustentación es generada por las hélices, que son accionadas directamente por motores o mediante ejes de transmisión.

Producto de la relación aerodinámica entre las alas giratorias (hélices), que se encuentran girando a una determinada velocidad angular y el aire incidiendo en ellas, se genera la fuerza de sustentación que permite que el UAV se desplace en el aire mediante la variación de velocidad de los motores, o se mantenga en una posición estática (Hover) si todos los motores giran a una misma velocidad [16]. Una ventaja de este tipo de UAV es la estabilidad en vuelo [16], debido a que los sistemas de control y comunicación de los multicópteros son implementados con tecnología electrónica como autopilotos, sensores y otros.

A) Helicópteros.

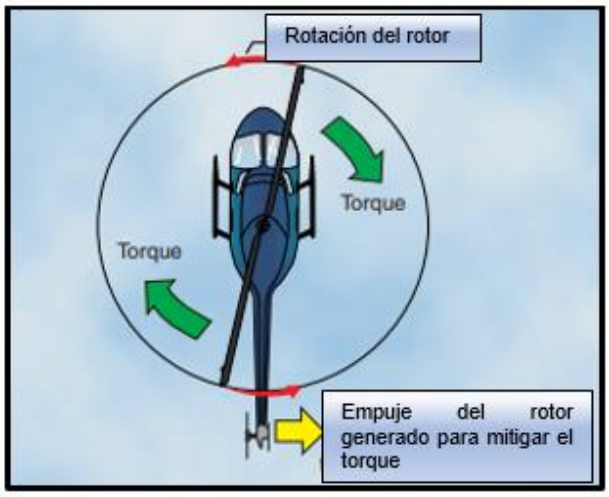
Al igual que los helicópteros convencionales, este tipo de UAV poseen una estructura; que permite el alojamiento o sujeción de los sistemas de aviónica (sistema de control y comunicación que incluyen microprocesadores como autopilotos, sensores, acelerómetros, giroscopios, etc.), motores, transmisiones y tren de aterrizaje [17] (ver figura 4). Pero la diferencia con las aeronaves convencionales, es que los helicópteros de pequeña escala (que es la categoría de nuestro proyecto) son radio controlados o autónomos, y no disponen de un compartimiento de carga en el interior del fuselaje, ya que generalmente la carga se transporta fuera de este. Tampoco requieren de una cabina para la tripulación, ya que el control del UAV puede ser ejecutado autónomamente o por radio control, y que en ambos casos se utiliza un sistema de comunicación y control.

Figura 4: Partes de un helicóptero [16].



Por otro lado, a diferencia de los cuadricópteros y hexacópteros, un problema inherente al funcionamiento de este tipo de UAV, que afecta a la estabilidad de estas aeronaves, es el efecto torque producido por el giro del rotor principal al ser accionado por el/los motor(es), siendo necesario la utilización de un sistema anti torque que mitigue este efecto [17].Para dicho problema, la solución más común es la utilización de un rotor de cola que varía su ángulo de paso y su velocidad, para generar suficiente fuerza que elimine el torque del rotor principal (ver figura 5).

Figura 5: Efecto torque [16].



Otra solución en aeronaves convencionales y la más utilizada en UAV es la disposición de dos rotores coaxiales principales, donde el primer rotor gira de forma opuesta al segundo rotor, el efecto torque producido por cada rotor es eliminado [17].

B) Cuadricóptero

El cuadricóptero es un UAV que posee una estructura de cuatro brazos dispuesta en forma de “+” o de “x” donde en los extremos de cada brazo se sitúan cuatro hélices accionadas directamente por sus respectivos motores y en la parte central se ubica el circuito electrónico de control y la carga útil [16] (ver la figura 6).

Debido a la distribución de hélices y peso del mismo, este UAV es más estable, ya que cada pareja de motores tiene un sentido de giro opuesto al otro. Además, posee un mayor tiempo de vuelo ya que tiene la capacidad de llevar más baterías y cada motor trabaja a un régimen de giro relativamente menor al de un helicóptero [16].

Figura 6: Vista general de un Cuadricóptero [15].

C) Hexacóptero

Los hexacópteros tienen el mismo principio de distribución simétrica que los cuadricópteros, pero la diferencia entre ambas configuraciones, es que poseen seis brazos donde cada uno de estos sujeta a los motores y hélices (ver la figura 7). La parte central de la estructura es el soporte del sistema electrónico y carga útil. Al igual que los cuadricópteros el giro opuesto de cada trio de motores genera un equilibrio de torque generado por cada hélice. Al ser una versión mejorada de los cuadricópteros, el hexacóptero posee una mayor capacidad de carga, agilidad y estabilidad; siendo el adecuado para grabación de videos en puntos estáticos y también como aeronave de competición [16].

Figura 7: Vista de un hexacóptero [15].

2.2.2. Despegue Horizontal.

Los UAV considerados en esta clase son las aeronaves de ala fija. Según la cantidad de alas, configuración de las alas y ubicación de las alas se obtendrá diferentes tipos de cualidades y características estructurales, aerodinámicas y de dinámica de vuelo del UAV. Las aeronaves de esta categoría se clasifican por la cantidad de alas, la posición de estas en el fuselaje y la geometría que las alas tengan, que contribuyen directamente sobre la estabilidad lateral, performance y maniobrabilidad de la aeronave [18].

Además, debido a las cargas aerodinámicas generadas en las alas y fuselaje durante el vuelo, se generan momentos en la aeronave que desestabilizan al UAV, donde el empenaje (estabilizador horizontal y vertical) equilibra estos momentos.

Otro tipo de UAV de ala fija son las alas volantes, donde las alas forman un solo cuerpo con el fuselaje y sin empenaje. La ventaja de este tipo de configuración es que es más eficiente aerodinámicamente, a pesar de que son inestables [19] y requieren de complicados sistemas computarizados de control.

2.3. Mecánica de vuelo de los UAV

La mecánica de vuelo es un área de estudios que hace uso de sistemas de referencias cartesianos ortogonales para proyectar las velocidades, aceleraciones, momentos, fuerzas posiciones y trayectorias inherentes a la dinámica de las aeronaves [20], [21] y [22]. Y dado que los UAV y aeronaves en general poseen 6 grados de libertad, el estudio del movimiento de estos requiere de un alto grado de complejidad [20] y [21], donde tres grados pertenecen al movimiento de traslación del centro de gravedad del avión y respecto al sistema de referencia de ejes tierra, y otros tres grados pertenecen al movimiento de rotación con respecto a las coordenadas de ejes al cuerpo del avión [22].

2.3.1. Sistema de coordenadas de los UAV

Los sistemas de referencia cartesianos más utilizados en aviación comprenden los siguientes [22]:

- Sistema de referencia inercial $F_I(O_I, x_I, y_I, z_I)$
- Sistema de referencia geocéntrico giratorio $F_g(O_g, x_g, y_g, z_g)$
- Sistema de ejes tierra $F_E(O_E, x_E, y_E, z_E)$
- Sistema de ejes horizonte local $F_h(O_h, x_h, y_h, z_h)$
- Sistema de ejes de cuerpo $F_b(O_b, x_b, y_b, z_b)$
- Sistema de ejes de viento $F_w(O_w, x_w, y_w, z_w)$

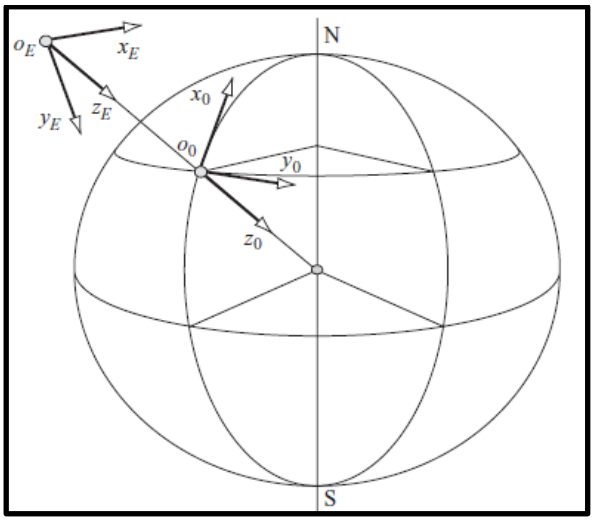
Donde "O" representa el centro del origen de coordenadas y "x", "y" y "z" son los ejes ortogonales de cada sistema.

No obstante, a continuación de describirá únicamente, el sistema de coordenadas de ejes en tierra, el sistema de coordenadas de ejes de cuerpo y las variaciones que se dan en estos.

A) Sistema de ejes tierra $\boldsymbol{F_E(O_E, x_E, y_E, z_E)}$

Considerado también un sistema navegacional, este sistema permite ubicar al UAV en el espacio, con respecto a su posición inicial (punto de despegue) y su trayectoria [22]. Y dado que este sistema supone a la superficie de la tierra como una plancha plana (para pequeñas traslaciones en el planeta), el plano (O_E, x_E, y_E) siempre es paralelo al plano (O_0, x_0, y_0) [20] (ver figura 8). Los ejes "Oz_E" y "Oz_0" coinciden con la fuerza de gravedad ejercida por la masa terrestre, mientras que el eje "O_0x_0" apunta al norte, el eje "O_0y_0" apunta al este [20]. Adicionalmente, a menudo, las coordenadas $F_E(O_E, x_E, y_E, z_E)$ coinciden con el sistema de coordenadas de cuerpo del UAV (cuando este describe un vuelo estable a altura constante) [20].

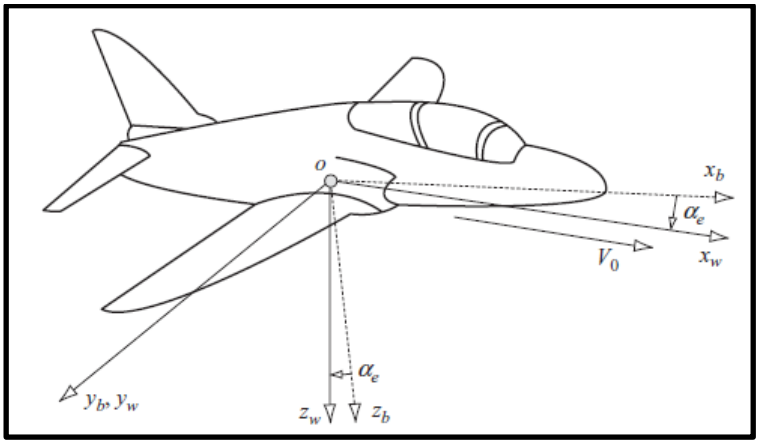
Figura 8: Sistema de coordenadas terrestres [19].



B) Sistema de coordenadas de cuerpo $F_b(O_b, x_b, y_b, z_b)$

Otro sistema de coordenadas es el de ejes fijados al cuerpo del UAV y cuyos ejes $F(OX_bY_bZ_b)$ se desplazan con el avión, donde el origen de coordenadas normalmente se sitúa en el centro de gravedad del avión [22] y [21](ver figura 9). El plano (O_b, x_b, z_b) corresponde al plano de simetría, el eje "Ox_b" es paralelo al fuselaje, el eje "Oy_b" es paralelo a la envergadura del ala y el eje "Oz_b" apunta hacia abajo (para un vuelo horizontalmente estable) [20].

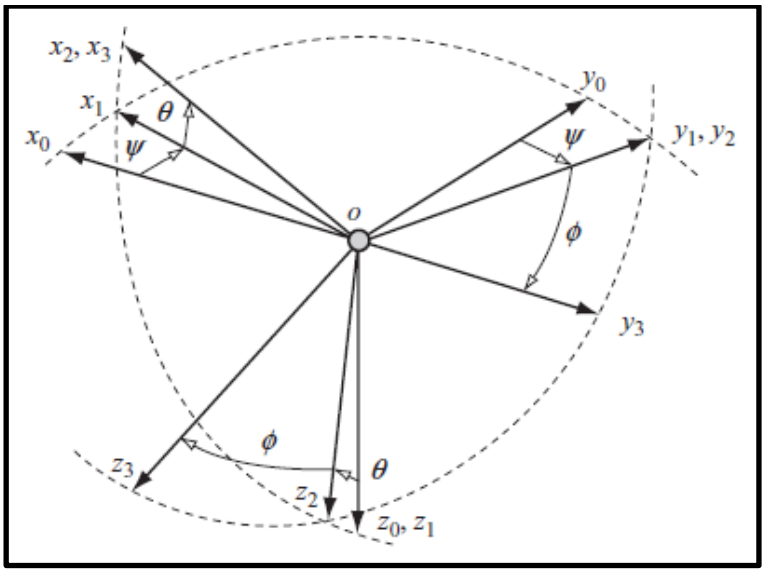
Figura 9: Coordenadas con ejes fijos al cuerpo del avión [19]



C) Ángulos de Euler

Para orientar dos sistemas de referencia $F(O_0X_0Y_0Z_0)$ y $F(O_3X_3Y_3Z_3)$, donde los subíndices “0” y “3” representan sistema de coordenada en una actitud inicial y sistema de coordenada en una actitud final respectivamente (Ver figura 10). Donde $O_0 = O_3 = O_2 = O_1$, lo cual significa que el origen no varía. Asimismo, al rotar el eje Oz_3 un ángulo ψ se generará un nuevo sentido para Ox_1 y Oy_1, que es denominado como guiñada “Yaw”. Al rotar por segunda vez un ángulo θ a lo largo del eje Oy_1 se genera un nuevo eje Ox_2 y Ox_2, conocido como cabeceo “Pitch”. Finalmente, al rotar un ángulo ϕ a través del eje Ox_2 se generan un eje Oy_3 y Oz_3 que se designa como alabeo “Roll” (ver la figura 10). La secuencia de cada movimiento de rotación y orientación angular establecerá el estado de la aeronave [21] y [22]. Es decir, se podrá conocer la posición y trayectoria de las aeronaves con respecto a otro sistema de coordenadas fijos [23].

Figura 10: Ángulos de Euler [19].



Conocidos las rotaciones descritas anteriormente, la manera matemática de representar cada rotación (guiñada, cabeceo y alabeo) se presentan a continuación [20].

- Guiñada "ψ"

$$\begin{bmatrix} Ox_1 \\ Oy_1 \\ Oz_1 \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Ox_0 \\ Oy_0 \\ Oz_0 \end{bmatrix} \quad (1)$$

- Cabeceo "θ"

$$\begin{bmatrix} Ox_2 \\ Oy_2 \\ Oz_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} Ox_1 \\ Oy_1 \\ Oz_1 \end{bmatrix} \quad (2)$$

- Alabeo "ϕ"

$$\begin{bmatrix} Ox_3 \\ Oy_3 \\ Oz_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} Ox_2 \\ Oy_2 \\ Oz_2 \end{bmatrix} \quad (3)$$

Ulteriormente, al multiplicar las matrices anteriores se obtiene la matriz que contiene el nuevo sistema de coordenadas [20].

$$\begin{bmatrix} Ox_3 \\ Oy_3 \\ Oz_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Ox_0 \\ Oy_0 \\ Oz_0 \end{bmatrix} \quad (4)$$

Otra manera de representar esta matriz es [20].

$$\begin{bmatrix} Ox_3 \\ Oy_3 \\ Oz_3 \end{bmatrix} = D \begin{bmatrix} Ox_0 \\ Oy_0 \\ Oz_0 \end{bmatrix} \quad (5)$$

Donde "D" esta dado por la siguiente matriz [20].

$$D = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \quad (6)$$

La matriz "D" es el producto de la multiplicación de las tres matrices anteriores.

2.3.2. Dinámica de vuelo de un UAV de alas fijas.

En los UAV de alas fijas o despegue horizontal, el modelo dinámico del UAV se fundamenta en el teorema de la cantidad de movimiento y el teorema del momento cinético [22], cuyas expresiones son presentadas respectivamente [22].

$$\vec{F} = \frac{d(m\vec{V})}{d(t)} \quad (7)$$

$$\vec{G} = \frac{d(\vec{h})}{d(t)} \tag{8}$$

$$\vec{h} = I\vec{\omega} \tag{9}$$

Donde:

$\vec{F}$: Fuerza resultante (N)

m : Masa del avión (kg)

$\vec{V}$: Velocidad absoluta (m/s)

t : Tiempo (s)

$\vec{G}$: Momento resultante (Nm)

$\vec{h}$: Momento cinético del avión (kgm^2/s)

I : Momento de inercia (kgm^2)

$\vec{\omega}$: Velocidad angular (rad/s)

Y al descomponer en los tres ejes la fuerza, momento, velocidad y velocidad angular se obtiene [22]:

$$\vec{F} = \left(F_x, F_y, F_z\right)^T;\ \vec{G} = (L, M, N)^T;\ \vec{V} = (u, v, \omega)^T;\ \vec{\omega} = (p, q, r)^T \tag{10}$$

Donde

F_x : componente del vector fuerza en el eje OX (N)

F_y : componente del vector fuerza en el eje OY (N)

F_z : componente del vector fuerza en el eje OZ (N)

L : componente del vector de momento en el eje OX (Nm)

M : componente del vector de momento en el eje OY (Nm)

N : componente del vector de momento en el eje OZ (Nm)

u : componente del vector de la velocidad en el eje OX (m/s)

v : componente del vector de la velocidad en el eje OY (m/s)

ω : componente del vector de la velocidad en el eje OZ (m/s)

p : componente del vector velocidad angular en el eje OX (rad/s)

q : componente del vector velocidad angular en el eje OY (rad/s)

r : componente del vector velocidad angular en el eje OZ (rad/s)

Entonces, remplazando los términos anteriores en las ecuaciones (7) y (8) [22].

$$\vec{F} = m\left(\frac{\partial \vec{V}}{\partial t} + \vec{\omega} \wedge \vec{V}\right);\ \vec{G} = \frac{\partial \vec{h}}{\partial t} + \vec{\omega} \wedge \vec{h} \tag{11}$$

Y operando las expresiones anteriores, se obtiene las ecuaciones de Euler del movimiento del avión [22]:

$$F_x = m(\dot{u} - rv + q\omega) \tag{12a}$$

$$F_y = m(\dot{v} + ru - p\omega) \tag{12b}$$

$$F_z = m(\dot{\omega} - qu + pv) \tag{12c}$$

$$L = I_x\dot{p} - J_{xz}\dot{r} + \left(I_z - I_y\right)qr - J_{xz}pq \tag{12d}$$

$$M = I_y\dot{q} - (I_z - I_x)pr + J_{xz}(p^2 - r^2) \tag{12e}$$

$$N = I_z\dot{r} - J_{xz}\dot{p} - \left(I_x - I_y\right)pq + J_{xz}qr \tag{12f}$$

Asimismo, es necesario considerar parámetros como el empuje (propulsión), fuerzas aerodinámicas y gravitatorias (T, A, G respectivamente) [22].

$$\vec{F} = \vec{F}_T + \vec{F}_A + \vec{F}_G \tag{13}$$

$$\vec{G} = \vec{G}_T + \vec{G}_A \tag{14}$$

$$\left(\vec{F}_G\right)_h = \begin{Bmatrix} 0 \\ 0 \\ mg \end{Bmatrix} \tag{15}$$

Donde

m : masa (kg)

g : gravedad (m/s^2)

Y considerando estos parámetros en las ecuaciones (12) [22].

$$-mg\sin\theta + F_{Tx} + F_{Ax} = m(\dot{u} - rv + q\omega) \tag{11a}$$

$$mg\cos\theta\sin\phi + F_{Ty} + F_{Ay} = m(\dot{v} + ru - p\omega) \quad (11b)$$

$$mg\cos\theta\cos\phi + F_{Tz} + F_{Az} = m(\dot{\omega} - qu + pv) \quad (11c)$$

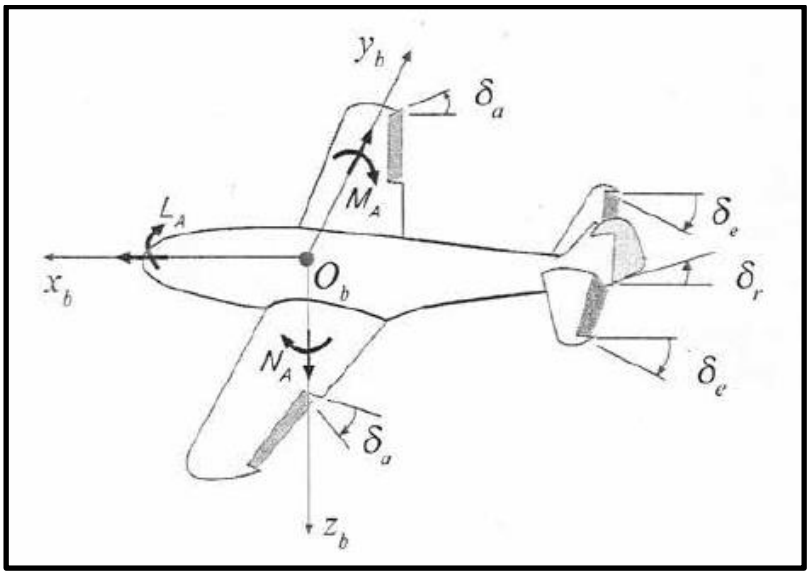
$$L_T + L_A = I_x\dot{p} - J_{xz}\dot{r} + \left(I_z - I_y\right)qr - J_{xz}pq \quad (11d)$$

$$M_T + M_A = I_y\dot{q} - (I_z - I_x)pr + J_{xz}(p^2 - r^2) \quad (11e)$$

$$N_T + N_A = I_z\dot{r} - J_{xz}\dot{p} - \left(I_x - I_y\right)pq + J_{xz}qr \quad (11f)$$

Sobre la base de las ecuaciones anteriores, el control de los tres momentos aerodinámicos de balanceo, cabeceo y guiñada (L_A, M_A y N_A), es mediante tres superficies de mando conocidas como alerones, timón de profundidad y timón de dirección [22] (ver figura 11). Donde "δ_a" es el ángulo de deflexión de los alerones, "δ_e" representa el ángulo de deflexión del timón de profundidad y "δ_r" es el ángulo de deflexión del timón de dirección [22]. Asimismo, las variaciones de los momentos afectan a las componentes de las fuerzas aerodinámicas $\left(F_{Ax}, F_{Ay}, F_{Az}\right)$ [22].

Figura 11: Superficies de control y ángulos de deflexión [21].



Finalmente, el propósito de este proyecto de investigación es mantener estables las fuerzas y momentos aerodinámicos, cuando el UAV realice vuelos horizontales rectilíneos y bajo efectos de perturbaciones (vientos cruzados y ráfagas de viento). Por lo tanto, las ecuaciones de momentos y fuerzas deberán satisfacer las ecuaciones siguientes [22].

$$-mg\sin\theta + F_{Tx} + F_{Ax} = 0 \quad (16a)$$

$$mg\cos\theta\sin\phi + F_{Ty} + F_{Ay} = 0 \quad (16b)$$

$$mg\cos\theta\cos\phi + F_{Tz} + F_{Az} = 0 \quad (16c)$$

$$L_T + L_A = 0 \quad (16d)$$

$$M_T + M_A = 0 \quad (16e)$$

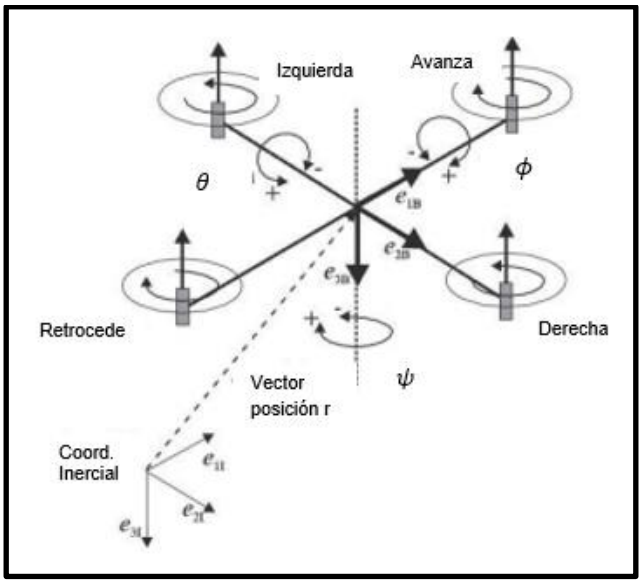
$$N_T + N_A = 0 \quad (16f)$$

2.3.3. Dinámica de vuelo de un UAV de despegue vertical

El vuelo de los UAV del tipo multirotores se basa principalmente en la velocidad angular (RPM) con que giran los motores, que afecta directamente en el empuje que genera cada hélice accionada por su respectivo motor [24] , [25] y [26]. El control de la estabilidad, maniobrabilidad, torque, guiñada, balanceo y cabeceo dependen del empuje generado por las alas giratorias, y es necesario tener un preciso ángulo de maniobra para realizar las operaciones de vuelo (despegue, vuelo estático o “hover”, avance/retroceso y giros) y efectuar trayectorias un tanto complejas dadas por la combinación de operaciones de desplazamiento y giros [24], [25] y [26]. Además, a diferencia de los helicópteros convencionales, los multirotores se caracterizan por poseer buena estabilidad y maniobra [26].

Los sistemas de referencia a considerar en la dinámica de vuelo de un cuadricóptero son el sistema de coordenadas inercial, que está situado en cualquier punto fijo en la superficie de la tierra [27] y el sistema de coordenadas ejes cuerpo del UAV [6] (ver figura12). Y dado que los movimientos de traslación y rotación en los seis grados de libertad de los UAV dependen de los de los ángulos de Euler (guiñada, cabeceo y alabeo), cuyo vector es [28].

Figura 12: Coordenadas inerciales y ejes al cuerpo de un cuadricóptero [27].



$$\Omega^T = (\phi, \theta, \psi) \tag{17}$$

Donde el ángulo de alabeo "ϕ" está comprendido entre $[-\pi; \pi]$, el ángulo de cabeceo "θ" esta entre $\left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$ y el ángulo de guiñada "ψ" esta entre $[-\pi; \pi]$ [6].

Y el vector posición r es [28].

$$r^T = (X, Y, Z) \tag{18}$$

Ulteriormente, la matriz de rotación es [28]:

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{19}$$

Asimismo, la fuerza de empuje generado por cada rotor está dada por [24] y [28].

$$\left|\vec{T_i}\right| = \rho A \omega_i^2 = b\omega_i^2 \tag{20}$$

Donde

$$b = \rho A \tag{21}$$

i : número de motor

ρ : densidad del aire (kg/m^3)

A : área transversal de la hélice (m^2)

ω_i : velocidad de giro del motor (rad/s)

Y el empuje total "$\vec{T}_{total}$" generado por todos los motores es:

$$\vec{T}_{total} = \vec{T}_1 + \vec{T}_2 + \vec{T}_3 + \vec{T}_4 + \cdots \vec{T}_n \tag{22}$$

Donde:

$\vec{T}_1$: Empuje del motor 1 (N)

$\vec{T}_2$: Empuje del motor 2 (N)

$\vec{T}_3$: Empuje del motor 3 (N)

$\vec{T}_4$: Empuje del motor 4 (N)

$\vec{T}_n$: Empuje del motor n (N)

Seguidamente, la ecuación de las aceleraciones del UAV son [28].

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = g \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - R \frac{\vec{T}_{total}}{m} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{23}$$

De igual manera, para el producto de la matriz inercial, primera derivada de los ángulos de Euler y momentos se tiene [28].

$$I\ddot{\Omega} = -\left(\dot{\Omega} \mathrm{X} I\dot{\Omega}\right) - M_G + M \tag{24}$$

Donde

I : matriz inercial

M_G : momento giroscópico

$$M = \begin{pmatrix} lb(\omega_3^2 - \omega_4^2) \\ lb(\omega_1^2 - \omega_2^2) \\ d(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \end{pmatrix} \tag{25}$$

Donde

l : distancia de la hélice al cg (m)

Posteriormente, se generan las siguientes ecuaciones [28].

$$u_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{26}$$

$$u_2 = b(\omega_3^2 - \omega_4^2) \tag{27}$$

$$u_3 = b(\omega_1^2 - \omega_2^2) \tag{28}$$

$$u_4 = d(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \tag{29}$$

Donde

u_1 : empuje en la estructura del UAV (Nm)

u_2 : torque de alabeo (Nm)

u_3 : torque de cabeceo (Nm)

u_4 : torque de guiñada (Nm)

Por otro lado, dado que los torques giroscópicos dependen de la velocidad de rotación, se tiene [28].

$$u^T = (u_1, u_2, u_3, u_4) \tag{30}$$

$$g(u) = \omega_1 + \omega_2 - \omega_3 - \omega_4 \tag{31}$$

Finalmente, al reemplazar las ecuaciones (25), (30) y (31) en la ecuación (24) se obtiene [28].

$$\ddot{x} = -(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\frac{u_1}{m} \tag{32}$$

$$\ddot{y} = -(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\frac{u_1}{m} \tag{33}$$

$$\ddot{z} = g - (\cos\phi \cos\theta)\frac{u_1}{m} \tag{34}$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{I_R}{I_x}\dot{\theta}g(u) + \frac{l}{I_x}u_2 \tag{35}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{I_R}{I_y}\dot{\phi}g(u) + \frac{l}{I_y}u_3 \tag{36}$$

$$\ddot{\psi} = \dot{\theta}\dot{\phi}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}u_4 \tag{37}$$

De acuerdo a las ecuaciones de la segunda derivada de los angulos de Euler, estos son no lineales, por lo que se procede a linealizarlos mediante el teorema de Taylor [6].

$$\dot{\phi} = 0 \tag{38}$$

$$\dot{\theta} = 0 \tag{39}$$

$$\dot{\psi} = 0 \tag{40}$$

Y reemplazando en la expresiones (35), (36) y (37) se obtiene.

$$\ddot{\phi} = \frac{l}{I_x} u_2 \tag{41}$$

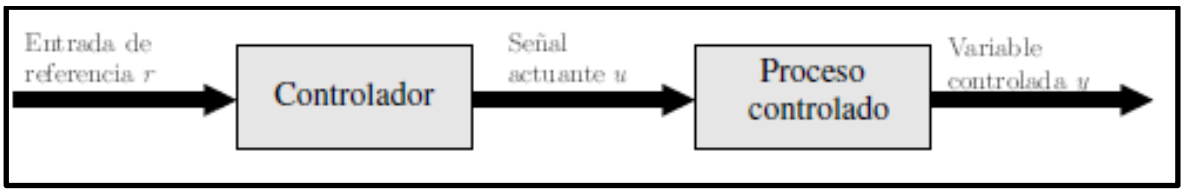
$$\ddot{\theta} = \frac{l}{I_y} u_3 \tag{42}$$

$$\ddot{\psi} = \frac{1}{I_z} u_4 \tag{43}$$

2.4. Sistemas de control

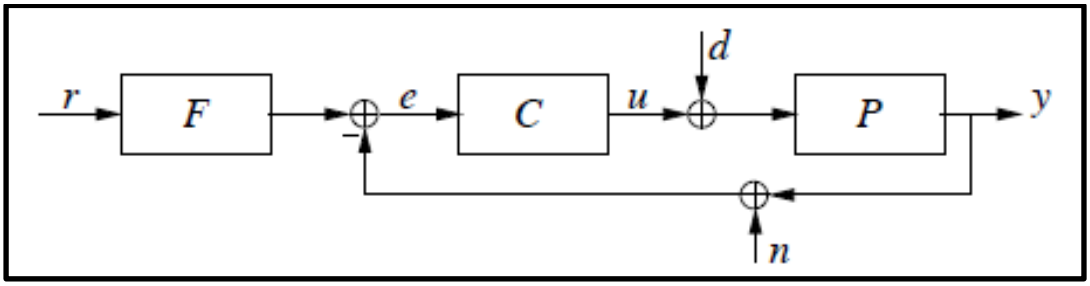
La función principal de los sistemas de control es obtener un determinado comportamiento o repuesta en los procesos de un sistema [29]. Esto se logra mediante la implementación de un sistema de control de lazo abierto, donde una variable de entrada pasa a través de un sistema y este emite una variable de salida o respuesta (ver figura 13) [4].

Figura 13: Diagrama de un sistema de control de lazo abierto [3].



Otro sistema de control y más ampliamente utilizado es el sistema de lazo cerrado, cuya exactitud respecto al de lazo abierto es mayor, debido a la realimentación propia del sistema (ver figura 14) [4].

Figura 14: Diagrama de un sistema de control de lazo cerrado [28].



Donde, el sistema está compuesto por un filtro de realimentación "F", el controlador "C" y el proceso "P". Asimismo, "r" es la variable de entrada, "y" es la variable de salida, "u" es la variable de control, "d" es la señal de perturbación externa, "n" es la señal de ruido medida y "e" es el control del error [29], y que este último está dado por la siguiente expresión [29].

$$e = r - y \tag{44}$$

Un sistema de control elemental de tipo lazo abierto es el controlador On-Off, donde las variables solo pueden tomar dos valores (u_{max} y u_{min}). Y dada esta condición, la desventaja son las oscilaciones constantes en el proceso de las variables [29].

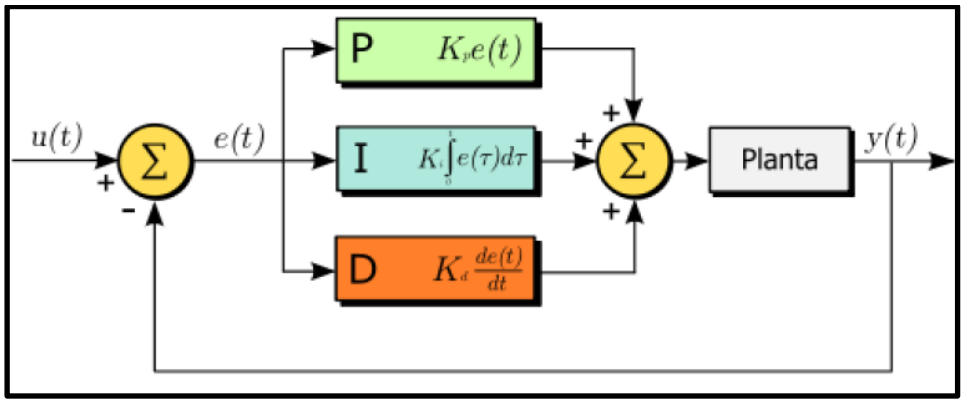
Otra alternativa para los sistemas de control son los modelos de lógica borrosa o difusa, cuyo principio de funcionamiento se enfoca en la toma de variables aleatorias, pero relacionadas entre sí [23]. Por ejemplo, si se deseará que un UAV recorra una distancia "X" en un tiempo "T" determinado, pero este vuela a una velocidad constante "V" que retrasará el tiempo estimado del UAV; la computadora a bordo ordenara que se incremente la velocidad y se consiga el objetivo. En este caso las dos variables son el tiempo "T" estimado y el tiempo "T1" calculado para la velocidad "V".

Sin embargo, el sistema con mayor aplicación en diferentes áreas de la industria, incluida la aviación, es el controlador PID (Proporcional, Integral y Derivativo); considerados como la mejor opción para controlar vehículos que se desplazan en tres dimensiones [23]. A continuación, se describirá las principales características del controlador PID, debido a que este algoritmo es usualmente utilizado para el control de la estabilidad de los sistemas monitoreados por plataformas y microcontroladores electrónicos.

2.4.1. Controlador PID

El controlador PID, es un sistema de tipo lazo cerrado (basado en realimentación de parámetros) compuesto de tres tipos de control; el parámetro proporcional determina la reacción de un error, el integral genera una corrección al reducir el error y el derivativo determina la reacción del tiempo en que el error se produce [23] y [29] (ver la figura 15).

Figura 15: Ecuaciones e interacción de parámetros del control PID [22].



Según la figura 15, el valor medido es la sumatoria de los parámetros: proporcional, integral y derivativo; donde las constantes K_P, K_I y K_D se ajustarán adecuadamente para aproximarse al valor deseado, como se observa en la siguiente expresión [29].

$$u(t) = MV(t) = K_p e(T) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} c(t) \qquad (45)$$

Donde

K_p: Ganancia proporcional (adimensional).

K_i: Ganancia integral (adimensional).

K_d: Ganancia derivativa (adimensional).

e: Error

t: Instante “t” (s).

A) Acción proporcional

La acción proporcional es el control del error actual del proceso, y a diferencia del controlador On-Off, este sistema proporciona pequeñas variables de control,

debidos a pequeñas variaciones del error. No obstante, la principal desventaja es que genera un estado constante de error. La expresión matemática de este parámetro es [29].

$$u(t) = K_p e(t) = K_p\big(r(t) - y(t)\big) \tag{46}$$

Donde "K_p" es la constante de ganancia proporcional, y este valor es multiplicado por el error actual (ecuación 44).

Y la derivada de la función de transferencia es [29].

$$C(s) = K_p \tag{47}$$

Y si aparece una perturbación, debía a una integración dinámica en el proceso, se adiciona a la expresión el término "u_b" [29].

$$u(t) = K_p e(t) + u_b \tag{48}$$

B) Acción integral

La acción integral es el producto de la constante de ganancia integral "K_i" por la integral del error (ver la expresión 49). Además, la integral está relacionada a los valores registrados del error, ya que representa la acumulación de este a través del tiempo [29].

$$u(t) = K_i \int_0^t e(\tau) d\tau \tag{49}$$

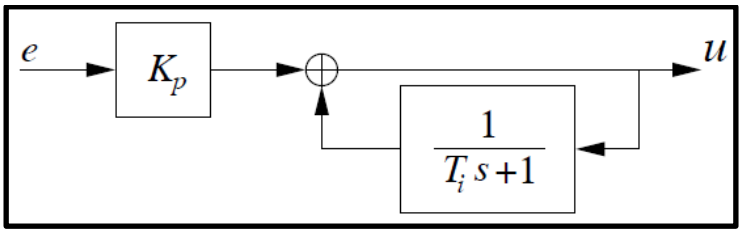
Donde la función de transferencia es [29].

$$C(s) = \frac{K_i}{s} \tag{50}$$

Asimismo, debido a que la acción integral es capaz de corregir automáticamente el valor de la variable de perturbación "u_b", siendo nulo el error de estado constante (ver figura 16). Donde la función de transferencia resultante es [29].

$$C(s) = K_p\left(1 + \frac{1}{T_i s}\right) \tag{51}$$

Figura 16: Corrección automática de la perturbación en el control PI [28].



Finalmente, la combinación de la acción proporcional e integral soluciona los problemas de respuestas oscilatorias, propias del control On-Off, y el error de estado constante, característico de la acción proporcional [29].

C) Acción derivativa

Esta acción permite mejorar el desempeño del control de un sistema, dado que está basada en la predicción de los valores del error, evitando desviaciones [29]. Donde la expresión correspondiente es [29].

$$u(t) = K_d \frac{de(t)}{dt} \tag{52}$$

Donde "K_d" es la ganancia derivativa. Y su respectiva función de transferencia es [29].

$$C(s) = K_d s \tag{53}$$

Sin embrago, la principal desventaja de esta acción es su extremada sensibilidad al ruido, que conlleva a limitar su uso [4] y [29].

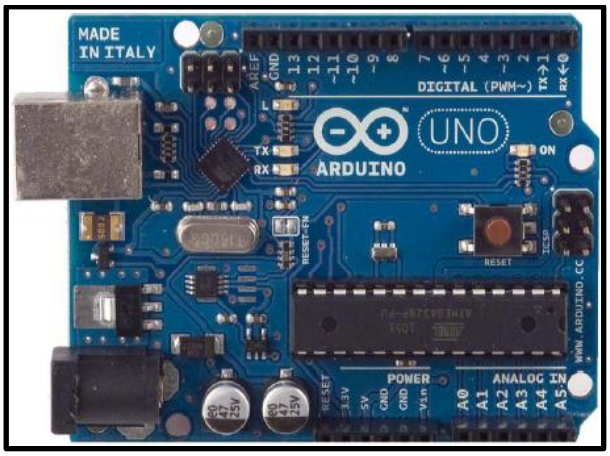
2.4.2. Componentes del sistema de control del UAV

El sistema de control de un UAV, generalmente incluye plataformas electrónicas (Arduino), sensores, acelerómetros, giroscopios, actuadores y una fuente de alimentación.

A) Plataforma electrónica Arduino

Arduino es una plataforma electrónica computarizada de código abierto [8] y [30], hardware y software libre [8] compuesto por un microcontrolador simple [31] que permite diseñar artefactos interactivos (ver la figura 17). Ya que Arduino está creado para leer datos de interruptores, sensores, acelerómetros y giroscopios; y controlar luces, motores servomotores y otros actuadores [8] y [31].

Figura 17: Hardware de la plataforma Arduino [30].



Por otro lado, la plataforma Arduino puede controlar proyectos de forma autónoma [31], mediante la creación de un codigo de control en la computadora que se ejecutara una vez grabado en el microcontrolador [30]. También se puede ejecutar ese programa en la computadora, y esta se conectará a Arduino mediante un sistema inalámbrico.

La creación del programa que Arduino ejecuta se puede desarrollar mediante el código de programación basado en “Wiring” más conocido como “Arduino Programming language” o el “Arduino Development Enviroment” basado en “Processing” [8]. Adicionalmente, para usuarios más experimentados en lenguaje de programación, es posible utilizar el lenguaje C++ y/o insertar directamente en los programas el lenguaje AVR C [8].

La plataforma Arduino se caracteriza por su bajo coste a comparación de otras plataformas, siendo la versión ensamblada la más costosa [8]. Además, debido a la facilidad de uso, esta es la más adecuada para principiantes, pero a la vez flexible para que pueda ser utilizado por usuarios experimentados [8].

Plataforma Arduino UNO

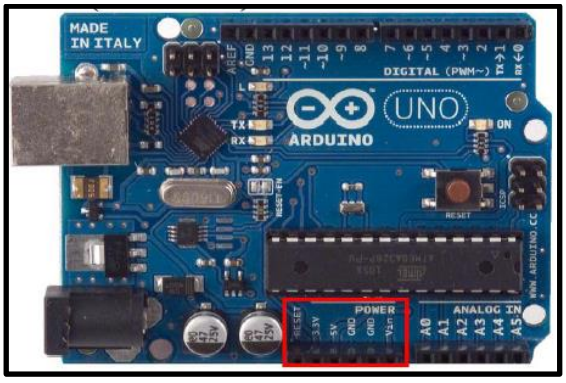
Las especiaciones técnicas son:

Tabla 1: Especificaciones técnicas de una plataforma Arduino [30].

Microcontrolador	AT Mega328
Voltaje de operación	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje limite	6-20V
Pines digitales I/O	14 (6 con PWM)
Pines de entrada analogos	6
Corriente DC para pin I/O	40mA
Corriente DC para pin de 3.3V	50mA
Memoria	32KB
Velocidad	16MHz

Para alimentar con energía eléctrica a la plataforma Arduino UNO se puede realizar mediante la conexión USB o mediante una fuente externa con un voltaje de 7-12V [8] y [31]. Cuando la plataforma es alimentada con una fuente externa (Adaptador AC-DC o batería), se deberá identificar la sección POWER (ver la figura 18) de la placa y conectar el polo positivo en el pin VIN y el polo negativo en el pin GRD, los pines “5V” y “3.3V” se utilizan para generar un voltaje inferior a 5 voltios (la placa se vuelve inestable [8]) en el primero y para 3.3 voltios generados por un regulador interno a 50 mA en el segundo.

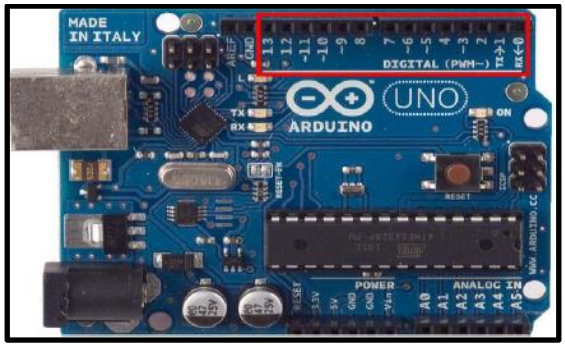
Figura 18: Pines de entrada de energía eléctrica [30].



Por otro lado, existen 14 pines de entrada y salida de información digital que pueden recibir como máximo 40 mA [31] y se encuentran en la sección del mismo nombre

(ver figura 19). Teniendo estos pines algunas funciones específicas se detallarán a continuación.

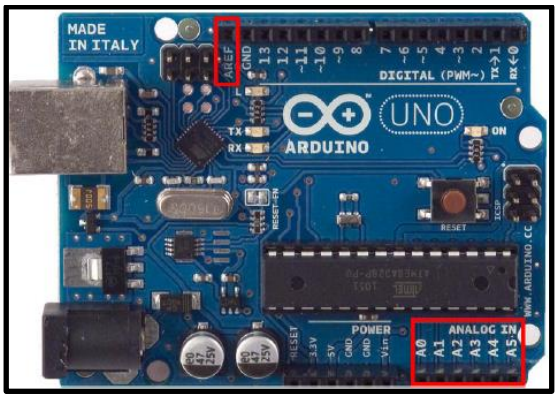
Figura 19: Pines de entrada y salida de información digital [30].



- Pin 0 y 1: Recepción (RX) y transmisión (TX) de datos de Lógica Transistor (TTL) [8] y [31].
- Pin 2 y 3: Encargados de interrumpir el programa secuencial establecido [8] y [31].
- Pin 3, 5, 6, 9, 10 y 11: Forman 8 bit de salida de modulación de ancho de pulso [8] y [31].
- Pin 10(SS), 11(MOSI), 12(MISO) y 13(SCK): se utilizan como apoyo a la comunicación serial de interface periférica (SPI) [8] y [31].
- Pin 13: Existe un LED conectado a este pin, que se enciende cuando el valor del voltaje en este pin es elevado [8] y [31].

Finalmente, la plataforma también tiene 6 pines de entrada (ver figura 20) de información analógica numeradas de A0 a A5 que trabajan por defecto con 5V [8] y [31]. Pero si se deseara modificar la tensión, se deberá de conectar el pin AREF y usar la función “analogReference” en el código de programación [8] y [31].

Figura 20: Pines de entrada de información analógica [30].



Aplicaciones de plataforma Arduino

Debido al bajo coste económico con que se puede acceder a este producto, Arduino ha sido aplicado en una gran cantidad de aplicaciones, llevadas a cabo por expertos y principiantes en el área de electrónica. Estas aplicaciones van desde proyectos de innovación de sistemas existentes hasta proyectos de investigación que buscan diseñar nuevos dispositivos tecnológicos. Algunos proyectos muy comunes que implican la utilización de la plataforma electrónica Arduino están subidos en la página oficial del producto.

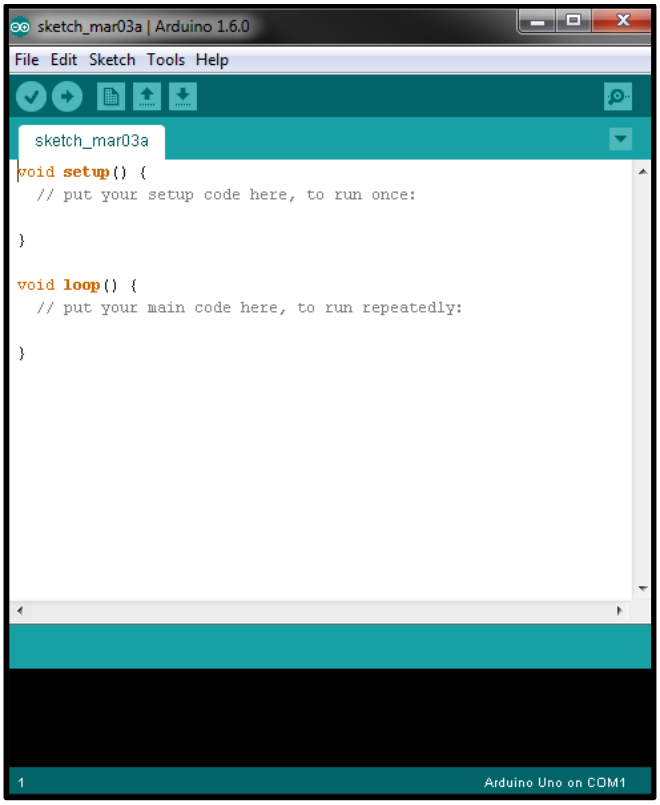
Finalmente, la meta de este trabajo de investigación es el uso de la plataforma Arduino para el diseño de un sistema de estabilización de un UAV. Remplazando los sistemas de control de los vehículos aéreos no tripulados, cuyas funciones (mantener estable y dirigir al UAV) serán programadas en un software mediante el lenguaje de comunicación compatible con la plataforma.

B) Editor de código de programación de Arduino

Para programar las placas Arduino, es necesario descargar el entorno de programación que estará disponible en la página oficial del producto. Si se ha de utilizar un Arduino USB se deberá de instalar unos drivers del chip FTDI (conversor USB a puerto serie) [8]. El entorno de programación denominado IDE (Integrated Development Environment) es un conjunto de herramientas de programación que

edita, compila, depura y construye interfaces para uno o varios lenguajes de programación [32] (ver la figura 21).

Figura 21: Interface del software IDE [31].



Los programas creados por el software IDE son llamados bocetos “Sketches” y cuyo archivo debe de ser guardado con la extensión “.ino” [33]. Así mismo, el lenguaje de programación de Arduino es extensible a código C o C++ [8], pero si se utiliza el lenguaje propio de Arduino, este al estar incluido en la librería IDE permite correr los programas usando solo dos funciones: setup() y loop() [34].

La función “setup()” se utiliza para iniciar el nuevo programa, una vez que el código este cargado en la plataforma [34]. Además, esta función permitirá que el programa inicie su ejecución automáticamente al alimentar con energía eléctrica a la plataforma. Se presenta a continuación un ejemplo, donde el pin trece es asignado como la salida de datos [34] (ver figura 22).

Figura 22: Ejemplo de función SETUP() [33].

```
void setup() {
    pinMode(13, OUTPUT);
}
```

Por otro lado, la función “loop()” es el núcleo de la mayoría de los proyectos. Permite que el programa se ejecute repetitivamente mientras la plataforma sea alimentada con energía eléctrica [34]. Un ejemplo de esta función se presenta a continuación en la figura 23.

Figura 23: Ejemplo de función LOOP() [33].

```
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

En el ejemplo anterior, el pin número 13 está conectado a un foco LED. El foco se encenderá mediante la función “digitalwrite(13,HIGH)” que enviara al foco una tensión de 5V [34]. La función “delay(1000)” permitirá la alimentación de tensión solo por un segundo, apagándose el foco a continuación, ya que la función “digitalwrite(13,LOW)” cortara la tensión por un segundo [34].

Finalmente, como se notó en los ejemplos anteriores, no se declararon variables dado que estos son ejemplos simples. Pero para un proyecto como el de diseño de software para un UAV, es mandatorio declarar las variables, cuyo número dependerá de la cantidad de elementos de entrada y salida que posee nuestro modelo. Por ejemplo, la cantidad de motores que se utilizan en un cuadricóptero son diferentes al de un helicóptero o hexacóptero, debiendo que declarar cada uno de estos motores. Las variables a declarar, son símbolos [34] utilizados en lenguaje

de programación para identificar cada uno de los componentes electrónicos utilizados [34].

C) Elementos de entrada de información

Los elementos de entrada son dispositivos diseñados para medir y detectar magnitudes, cambios o estados como la temperatura, presión, velocidad (lineal y angular) radiación, altitud, inclinación y otros [9] y [32]. La función de los elementos de entrada o sensores es de tomar información en forma de datos analógicos del medio exterior, que luego serán comparados con magnitudes escalares ideales.

Sensores, giroscopios y acelerómetros son utilizados principalmente para la estabilidad y navegación de los UAV [9]. Pero en diseños más avanzados y destinados funciones muy complejas se llega a utilizar sensores de presión, temperatura, proximidad, termografía y otros [9], ya que algunos UAV tienen la capacidad de realizar despegues y aterrizajes completamente autónomos o incluso realizan estas operaciones sobre bases móviles como barcos, aviones y vehículos terrestres. La expresión matemática que permite la lectura de las velocidades y aceleraciones de la IMU están dadas por la expresión 54 y 55 [35].

$$Ángulo\ Y = atan\left(\frac{x}{\sqrt{y^2 + z^2}}\right) \tag{54}$$

$$Ángulo\ X = atan\left(\frac{y}{\sqrt{x^2 + z^2}}\right) \tag{55}$$

Para nuestro trabajo de investigación, solo nos remitiremos a utilizar giroscopios y acelerómetros, ya que el alcance de este proyecto no contempla el rediseño de un UAV autónomo. Siendo por ende nuestro prototipo un vehículo aéreo no tripulado controlado remotamente.

D) Elementos de salida de información (Actuadores)

Los elementos de salida, son aquellos dispositivos que transforman la energía eléctrica, hidráulica o neumática en energía mecánica, térmica, etc. Para llevar a cabo esta transformación es necesario contar con un sistema electrónico (en nuestro caso Arduino) que permita procesar la información obtenida por los sensores o por comandos y/o programas (IDE) establecidos previamente. Los actuadores pueden ser de distintos tipos, dependiendo de su utilización que posean, los más comunes son: electrónicos, hidráulicos, neumáticos, eléctricos, motores, bombas, entre otros [32].

Los actuadores más frecuentados en los proyectos relacionados a UAV y en especial a nuestro tema de investigación son motores (la cantidad de motores dependerá del tipo de UAV), siendo más de tres motores si nuestro UAV fuese un multicóptero. Por otro lado, si el UAV fuese un aeroplano, se requeriría de servomotores, destinados al accionamiento de las superficies principales de control. La cantidad de actuadores definirán la cantidad de variables a declarar en el lenguaje de programación.

Además, los motores eléctricos utilizados en proyectos de electrónica, son en su mayoría motores de corriente eléctrica DC. Este tipo de motores están conformados por un estator, y un rotor compuesto por una bobina, lo cual los hace un poco más pesados y lentos, en comparación con los motores usados en los UAV que son de tipo “Brushless” (ver la figura 24). Este tipo de motores se caracteriza por la ausencia de un bobinado, que a diferencia de los anteriores, funcionan con corriente eléctrica trifásica AC, lo cual los hace superiores en cuanto a la relación potencia-peso y la eficiencia para transformar la energía eléctrica en movimiento mecánico [9] y [10].

Figura 24: Motor brushless [9].

E) Fuente de alimentación eléctrica

Las fuentes de alimentación utilizadas en la mayoría de los proyectos de electrónica se clasifican en dos tipos; las fuentes primarias o no recargables (siendo principalmente las pilas) y las fuentes secundarias, recargables llamadas baterías (ver figura 25) [9]. Asimismo, para obtener una rápida aceleración y duración de las baterías en los UAV, será necesario que la batería cuente con una buena potencia y energía [9].

Figura 25: Batería de polímero de litio (Li-Po) [8].

Por otro lado, la primera característica de las baterías es la capacidad de producir carga eléctrica, mediante un proceso electroquímico de sus componentes, y se mide en unidades de Coulomb (C) o Amperios (A). Otra característica se basa en la cantidad de tensión de salida y la capacidad de carga o descarga (medido en Ampere por hora A/h o mA/h), que establece la eficiencia de las baterías y es directamente proporcional al precio de los mismos [9].

Las clases de baterías más usadas en los UAV son:

Batería Níquel-Cadmio (Ni-Cd)

Este tipo de baterías fue usado en los inicios del desarrollo de los UAV. Y se caracteriza por poseer un voltaje de 1.2V y cuyas desventajas son la toxicidad del cadmio y el efecto memoria, que se da cuando se pone a cargar una batería que no fue consumida completamente, entonces interiormente no se satisface la carga completa de esta.

Batería de Níquel-Metal Hidruro (Ni-MH)

Este tipo de baterías fueron implementadas como remplazo de las Ni-Cd, y que es superior a las anteriores en cuanto a capacidad y la ausencia del efecto memoria. A pesar de que son limitadas para alimentar motores de alta potencia [9].

Batería de iones de litio (Li-ion)

Este tipo de batería genera un voltaje de 3.7V, siendo también más ligeras, con más capacidad que el Ni-Cd (el doble) y sin efecto memoria [9]. Pero que requiere de un circuito de control para regular el máximo voltaje de salida [9].

Batería de polímero de litio (Li-Po)

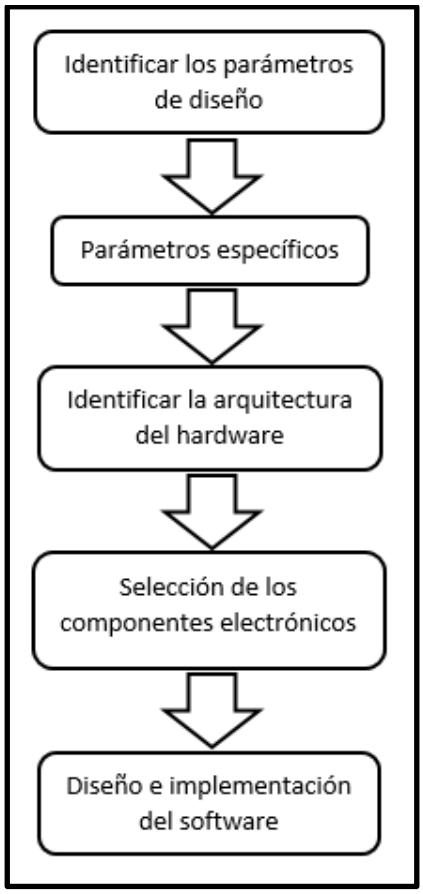
La capacidad de este tipo de baterías es de aproximadamente 15-20 veces a la del Ni-Cd, siendo además muy ligeras, sin efecto memoria y con un voltaje de 3.7V; son ideales para ser montados en cualquier parte y tienen la capacidad de alimentar motores de alta potencia [9]. Pero la desventaja más significante es que demora para recargarse de energía [9].

CAPÍTULO 3

MÉTODO DE SOLUCIÓN

El método de solución presentado en la figura 26 fue elaborado según la problemática y sus respectivos objetivos planteados en la primera parte de esta investigación. Los pasos presentados en el siguiente esquema que son parte del método de solución son cuatro. Donde la primera fase es la identificación de los parámetros de diseño, que prácticamente establecen una relación de complemento entre lo que se requiere para solucionar una problemática y lo que se tiene a nuestro alcance para mitigar dicha problemática. La segunda fase consiste en identificar los componentes electrónicos (incluyendo la plataforma Arduino) que se utilizan usualmente en los sistemas de estabilidad de los UAV. Posteriormente, se analiza y coteja minuciosamente cada componente propuesto, según su disponibilidad en el mercado peruano. Finalmente, una vez elegidos los componentes más idóneos para el proyecto, se procederá a diseñar los códigos de programación, para integrar todos los componentes en el sistema que se pretende construir.

Figura 26: Diagrama de flujo.



3.1. Parámetros de diseño

Los aspectos físicos y funcionales del dispositivo que se pretende diseñar, se seleccionaron mediante la comparación de las propiedades inherentes de los dispositivos convencionales.

Tabla 2: comparación de características de controladores de vuelo.

CARACTERISTICAS	PIXHAWK	CRIUS MULTIWII SE	KAKUTE F7
Costo (US$)	225	no especifica	59.29
Sensores (DOF)	9 + Barómetro	9+Barómetro	9+Barómetro
GPS	SI	SI	SI
Salidas PWM	16	11	no especifica
Voltaje de entrada (V)	4.9 -5.5	3.3	5
Max. voltaje (V)	6	5	42
Dimenciones (mm)	44x84x12	50x50	35x41x7
Masa (g)	15.8	14	9
Expandible	SI	SI	SI

Según la tabla 2, las características que más se alinean con los objetivos específicos de este trabajo de investigación son:

- El costo, que para nuestro diseño se deberá mantener a un presupuesto lo más económico posible, siendo el límite máximo 30 dólares.
- La cantidad de sensores, siendo común para los controladores convencionales el uso de IMU's de 9 grados de libertad e incluido un barómetro, pero para este proyecto se considera usar una IMU de 6 grados de libertad (acelerómetro y giroscopio). Lo cual se estima que va a satisfacer con la estabilización del UAV en perturbaciones de alabeo y cabeceo, que se consideran las más críticas en vuelo crucero.
- Un parámetro muy importante, considerado en este proyecto, es la disponibilidad o acceso de los componentes electrónicos que serán parte del estabilizador. Ya que es infructuoso la compra de los componentes mediante pedidos de correo internacional, demandando el envío una cantidad de tiempo muerto (no productivo para el proyecto) y un adicional de presupuesto.

3.1.1. Parámetros específicos

Por otro lado, es necesario también delinear la forma y capacidades de funcionamiento del sistema, dado que se considera infructuoso, por ejemplo, diseñar un controlador de vuelo con un peso excesivamente superior a los convencionales.

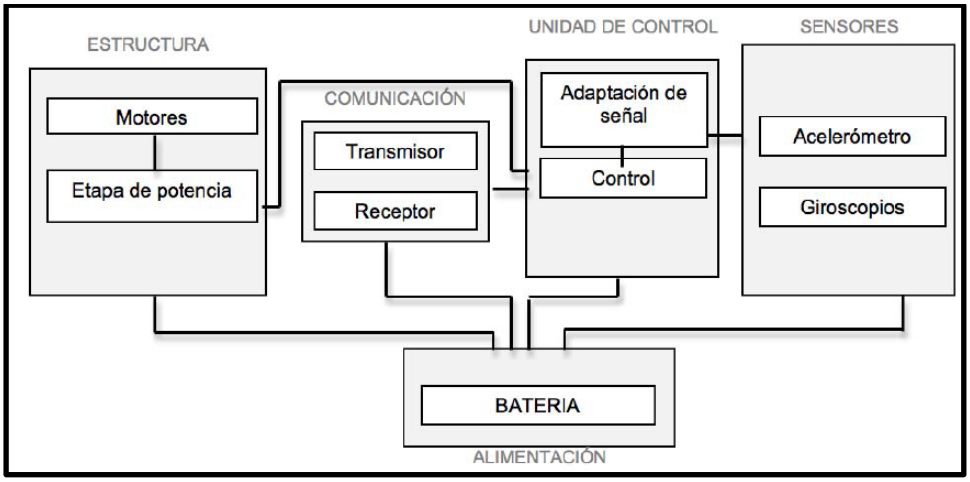
- El dispositivo que se pretende diseñar tendrá un voltaje de operación de entre 3.3V (tensión filtrada) y una entrada de voltaje de 5V. Estos voltajes de operación están determinados por las características de operación del microcontrolador Arduino.
- Los pines de salida o entrada digital del estabilizador serán 14 de los cuales 6 pines tienen la función PWM (Pulse With Modulation).
- El volumen del sistema de mantendrá a por debajo de 70x70 mm, siendo este límite muy razonable para la ulterior implementación del dispositivo en el UAV.

- Respecto al peso del sistema, se estima que superara el promedio de la tabla 2 pero manteniéndose a un nivel inferior a 80 g.
- Respecto a la capacidad de expansión del dispositivo, para poseer nuevas herramientas en proyectos posteriores, se considera también de importancia.

3.2. Arquitectura del hardware

En términos generales, los UAV son un conglomerado de componentes electrónicos eléctricos y mecánicos sincronizados entre sí. Y que debido a que cada UAV tiene características inherentes a la misión que estos desarrollan, algunos componentes también suelen variar unos de otros. Así, por ejemplo, un UAV que está diseñado para realizar vuelos de investigación en la atmosfera estará dotado de sensores de presión, temperatura, y otros. Sin embargo, los sistemas y algunos componentes encargados de la estabilidad y vuelo del UAV son los mismos ya que en general todos lo UAV obedecen a principios físicos aerodinámicos durante el vuelo. En la siguiente figura 27 se muestra un sistema de comunicación y estabilidad de un UAV multirotor.

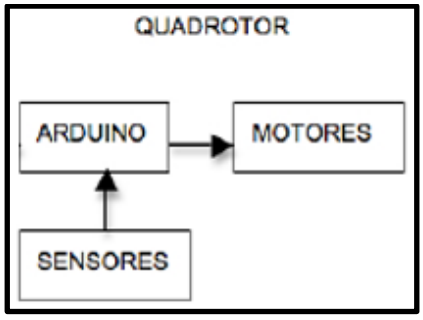
Figura 27: sistema de comunicación y estabilidad de un multirotor [8].



Por otro lado, los componentes principales para el control de la estabilidad y vuelo a bordo del UAV están compuestos por tres grupos: control de vuelo (controlador), actuadores y sensores [9] y [10] (ver figura 28). La fuente de alimentación o batería de la figura 27 es un componente importante para el sistema, por que suministra energía eléctrica a los demás

componentes, pero a pesar de su función, no forma parte del sistema de control y estabilidad del UAV.

Figura 28:Conjunto de componentes principales de un Quadrotor [8].



En la figura 28, el grupo de actuadores para un caudricóptero está compuesto por motores, que controlan al UAV (multirotores) mediante la variación del giro se los motores. Sin embargo, en las aeronaves de ala fija, las superficies de control son accionadas por servomotores.

La plataforma Arduino es el sistema de control o cerebro [10], que remplaza a los sistemas de control de vuelo que son implementados en los UAV desde la fábrica. Para lograr la misma efectividad que los sistemas comerciales de fábrica, la plataforma Arduino tiene que funcionar en conjunto con lo sensores (acelerómetro y giroscopio).

Además, la plataforma Arduino, los sensores y actuadores conectados, tienen que ser alimentados una fuente adicional de energía eléctrica de 9V [10]. Es decir que el UAV va a poseer una batería que alimente los motores, servomotores, entre otros; y una segunda batería pequeña para la plataforma Arduino, módulos giroscopios y demás componentes mencionados. Ya que el voltaje de la batería principal es demasiado elevado, lo cual se requeriría de un transformador para alimentar a dicho microcontrolador. Por otro lado, la batería principal es la encargada de alimentar a los motores (si se tuviese multirotores o UAV de ala fija con varios motores) y/o servomotores, cuya función es de accionar las superficies de control primarias del UAV. Pero la alimentación de la batería a los actuadores

no es directamente, sino que, la energía eléctrica tiene que fluir por unos controladores de velocidad llamados ESC “Electronic Speed Controller”.

3.3. Selección de plataforma electrónica y componentes

La selección de la plataforma electrónica que va a remplazar el sistema de estabilización del UAV, debe de ser igual o menor que el sistema convencional de fábrica, además de cumplir todas las funciones que tiene el sistema original.

3.3.1. Plataforma electrónica Arduino

Según la página oficial de Arduino, actualmente existe una amplia gama de plataformas electrónicas de este tipo, siendo las más comunes Arduino UNO, Arduino Mega, Arduino 101, Arduino Leonardo y otros. Así mismo, cada plataforma electrónica varía de los otros en la cantidad de pines de entrada/salida, peso, precios, disponibilidad y componentes. A continuación, se presenta una tabla de comparación entre las dos plataformas Arduino que se consideran las más idóneas para la ejecución del proyecto.

Tabla 3: Comparación de características entre Arduino UNO y 101.

PLATAFORMA ELECTRONICA	Arduino	
	UNO	101
Microcontrolador	ATMega328P	Intel Curie
Voltaje de operación (V)	5V	3.3
Voltaje de entrada recomendado (V)	7-12V	7-12 V
Voltaje de entrada maxima (V)	6-20V	7-17V
Pines I/O digital	14	14
Pines de entrada analogicos	6	6
Bluetooth	No	Si
Acelerometro	No	Si
Giroscopio	No	Si
Longitud (mm)	68.6	68.6
Ancho (mm)	53.4	53.4
Masa (g)	25	34

En la tabla 3 se puede observar que el voltaje de operación, el voltaje máximo y la masa son diferentes en cada modelo, mientras el voltaje de entrada, la cantidad de pines y las dimensiones son iguales para ambas plataformas [33].

Además, la plataforma 101 posee sensores de movimiento y un módulo bluetooth, que posibilita su conexión con un teléfono celular. El microcontrolador del 101 es un Intel Curie integrado (ver figura 29 en recuadro amarillo), mientras que Arduino UNO posee un microcontrolador desmontable ATMEGA328 (ver figura 30 en recuadro amarillo) [33]. Otra diferencia importante es la elevada masa de la plataforma 101, que es 34g [33], que inevitablemente va a repercutir en la variación de peso del sistema en general. Por estas diferencias y además del elevado costo de adquisición del Arduino 101, se decide elegir como el más idóneo al Arduino UNO.

Figura 29: Plataforma electrónica Arduino 101 con microcontrolador Intel integrado [32].

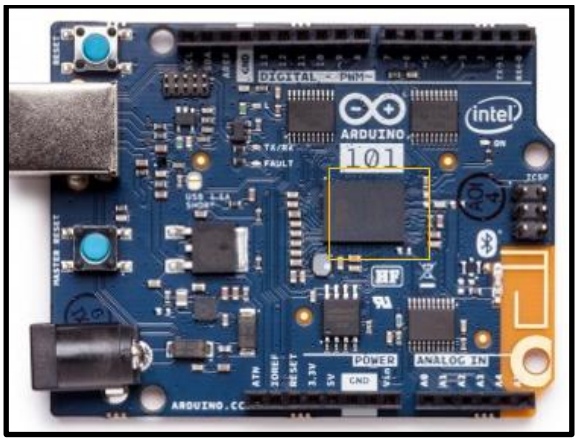


Figura 30: Plataforma electrónica Arduino UNO con microcontrolador ATMEGA328 removible [32].

3.3.2. Sensores

Al igual que las plataformas electrónicas Arduino, actualmente en el mercado existe una amplia variedad de sensores de todo tipo. Los tipos de sensores disponibles se encuentran desde medidores de presión, temperatura, movimiento hasta medidores de posición y dirección, que es lo que nos interesa. Y los precios también varían desde 10 soles hasta los 200 soles, según la función.

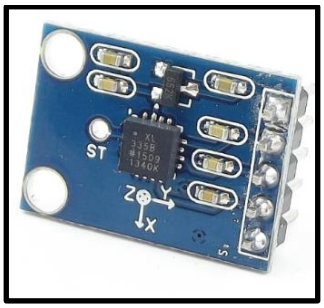
En este trabajo de investigación, los sensores a estudiar son los acelerómetros y giroscopios, que se encargan de medir la variación de aceleración generadas por fuerzas que afectan al móvil (UAV) y la posición de este en el espacio. La medición de dichas variaciones es necesaria para conocer el sentido y orientación del UAV, debido a que la acción restauradora de estas desviaciones (generada por los actuadores) depende de la intensidad de las variaciones medidas. Además, considerando los problemas específicos que se plantean en este trabajo, la selección de estos sensores tiene que ser a un bajo coste de adquisición y peso del componente.

Según Naylamp Mechatronics [36], los sensores acelerómetros y giroscopios disponibles y más comunes son el sensor Módulo ADXL335, Módulo MPU6050, Módulo MPU9250 y el Módulo GY-91 MPU9250+BMP280.

A) Módulo ADXL335

Este módulo está compuesto por un acelerómetro analógico de 3 ejes (Ver figura 31), capaz de convertir los datos medidos mediante la interacción con un microcontrolador tipo PIC, ATMEL o Arduino [36].

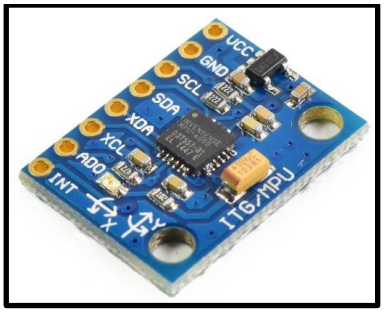
Figura 31: Modulo ADXL335 [35].



B) Módulo MPU6050

Este módulo está integrado por un acelerómetro de 3 ejes, un giróscopo de 3 ejes (Ver figura 32) y un Procesador Digital de Movimiento PDM que permite la ejecución de complejos algoritmos de medición en 9 ejes [36]. Para conectar este módulo con la plataforma Arduino, es necesario la implementación de la librería i2cdevlib [36].

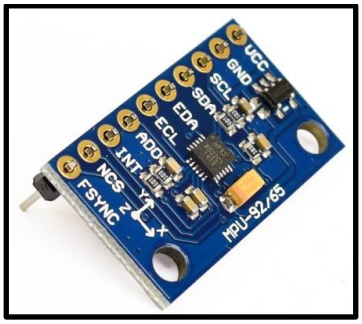
Figura 32: Modulo MPU6050 [35].



C) Módulo MPU9250

Este módulo es la integración de un acelerómetro de 3 ejes, un giroscopio de 3 ejes y un magnetómetro de 3 ejes, e implementado con un PDM, que entrega 9 grados de libertad (Ver figura 33) [36].

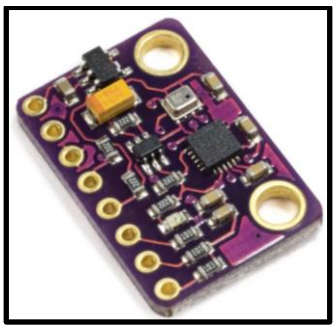
Figura 33: Modulo MPU9250 [35].



D) Módulo GY-91 MPU9250+BMP280

Este módulo es una integración contiene un acelerómetro, giroscopio, un magnetómetro y un altímetro, que proporciona 10 grados de libertad (Ver figura 34) [36]. Asimismo, viene implementado con una unidad inercial de medida MPU9250 y un barómetro BMP280, lo cual, al combinarlo con un GPS lo hace un dispositivo ideal para la aplicación en vuelos autónomos [36].

Figura 34: Modulo MPU9250+BMP280 [35].



A continuación, en la tabla 4 se presenta una comparación de las características más importantes de los módulos descritos.

Tabla 4: Comparación de sensores.

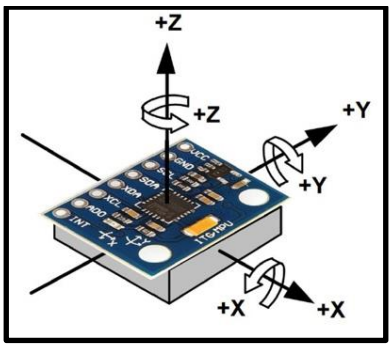
PROPIEDADES	MODULO			
	ADXL335	MPU6050	MPU9250	GY-91 MPU9250+BMP280
Voltaje de operación (V)	3.3-5	3.3-5	3.3-5	3.3-5
Consumo de corriente (µA)	350	No especifica	No especifica	No especifica
Sensibilidad	Si	Si (En Giroscopio)	No especifica	No especifica
Rango acelerometro (G)	3	2,4, 8, 16	2,4, 8, 16	2,4, 8, 16
Rango giroscopio (Grado/seg)		250, 500, 1000, 2000	250, 500, 1000, 2000	250, 500, 1000, 2000
Grados de libertad	3	6	9	10
Regulador de voltaje	Si	Si (En placa)	Si (En placa)	Si (En placa)
Rango Magnetometro (µT)	No especifica	No especifica	4800	4800

En la tabla 4, se puede observar que el voltaje de operación es el mismo para todos los módulos, y todos poseen un regulador de voltaje. Solo se conoce la corriente de operación del ADXL335 que es 350 miliamperios, además solo los módulos ADXL335 y MPU6050 poseen sensibilidad, mientras no se especifican en los otros. Sin embargo, el rango del acelerómetro y giroscopio es el mismo para todos los módulos a excepción del ADXL 335. Además, una diferencia importante de cada

módulo es la cantidad de grados de libertad, siendo el más completo el Módulo GY-91 MPU9250+BMP28, lo cual lo hace más costoso y pesado también.

Finalmente, para este trabajo se seleccionó el módulo acelerómetro, giroscopio MPU6050. Ya que, para el desarrollo de este trabajo debemos limitarnos a mantener un costo de adquisición y peso del sistema mínimos. Además, según la tabla 4, este módulo tiene casi las mismas características que los otros (rangos de acelerómetro y giroscopio, voltaje de operación y regulador de voltaje), brinda seis grados de libertad, 3 ejes que permiten medir la velocidad angular mediante el giroscopio y los otros 3 que miden las aceleraciones en las componentes XYZ mediante el acelerómetro (ver figura 35) [36]. Adicionalmente, el módulo cuenta con un regulador de voltaje integrado de 3.3 V que puede ser alimentado directamente por la plataforma Arduino.

Figura 35: Ejes de referencia del módulo acelerómetro y giroscopio MPU6050 [35].



3.3.3. Actuadores

Para la elección de los actuadores, se debe de tener en cuenta el tipo de aeronave donde se implementa (ala fija o giratoria). En el caso de un UAV de alas fijas de este proyecto, se va a utilizar un motor eléctrico (UAV monomotor) y 3 servomotores. Los tres servos serán distribuidos para accionar las superficies de control de la aeronave (timón de profundidad, timón de dirección y alerones). Además, téngase en cuenta que el motor deberá estar conectado con el ESC (Electronic Speed Controller).

A) Motor

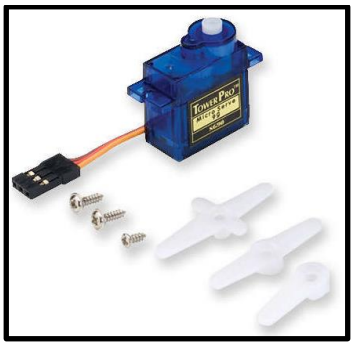
La elección del motor fue hecha tomando en cuenta el tamaño y la geometría del UAV, adicionalmente este componente no es de mucha importancia para esta investigación, ya que el motor solo estará reservado para proporcionar la potencia para el vuelo. Hecha la consideración anterior, se tomó como opción el motor brushless A2212 Kv 1 000 (Ver la figura 36), cuyo diámetro es de 3.175 cm e intensidad de corriente es de 12 A/60s [37]. El motor deberá estar conectado con un ESC 30 A y la hélice adecuada para este tipo de motor es 8 -10 cm de diámetro. Además este tipo de motor, a pesar de ser costoso frente a sus equivalentes con escobillas, es ideal para proyectos de este tipo debido a la posibilidad de ser conectado con baterías de tipo Lipo 2 a 3 celdas [37].

Figura 36: Motor eléctrico Brushless A2212 Kv 1 000 [36].

B) Servomotores

Según un amplio rango de opciones suministrada por la referencia [36]. Se eligió el servo motor de tipo Micro Servo SG90 de 1.8 kg de fuerza (ver figura 37). La elección de este tipo de actuador se hizo, considerando la carga que este ha de soportar durante las maniobras de vuelo y el peso que este posea (9 gramos) [36]. Además, este servo es compatible con la mayoría de las plataformas Arduino (incluido Arduino UNO) y cuyo voltaje de operación está entre 3-7.2 voltios [36].

Figura 37: Servomotor Micro Servo SG90 de 1.8 kg [35].



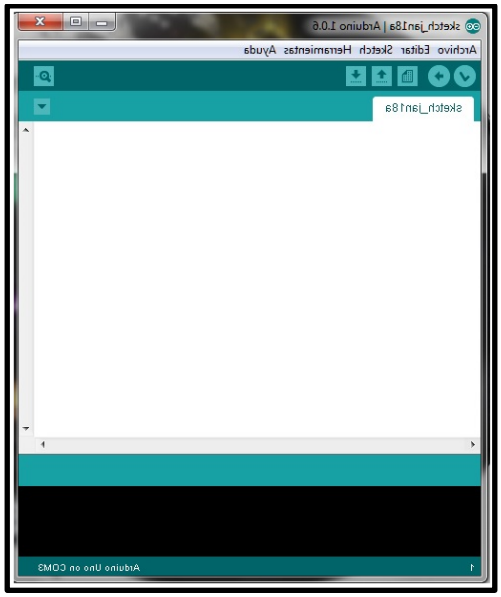
3.4. Diseño e implementación del software

Antes de iniciar con la descripción del diseño e implementación del software, es menester mencionar que estas actividades se realizaran, considerando la comunicación entre la plataforma Arduino-radio control y el sistema de estabilidad que lleva a bordo el UAV [10].

3.4.1. Diseño del programa en Arduino UNO

El procedimiento de implementación del programa se inicia con la conexión de la plataforma electrónica (microprocesador) a la computadora [38], paso seguido se abre la ventana “entorno de desarrollo” IDE (ver figura 38).

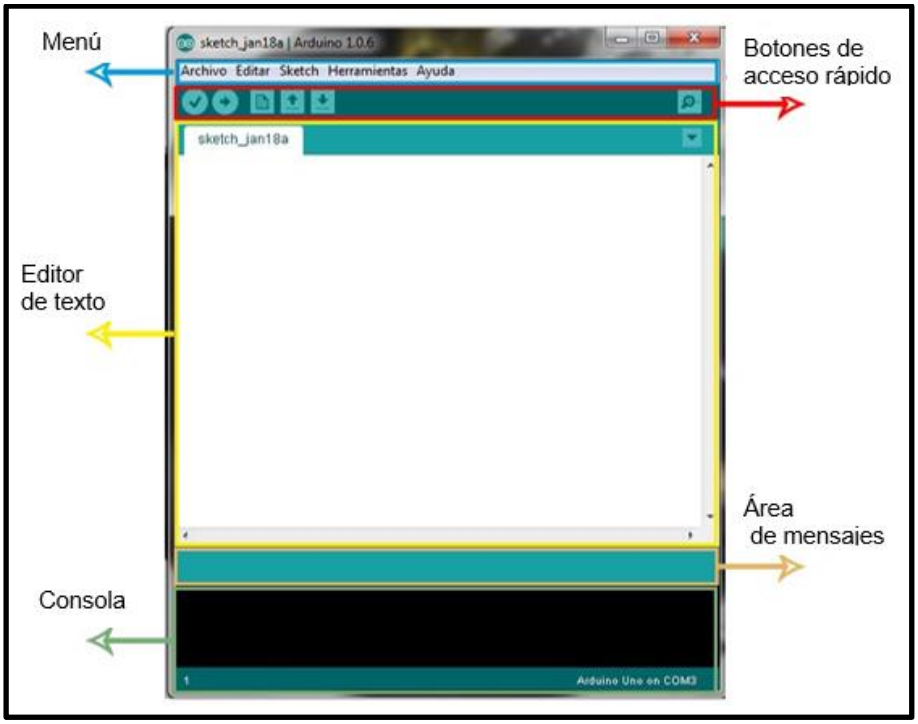
Figura 38: Entorno de desarrollo IDE [37].



Después, se carga o redacta el programa deseado en el editor de textos (ver figura 39) y se pulsa el segundo botón de acceso rápido para descargar el programa en el

microprocesador (ver figura 39) , y una vez que se alimente una tensión de 9 voltios a la plataforma electrónica, este empieza a ejecutar el programa [10].

Figura 39: Entorno de desarrollo y sus partes [37].



3.4.2. Implementación del software

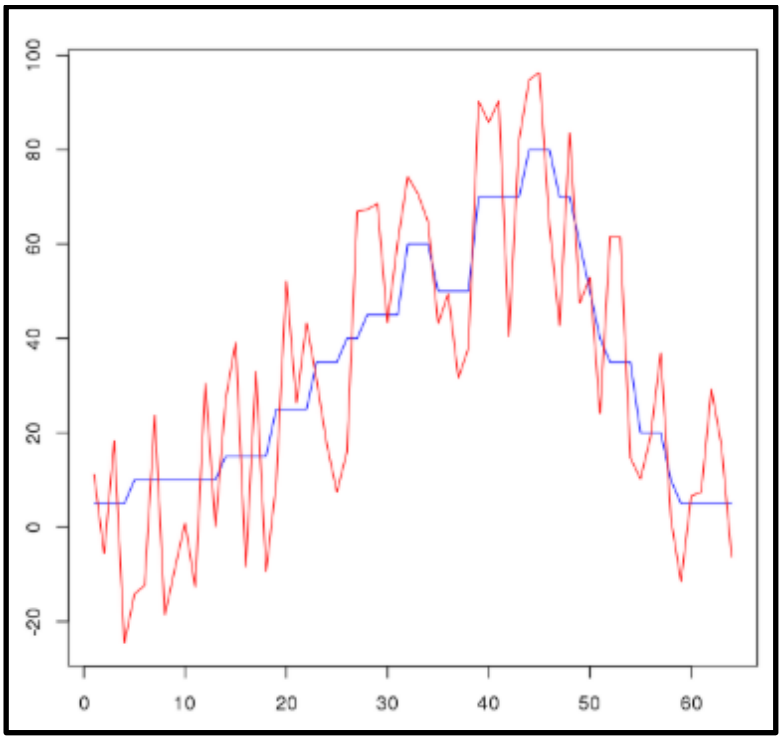
Como se mencionó anteriormente, el proceso de implementación se divide en dos secciones, la primera parte está enfocada al cálculo de los ángulos de Euler obtenidos de las mediciones del módulo MPU, mientras en la otra sección se menciona la comunicación entre la plataforma Arduino y los actuadores (servomotores).

A) Lectura de la Unidad Inercial de Medida (IMU)

Las lecturas obtenidas de las IMU's, teóricamente deben generar ángulos precisos, gracias a las mediciones de los acelerómetros y giroscopios [10] y [39]. Pero experimentalmente, la presencia de ruidos blancos y errores producen ciertas imprecisiones y distorsiones de los valores de los ángulos medidos (ver figura 40) [39].

En la figura 40 se puede observar que las medidas de los ángulos ideales están representadas por la línea azul, mientras las medidas reales se representan por la línea roja.

Figura 40: Diferencia de una medición real e ideal de una IMU [38].



Así mismo, los errores son producto de los ruidos, que generan interferencia en las mediciones; cualquier pequeño desplazamiento en uno de los ejes del acelerómetro se confundirá con una rotación [39]. Por otro lado, a pesar de que los giroscopios tienen mediciones un poco más precisas que los acelerómetros, los errores generados se suelen ir acumulando con el tiempo [39]. Considerando finalmente, como método de solución frente a este inconveniente la utilización de diferentes tipos de filtros de complementación, siendo el método más utilizado el filtro de Kalman.

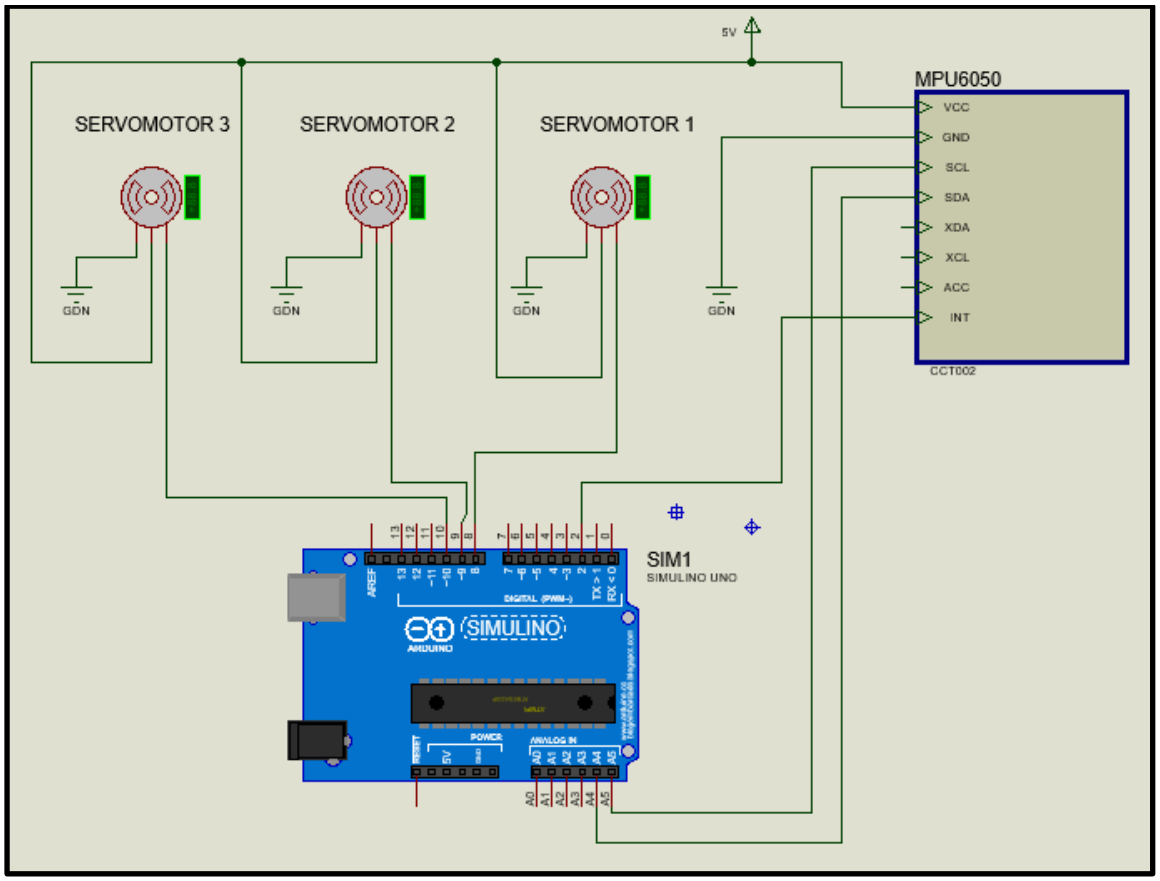
B) Conexión del módulo MPU6050, Arduino UNO y los servomotores SG90

Antes de empezar con la implementación de los algoritmos para el control de los 2 servomotores que controlaran el UAV es necesario conectar todos los componentes seleccionados anteriormente (Ver figura 41). El servo SG90 posee tres cables de

conexión, donde el cable amarillo es la conexión de datos, el cable rojo es la alimentación de tensión positiva y el cable marrón es la conexión a tierra o negativo. El cable de datos de un servomotor se debe conectar al pin de salida digita 8 (Ver figura 41) y el cable del otro servomotor se conecta con el pin 9. Mientras los otros cables sobrantes se alimentan por una batería.

Por otro lado, la conexión del módulo MPU6050 se realiza con los pines VCC, GND, SCL y SDA en el mencionado, y los pines 3.3V, GND, A4 y A5 de la plataforma Arduino, respectivamente (ver la figura 41).

Figura 41: Conexión de la plataforma Arduino Uno, módulo MPU6050 y servomotor SG90.



C) Vinculación de la plataforma Arduino con la computadora

Una vez realizada todas las conexiones de la sección anterior, se procede a enchufar el puerto USB de la plataforma Arduino con la computadora y posteriormente se abre el editor de textos de Arduino previamente instalado en el

ordenador. Paso seguido se selecciona la pestaña “herramientas”, opción “Fuente” y finalmente seleccionar el nombre del microcontrolador “ATMega328”.

D) Implementación del algoritmo de control

Una vez vinculada la computadora y la plataforma, es necesario comenzar con la edición del algoritmo de control en el panel “editor de texto”. Para ello se empieza con la inclusión de algunas librerías que ya están cargadas en el software. El siguiente código de programación fue tomado de Humberto Higinio.

Las funciones “#include<>” llaman o incluyen las librerías SPI, Wire y Servo. Mientras que la función “#define” define la dirección I2C del MPU-6050, que es un bus de transmisión de datos en serie.

```
#include <SPI.h>
#include <Wire.h>
#include <Servo.h>
#define MPU 0x68
```

Seguidamente, la función “Servo” asigna los nombres a cada servo “ServoX” y “ServoY”, y la función “Double” asigna a las variables AcX, AcY AcZ doble precisión y finalmente, la función “int” declara a las variables Pitch y Roll como valores enteros.

```
Servo ServoX, ServoY;
double AcX,AcY,AcZ;
int Pitch, Roll;
```

A continuación, la función “Setup()” inicia su única ejecución, donde declara una transferencia de datos de 9600 bps mediante “Serial.begin()”, conecta los servos X y Y a los pines 8 y 9 respectivamente, y finalmente inicia la ejecución del módulo MPU6050.

```
void setup(){
  Serial.begin(9600);
  ServoX.attach(8);
  ServoY.attach(9);
```

```
  init_MPU();
}
```

Por otro lado, la función “loop()” que contiene la parte principal del código, empieza su ejecución con la obtención de los ejes AcX, AcY y AcZ proporcionados por la función “FunctionsMPU()”; posteriormente, calcula los ángulos Pitch y Roll, mientras se declara dichos valores como enteros (función “int”). Posteriormente, se registra los valores de “ServoRoll” y “ServoPitch”, para posteriormente proporcionarlos mediante las funciones “Serial.print()”.

```
void loop()
{
  FunctionsMPU();

  Roll = FunctionsPitchRoll(AcX, AcZ, AcY);
  Pitch = FunctionsPitchRoll(AcY, AcX, AcZ);

  int ServoRoll = map(Roll, -90, 90, 0, 179);
  int ServoPitch = map(Pitch, -90, 90, 179, 0);

  ServoX.write(ServoRoll);
  ServoY.write(ServoPitch);


  Serial.print("Pitch: "); Serial.print(Pitch);
  Serial.print("\t");
  Serial.print("Roll: "); Serial.print(Roll);
  Serial.print("\n");

}
```

En la función “intit_MPU()” que está incluida en la función “Setup()”, se inicia la ejecución de la librería “Wire” mediante la función “Wire.begin()” que conectara al bus maestro con Arduino y seguidamente se empieza la transmisión de datos con el MPU. Asimismo, se registra los datos mediante el inicio de la ejecución del MPU6050, para luego terminar la transmisión de datos y esperar un segundo.

```
void init_MPU()
{
  Wire.begin();
  Wire.beginTransmission(MPU);
```

```
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  delay(1000);
 }
```

En este segmento del código, se asigna doble precisión a los valores de A, B, C y sus respectivos “Dato”. Asimismo, DatoA es igual a A, DatoB es igual al valor de B al cuadrado más el valor de C al cuadrado y el dato final de B es igual a la raíz cuadrada del anterior “DatoB”. Por otro lado, el primer valor de “Value” es igual al arco tangente al cuadrado de “DatoA” y “DatoB”, siendo este valor convertido a grados segsagesimales.

```
 double FunctionsPitchRoll(double A, double B, double C){
  double DatoA, DatoB, Value;
  DatoA = A;
  DatoB = (B*B) + (C*C);
  DatoB = sqrt(DatoB);

  Value = atan2(DatoA, DatoB);
  Value = Value * 180/3.14;

  return (int)Value;
  }
```

Finalmente, la función “FunctionsMPU()” obtiene los ejes XYZ del módulo MPU. Se inicia con la transmisión de la librería “Wire” del MPU, luego se registra la aceleración en eje X. La función “Wire.endtransmission(false)” indica que la transmisión no se detiene, además se requieren 6 mediciones del MPU para que las funciones “Wire.read()” lea los datos obtenidos desde el MPU.

```
 void FunctionsMPU(){
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,6,true);
  AcX=Wire.read()<<8|Wire.read();      // 0x3B (ACCEL_XOUT_H) & 0x3C
(ACCEL_XOUT_L)
```

```
  AcY=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E
(ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E
(ACCEL_YOUT_L)
 }
```

Adicionalmente, el software Arduino posee dos herramientas que permiten acceder a los datos generados y procesados por el MPU y la plataforma electrónica. Al seleccionar la opción “Herramientas”- “Serial plotter” (Ver figura la 42 elaborada por el autor de este trabajo) se representa gráficamente la variación de las aceleraciones y posiciones, censadas por el módulo (Ver la figura la 43 elaborada por el autor de este trabajo). Por otro lado, si se selecciona nuevamente la opción “Herramientas” y la pestaña “Monitor serie” (Ver la figura la 44 elaborada por el autor de este trabajo), la información obtenida por el módulo y la plataforma electrónica se presenta en forma de datos (Ver la figura la 45 elaborada por el autor de este trabajo).

Figura 42: Selección de la representación gráfica de las aceleraciones y posiciones.

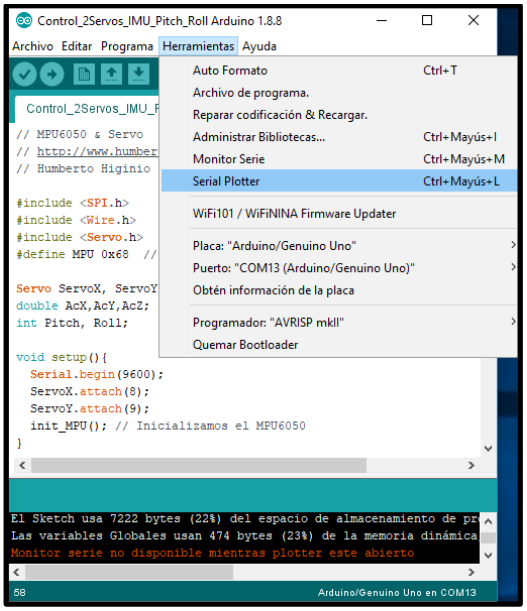


Figura 43: Gráfico de la variación de aceleraciones y posiciones del módulo.

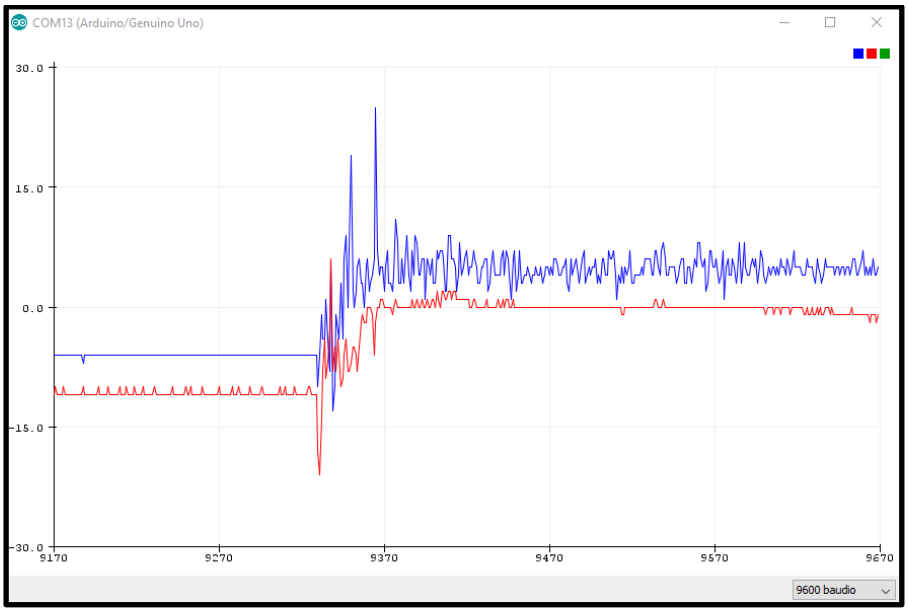


Figura 44: Selección de la representación en forma de datos de las aceleraciones y posiciones.

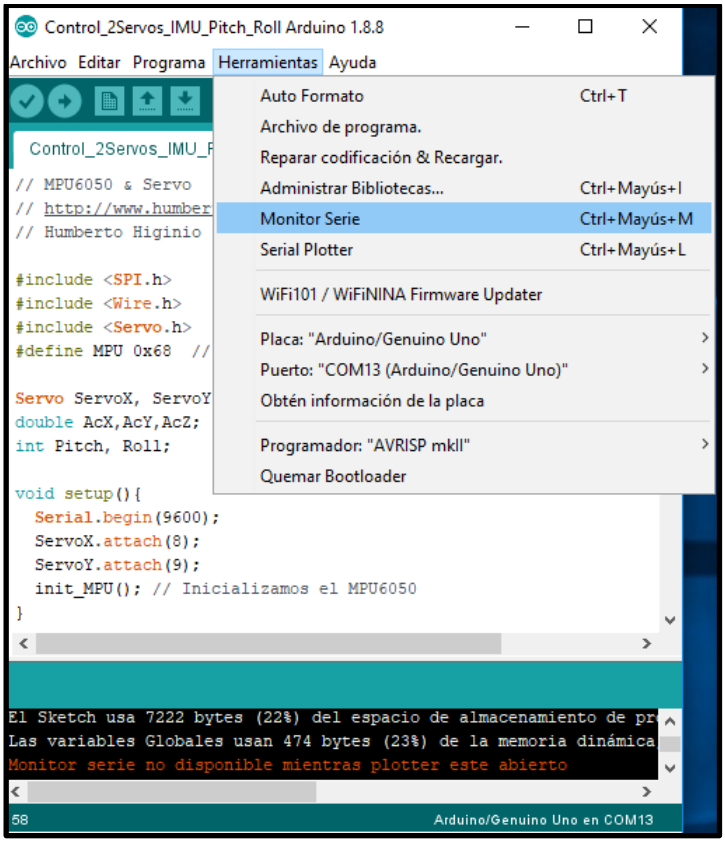
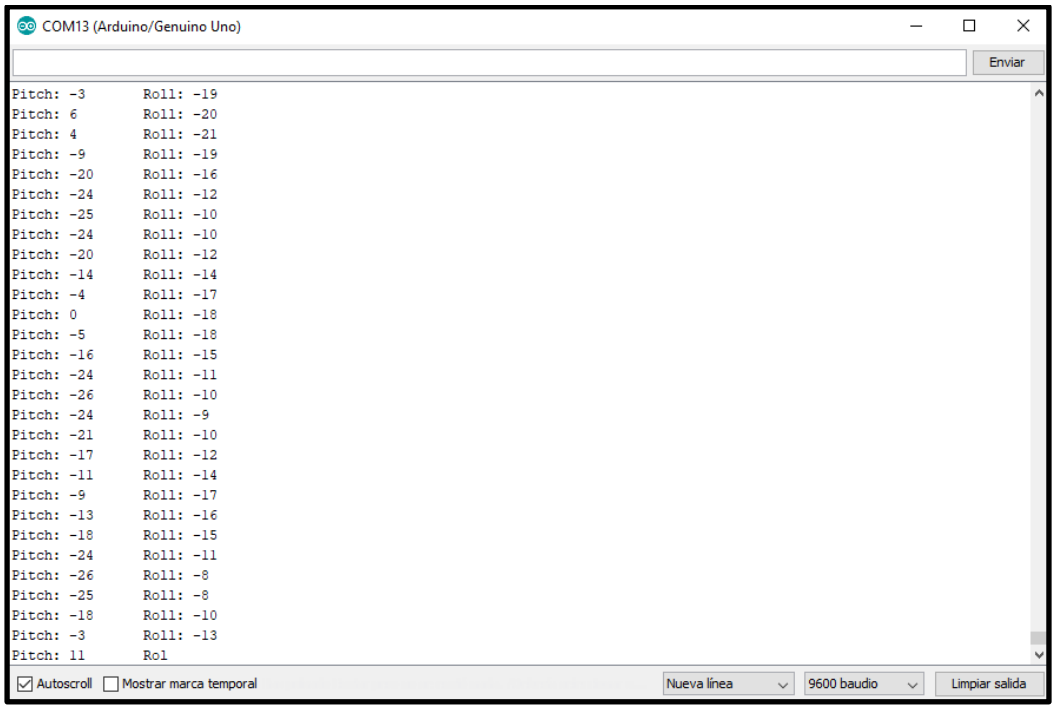


Figura 45: Representación en forma de datos de las aceleraciones y posiciones.



E) Implementación del código con control PID.

La implementación del control PID se incorpora en el código que la plataforma Arduino ejecuta para leer las mediciones de los ángulos de la IMU y enviar las señales eléctricas para accionar los Servomotores. Para generar el código con el controlador PID, se tomó de un algoritmo PID prediseñado para la estabilidad de un cuadrotor en el eje X [40]. Posteriormente, se modificó el código para adaptarlo al control de un UAV de alas fijas. El código completo se muestra en el anexo 3.

En la sección de declaración de variables del código se menciona las constantes, valores iniciales y variables que serán parte del código, donde los comandos “float” y “doublé” asignan valores con puntos decimales.

```
float elapsedTime, time, timePrev;
float PID1, PID2, error1, error2, previous_error;
float pid_p1=0;
float pid_i1=0;
```

```
float pid_d1=0;
float pid_p2=0;
float pid_i2=0;
float pid_d2=0;
```

Posteriormente se asigna valores decimales para las constantes proporcional, integral y derivativo.

```
double kp=3.55;
double ki=0.005;
double kd=2.05;
```

A continuación, se establece la posición inicial de los servos y se le da un valor entero de 90 grados, mientras se desea establecer un ángulo de 0 grados de la IMU.

```
int pos=90;
float desired_angle=0;
```

Ulteriormente, la parte repetitiva del código “void loop()” se procede a realizar los cálculos del control PID.

```
error1=Total_angle[0]-desired_angle;
error2=Total_angle[1]-desired_angle;
```

Seguidamente, se multiplica el valor establecido de cada constante por el error calculado.

```
pid_p1=kp*error1;
pid_p2=kp*error2;
pid_i1=pid_i1+(ki*error1);
pid_i2=pid_i2+(ki*error2);
pid_d1=kd*((error1-previous_error)/elapsedTime);
pid_d2=kd*((error2-previous_error)/elapsedTime);
```

Después, se suma la parte proporcional, integral y derivativa

```
PID1=pid_p1+pid_i1+pid_d1;
PID2=pid_p2+pid_i2+pid_d2;
```

Seguidamente, se emplea la condicional, que significa si “PD1” es menor a menos 90 entonces “PID1” es igual a -90 y si, por el contrario, si “PID1” es mayor a 90, entonces “PID1” es igual a 90.

```
  if(PID1<-90)
{
  PID1=-90;
}
if(PID1>90)
{
  PID1=90;
}
if(PID2<-90)
{
  PID2=-90;
}
if(PID2>90)
{
  PID2=90;
}
```

Posteriormente, los valores de “PID#” restan al valor de “pos”, que es 90 grados.

```
aileron1=pos+PID1;
aileron2=pos-PID1;
VT=pos+PID2;
```

Finalmente, se ordena a los servomotores desplazarse según los valores de “aileron#” y “VT”, donde si “aileron” es menos 90 entonces la posición del servo será 0 grados y si es 90 la posición será 179 grados y el comando “write()” representa la ejecución de la orden al servomotor . Note que para la máxima posición del servo se establece 179 grados, para evitar daños en el elemento.

```
  //right
  int Servoright=map(aileron1, 0, 180, 0, 180);
  right.write(Servoright);
  //left
  int Servoleft=map(aileron2, 0, 180, 0, 180);
  left.write(Servoleft);
  //vtail
  int Servovtail=map(VT, 0, 180, 0, 180);
  vtail.write(Servovtail);
 }
```

Nota: el carácter “}” cierra toda la función “void loop()”.

CAPÍTULO 4

ANÁLISIS DE RESULTADOS Y DISCUSION

El diseño final del sistema de estabilización tiene una masa de 40 g, siendo relativamente más pesado que los sistemas autopilotos convencionales (Pixhaw, Kakute, otros). Sin embargo, las funciones de estabilización del sistema satisfacen las acciones restauradoras en el eje longitudinal (alabeo) y lateral(cabeceo) del avión. La estabilización del avión en el eje vertical (guiñada) no se considera, ya que esta desviación no es crítica por que la variación del ángulo de guiñada en condiciones de vuelo crucero no termina con un impacto directo de la aeronave.

Por otro lado, la parte del hardware del sistema de estabilización tiene las dimensiones generales presentadas en la tabla 5, que, comparado con los autopilotos convencionales, el volumen excede aproximadamente en 77.6 centímetros cúbicos, debido a que la plataforma electrónica Arduino UNO tiene considerables dimensiones.

Tabla 5: Dimensiones del sistema diseñado.

DIMENSIONES	
Longitud (cm)	8
Ancho (cm)	6.5
Altura (cm)	2

En cuanto a la lectura de las señales registradas por la Unidad de Medición Inercial MPU 6050, en la figura 46 y 47 se observa que la variación de los ángulos con respecto a la velocidad de transferencia de datos (baudios) describe una gráfica suave, no existen los picos de amplitud registrados en la figura 43. Este cambio en la gráfica mostrada, es debida a la presencia del controlador PID en el código de programación. La asignación de los

valores de las constantes proporcional, integral y derivativa se hizo mediante un método experimental.

Figura 46: Variación del ángulo de cabeceo (línea roja).

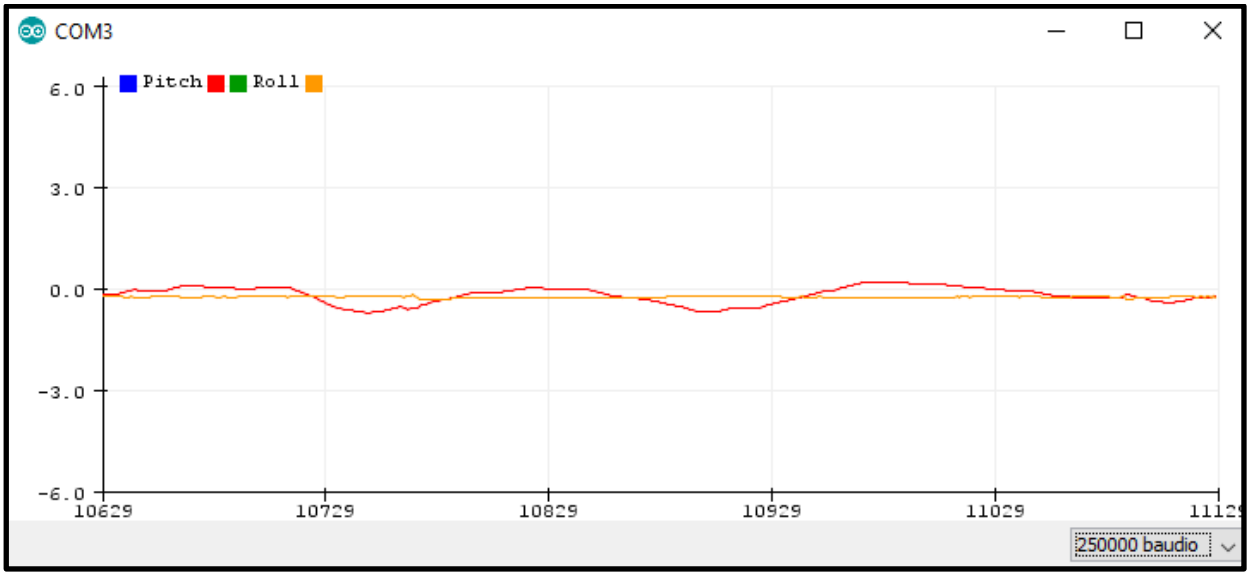
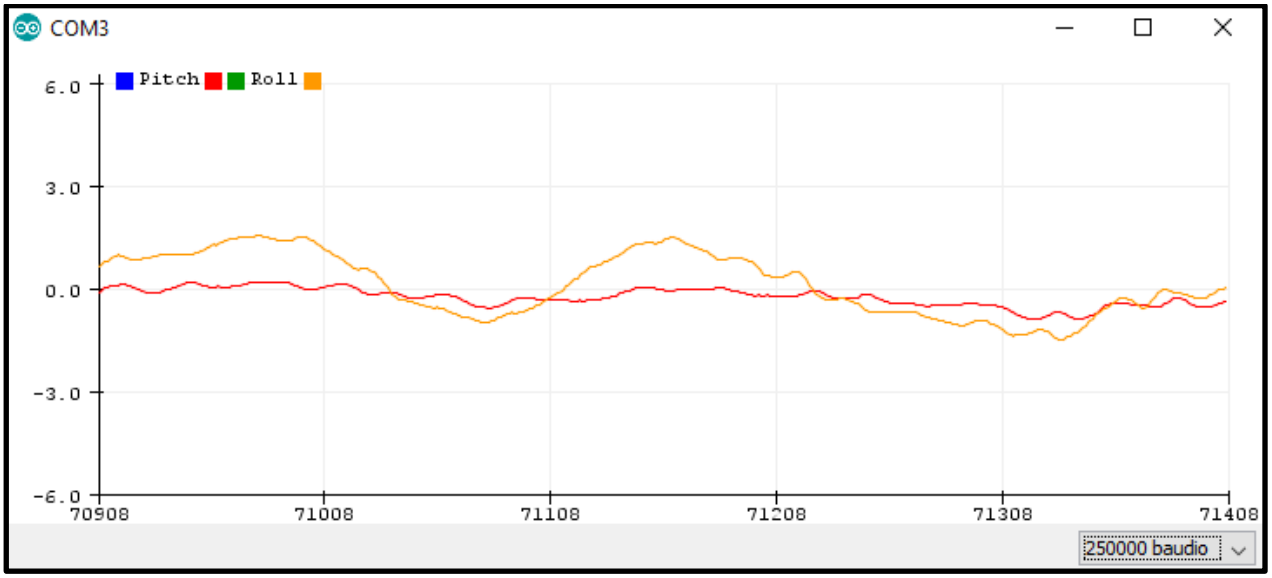
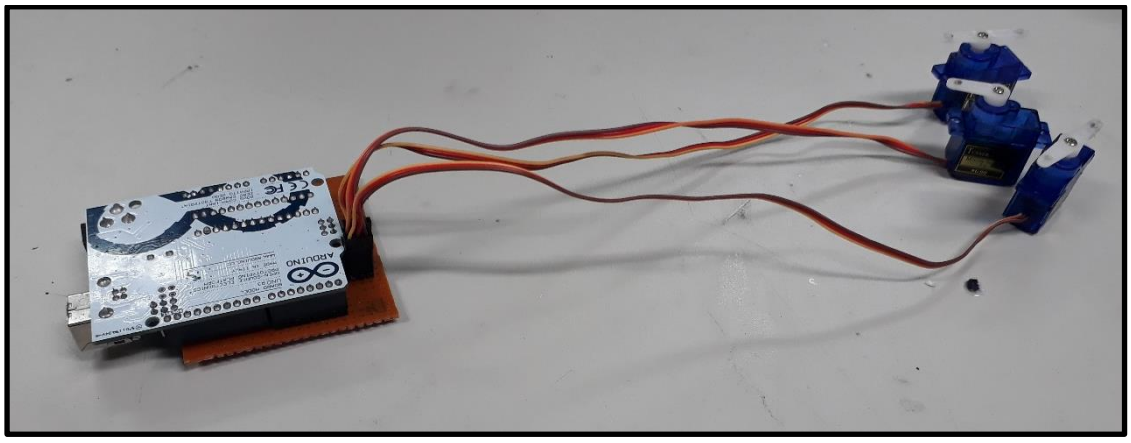


Figura 47: Variación del ángulo de alabeo (línea naranja).



La figura 48 es el diseño final del sistema de estabilización, conectado a los tres servomotores que accionan las superficies de control (alerones y timón de profundidad).

Figura 48: Sistema de estabilización diseñado.



El costo de adquisición de los componentes del sistema de estabilización, incluidos tres servomotores SG90, se presenta en la tabla 6.

Tabla 6: Costo unitario y total de los componentes electrónicos.

Componente	Cantidad	Costo unitario (S/.)	Costo (S/.)
Arduino UNO R3	1	30.00	30.00
MPU 6050	1	14.50	14.50
Servomotor SG90S	3	8.50	25.50
Placa PCB	1	1.00	1.00
Set de Cables	2	6.50	13.00
Set de pines	1	1.00	1.00
			85.00

Al comparar el costo total del sistema diseñado, con el costo de adquisición de los sistemas autopilotos convencionales, se llega a la conclusión de que el costo monetario del dispositivo es inferior a los sistemas convencionales en el mercado. Pero este costo tiene como penalización un relativo incremento del volumen y peso del dispositivo. Además, los sistemas convencionales poseen muchas otras funcionalidades como barómetros, magnetómetros, pero las necesidades principales de estabilización longitudinal y lateral de los UAV son alcanzadas por el nuevo sistema.

CONCLUSIONES

- Se diseñó satisfactoriamente un sistema de estabilización basado en Arduino UNO para un avión no tripulado con motor eléctrico, cuya masa es 40 g y sus dimensiones son 8 cm de largo 6.5 cm de ancho y 2 cm de alto.
- El sistema de estabilización corrige las perturbaciones de cabeceo y alabeo de manera simultánea, donde el tiempo en que los servomotores responden a las perturbaciones es de 1 segundo. Además, gracias a la implementación del controlador PID, las señales registradas por el IMU presentan una gráfica suavizada.
- La adquisición de los componentes del sistema de estabilización se realizó en tiendas locales, especializadas en electrónica, con lo que se concluye que existe la disponibilidad de los elementos en el mercado peruano.
- El costo total de los componentes requeridos para la construcción del sistema de estabilización es S/. 85.00, siendo este valor muy accesible en relación con el costo de los autopilotos convencionales.

RECOMENDACIONES

- La reducción del peso y volumen del sistema diseñado se puede alcanzar mediante la implementación de otro tipo de plataforma electrónica, que requerirá una nueva distribución de los componentes utilizados.
- Se recomienda implementar magnetómetros y barómetros para medir la altura de vuelo del UAV, pero que adicionará otro código de programación al principal e incrementará el costo total de diseño.
- Para alcanzar un adecuado nivel estético del circuito eléctrico PCB es recomendables elaborar un circuito electrónico impreso mediante el software Proteus.

BIBLIOGRAFIA

[1] J. Morales, *"Control para la navegacion pre-programada de trayectorias de un vehiculo aereo no tripulado (UAV) aplicado a la supervision y transmision en linea de la calidad del aire",Tesis de grado, Escuela Superior Politécnica de Chimborazo,* Riobamba, 2016.

[2] R. Beard, W. Johnson, R. Christiansen, J. Hintze y T. Mc Lain, «"Programble autopilot system for autonomous flight of unmanned aerial vehicles"». Estados Unidos Patente US-7302316B2, 16 Marzo 2006.

[3] A. Ahmed, B. Gamal, A. Ouda, A. Kamel y Y. El-Halwagy, «Autopilot design of unmanned aerial vehicle,» *Journal of Aeronautics and Aerospace Engineering,* vol. 7, nº 2, p. 7, 2018.

[4] A. Pico, *"Diseño e implementación de uns sistema de control para un cuadricóptero", Tesis de grado, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional,* México, Distrito Federal, 2012.

[5] F. Paredes, *"Vehículo aéreo no tripulado basado en control de lógica difusa", Tesis de grado, Universidad Nacional de Ingeniería,* Lima, 2012.

[6] Y. Sanchez, *"Diseño e implementación de un controlador PID-Difuso para el vuelo parado de un robot UAV tipo quadrotor", Tesis de grado, Universidad Piloto de Colombia,* Bogota DC, 2015.

[7] R. Lancaster, *"Formation flight autopilot design for the GAF Jidivik MK 4A UAV", Tesis de grado, Cranfield University,* Cranfield, 2004.

[8] R. Enríquez, *"Guía de usuario de arduino",* Cordoba: I.T.I. Sistemas, 2009.

[9] C. Nadales, *"Control de un quadrotor mediante la plataforma arduino", Tesis de titulación, Universidad Politécnica de Catalunya,* Barcelona, 2009.

[10] X. Legasa, *"Cuadricoptero arduino por control remoto android", Tesis de titulación, Universidad Politécnica de Catalunya,* Barcelona, 2012.

[11] J. Barton, «"Fundamentals of Small Unmanned Aircraft Flight",» *Johns Hopkins APL Technical Digest,* vol. 31, nº 2, p. 18, 2012.

[12] A. Barrientos, J. Del Cerro, R. San Martin y A. Martinez, «"Vehículos aéreos no tripulados para uso civil. Tecnología y aplicaciones",» *Grupo de Robotica y cibernetica, Universidad Politecnica de Madrid,* nº 266245324, p. 30, 2015.

[13] K. Valavanis y G. Vachtsevanos, Handbook of unmanned aerial vehicles, Springer Dordrecht, 2015.

[14] M. Haluani, *"Tecnología aviónica militar en los conflictos asimétricos: Historia, tipos y funciones de los drones letales", Universidad Simón Bolívar,* Caracas, 2014.

[15] M. Mamani, *"Diseño e implementación de interfaz cerebro-computacional para el control de vuelo de vehículo aéreo no tripulado usando señales cerebrales a travez*

de headset electroencefalografo", Tesis de titulación, Universidad Nacional de San Agustin, Arequipa, 2018.

[16] F. Sarche y K. Vásconez, *"Diseño y construccion de un prototipo de tricoptero controlado de forma remota mediante radiofrecuencia", Tesis de titulación, Escuela Politecnica Nacional,* Quito, 2012.

[17] U.S. Department of transportation federal aviation administration, Rotorcraft flying handbook, Washington DC: U.S. Department of transportation Federal Aviation Administration, 2000.

[18] M. Sadraey, Aircraft design: A systems engineering approach, New Hampshire: John Wiley & Sons, Ltd, 2013.

[19] U. Yayli, C. Kimet, A. Duru, O. Cetir, U. Torun, A. Aydogan, S. Padmanaban y A. Ertas, «"Design optimization of a fixed wing aircraft",» *Advances in Aircraft and Spacecraft Science,* vol. 4, nº 1, p. 17, 2017.

[20] M. Cook, Flight dynamics principles, Cranfield: Butterworth-Heinemann, 2007.

[21] R. Hernández, *"Análisis de la dinámica de vuelo de un minihelicóptero de diseño y construcción nacional", Tesis de grado, Instituto Politécnico Nacional,* Mexico D.F., 2007.

[22] M. Gómez, M. Pérez y C. Puentes, Mecanica de vuelo, Madrid: Garceta, 2012.

[23] D. Garijo, J. López y I. Pérez, *"Control de un vehiculo aereo no tripulado", Universidad Complutense de Madrid,* Madrid, 2009.

[24] M. Khan, «"Quadcopter flight dynamics",» *International Journal of Scientific & Technology Research,* vol. 4, nº 8, p. 6, 2014.

[25] A. Singh, S. Sharma, N. Ahmad, L. Teotia y B. Kumar, «"Quadcopter flight dynamics",» *International Journal for Research in Applied Science & Engineering,* vol. 5, nº 5, p. 5, 2017.

[26] S. Mandal, S. Kumar, K. Shaw, A. Kabiraj, V. Seth y P. Singh, «"Low-Cost bluetooth-arduino hover control design of a quad Copter",» *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),* vol. 11, nº 4, p. 11, 2016.

[27] E. Pillco, *"Sistema de control de actitud de un UAV tipo quadrotor usando el FPGA", Tesis de titulación, Universidad Tecnológica del Perú,* Lima, 2017.

[28] E. Abbasi y M. Mahjoob, «"Controlling of quadrotor UAV using a fuzzy system for tuning the PID gains in hovering mode",» *University of Tehran,* p. 6, 2010.

[29] A. Visioli, Practical PID control, Brescia, Italia: Springer, 2006.

[30] W. Durfee, *"Arduino Microcontroller Guide", Universidad de Minesota,* Minesota, 2011.

[31] B. Pérez, *"Comenzando con Arduino", Universidad de Cadiz,* Cadiz.

[32] E. Crespo, «"Aprendiendo Arduino",» Arduino Open-Source Community, 18 Junio 2017. [En línea]. Available: https://aprendiendoarduino.wordpress.com/2017/06/18/ide-arduino-y-configuracion/. [Último acceso: 07 Mayo 2019].

[33] Arduino, «Arduino,» Arduino, 07 Septiembre 2015. [En línea]. Available: https://www.arduino.cc/en/Guide/Environment#. [Último acceso: 07 Mayo 2019].

[34] E. Ramos y C. Castro, Arduino and Kinect Projects: Design, Build Blow Their Minds, New York: Apress, 2012.

[35] F. Báguena, *"Diseño y control de un cuadricóptero controlado por bluetooth vía android app", Universidad Politécnica de Valencia,* Valencia, 2016.

[36] Naylamp Mechatronics, «Naylamp Mechatronics,» Naylamp Mechatronics, [En línea]. Available: https://naylampmechatronics.com. [Último acceso: 3 Junio 2019].

[37] Electropro, «Electropro,» Electropro, 2017. [En línea]. Available: www.electropro.pe. [Último acceso: 3 junio 2019].

[38] F. Martínez, «Openwebinars,» Openwebinars, 5 Febrero 2015. [En línea]. Available: https://openwebinars.net. [Último acceso: 13 Junio 2019].

[39] Robologs, «Robologs,» Robologs, 15 Octubre 2014. [En línea]. Available: https://robologs.net. [Último acceso: 13 Junio 2019].

[40] Electronoobs, «Electronoobs,» Electronoobs, 29 Mayo 2017. [En línea]. Available: http://www.electronoobs.com/index.php. [Último acceso: 11 Setiembre 2019].

ANEXOS

FICHA DEL TRABAJO DE INVESTIGACIÓN

FACULTAD: _____

CARRERA: _____

1. **Título del Trabajo de Investigación propuesto**

   _____

2. **Indica la o las competencias del modelo del egresado que serán desarrolladas fundamentalmente con este Trabajo de Investigación:**

   _____

   _____

3. **Número de alumnos a participar en este trabajo**. (máximo 2)
   Número de alumnos: _____

4. **Indica si el trabajo tiene perspectivas de continuidad, después de obtenerse el Grado Académico d Bachiller, para seguirlo desarrollando para la titulación por la modalidad de Tesis o no.**

   _____

5. **Enuncia 4 o 5 palabras claves que le permitan realizar la búsqueda de información para el Trabajo en Revistas Indizadas en WOS, SCOPUS, EBSCO, SciELO, etc., desde el comienzo del curso y obtener así información de otras fuentes especializadas.**
   Ejemplo:

Palabras Claves	REPOSITORIO 1	REPOSITORIO 2	REPOSITORIO 3
1.-			
2.-			
3.-			
4.-			
5.-			

6. **Como futuro asesor de investigación para titulación colocar:**
   (Indique sus datos personales)
   a. **Nombre:** _____
   b. **Código docente:** _____
   c. **Correo institucional:** _____
   d. **Teléfono:** _____

7. **Especifica si el Trabajo de Investigación:**
   (Marca con un círculo la que corresponde, puede ser más de una)

   a. **Contribuye a un trabajo de investigación de una Maestría o un doctorado de algún profesor de la UTP.**
   b. **Está dirigido a resolver algún problema o necesidad propia de la organización.**
   c. **Forma parte de un contrato de servicio a terceros.**

d. **Corresponde a otro tipo de necesidad o causa (explicar el detalle):**

_____

_____

_____

8. **Explica de forma clara y comprensible los objetivos o propósitos del trabajo de investigación**

_____

_____

_____

_____

9. **Brinde una primera estructuración de las acciones específicas que debe realizar el alumno para que le permita iniciar organizadamente su trabajo**

_____

_____

_____

_____

10. **Incorpora todas las observaciones y recomendaciones que consideres de utilidad para el alumno y a los profesores del curso con el fin de que desarrollen con éxito todas las actividades**

_____

_____

_____

_____

_____

11. **Fecha y docente que propone la tarea de investigación**

Fecha de elaboración de ficha (día/mes/año): _____/ _____/ _____

Docente que propone la tarea de investigación: _____

12. **Esta Ficha de Tarea de Investigación ha sido aprobada como Tarea de Investigación para el Grado de Bachiller en esta carrera por:**

(Sólo para ser llenada por la Facultad)

Nombre: _____

Código: _____

Cargo: _____

Fecha de aprobación de ficha (día/mes/año): _____/ _____/ _____

Anexo 2: Glosario

Actuador: Componente eléctrico/electrónico capaz de convertir la energía eléctrica o señales eléctricas en movimiento mecánico.

Aviónica/Aviónica: Conjunto de funcionalidades combinadas propias de la aviación y la electrónica.

Autopiloto/Autopilot: Dispositivo electrónico que permite a los vehículos ejecutar misiones de manera autónoma, sin la intervención de un operador humano.

Bluetooth: Modo de conexión inalámbrica de dos o más dispositivos electrónicos mediante señales de radiofrecuencia.

Brushless: Motor eléctrico sin escobillas.

Código Abierto: Propiedad del dispositivo que permite al usuario acceder a los algoritmos de control en el microprocesador.

Cuadrotor/Cuadricóptero: Vehículo aéreo de despegue vertical compuesto de cuatro hélices.

Drone: Nombre coloquial con que se le denomina a los UAV, especialmente multirotores.

Envergadura: Distancia de punta a punta del ala de un avión.

Empenaje: Conjunto formado entre el estabilizador vertical y horizontal de un avión.

Fuerza de sustentación: Fuerza aerodinámica en sentido opuesto a la gravedad, producida por las hélices o alas. No confundir con la fuerza de empuje, en sentido de vuelo del avión.

Hover: Vuelo estático a una altura determinada, principalmente descrita por aeronaves de despegue vertical (helicópteros y multirrotores).

Hardware: Conjunto de elementos que conforman un sistema.

Hexacóptero: Vehículo aéreo de despegue vertical compuesto de seis hélices.

Microcontrolador: Conjunto de componentes electrónicos que ejecutan ordenes mediante un código implementado en su memoria.

Multirotor/Multicóptero: Vehículo aéreo con dos o más hélices.

Plataforma electrónica Arduino: Dispositivo electrónico compuesto de un microprocesador capaz de ejecutar procesos mediante códigos de programación.

PID: Algoritmo matemático de control compuesto de una parte proporcional, integral y derivativa.

PID-difuso: Combinación de algoritmos matemáticos de control PID y lógica difusa

Software: Conjunto de algoritmos que forman parte de un dispositivo computarizado.

Sistema lazo abierto: Proceso lineal compuesto por una entrada y una salida.

Sistema lazo cerrado: Proceso no lineal compuesto de una entrada, una salida y un algoritmo de realimentación.

Vuelo Crucero: Etapa de vuelo horizontal rectilíneo de un avión, donde se consigue una mayor distancia recorrida a un menor consumo de combustible.

Anexo 3: Acrónimos

EASA: European Aviation Safety Agency

ESC: Electronic Speed Controller

FAA: Federal Aviation Administration

IDE: Integrated Development Environment

IMU: Inetial Measurement Unit

PID: Proporcional Integral y Derivativa

PWM: Pulse-Width Modulation

PCB: Printed Circuit Board

SAA: Sistema Aéreo Autónomo

UAS: Unmanned Aerial System

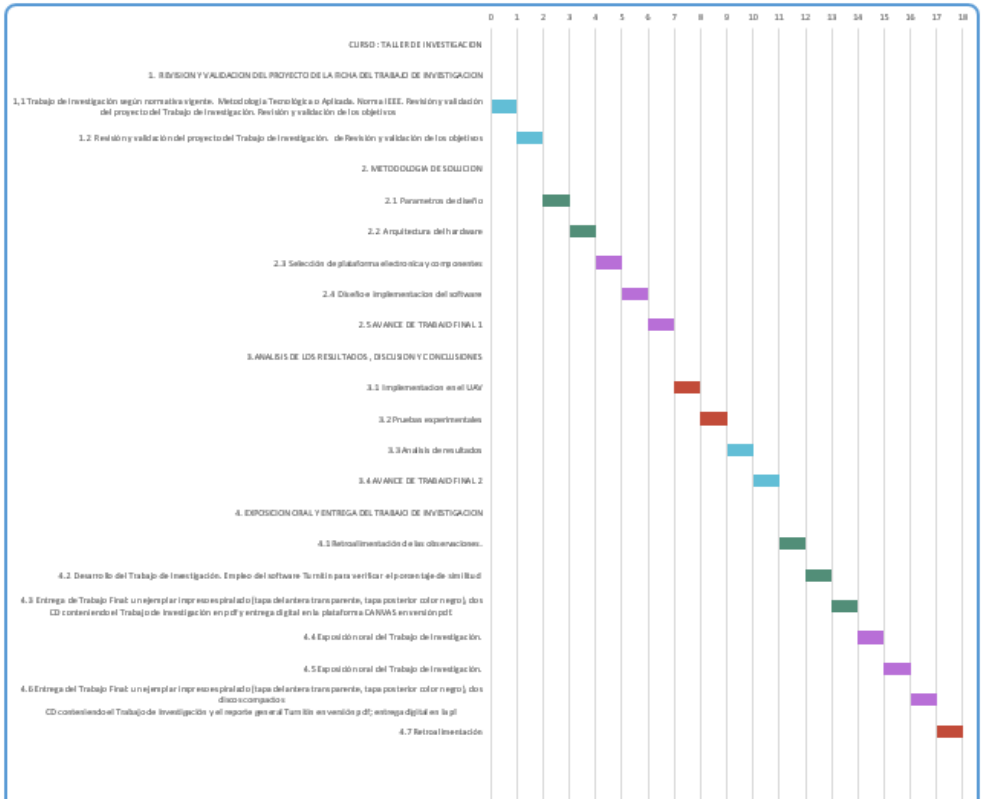
UAV: Unmanned Aerial Vehicle

VANT: Vehiculo Aereo No Tripulado

Anexo 4: Diagrama de Gantt

TAREA	INICIO	FIN	DURACION EN SEMANAS
CURSO : TALLER DE INVESTIGACION			
1. REVISION Y VALIDACION DEL PROYECTO DE LA FICHA DEL TRABAJO DE INVESTIGAC			
1,1 Trabajo de Investigación según normativa vigente. Metodología Tecnológica o Aplicada. Norma IEEE. Revisión y validación del proyecto del Trabajo de Investigación. Revisión y validación de los objetivos	0	1	1
1.2 Revisión y validación del proyecto del Trabajo de Investigación. de Revisión y validación de los objetivos	1	2	1
2. METODOLOGIA DE SOLUCION			
2.1 Parametros de diseño	2	3	1
2.2 Arquitectura del hardware	3	4	1
2.3 Selección de plataforma electronica y componentes	4	5	1
2.4 Diseño e implementacion del software	5	6	1
2.5 AVANCE DE TRABAJO FINAL 1	6	7	1
3.ANALISIS DE LOS RESULTADOS , DISCUSION Y CONCLUSIONES			
3.1 Implementacion en el UAV	7	8	1
3.2 Pruebas experimentales	8	9	1
3.3 Analisis de resultados	9	10	1
3.4 AVANCE DE TRABAJO FINAL 2	10	11	1
4. EXPOSICION ORAL Y ENTREGA DEL TRABAJO DE INVESTIGACION			
4.1 Retroalimentación de las observaciones.	11	12	1
4.2 Desarrollo del Trabajo de Investigación. Empleo del software Turnitin para verificar el porcentaje de similitud	12	13	1
4.3 Entrega de Trabajo Final: un ejemplar impreso espiralado (tapa delantera transparente, tapa posterior color negro), dos CD conteniendo el Trabajo de Investigación en pdf y entrega digital en la plataforma CANVAS en versión pdf.	13	14	1
4.4 Exposición oral del Trabajo de Investigación.	14	15	1
4.5 Exposición oral del Trabajo de Investigación.	15	16	1
4.6 Entrega del Trabajo Final: un ejemplar impreso espiralado (tapa delantera transparente, tapa posterior color negro), dos discos compactos CD conteniendo el Trabajo de Investigación y el reporte general Turnitin en versión pdf; entrega digital en la plataforma CANVAS en	16	17	1
4.7 Retroalimentación	17	18	1

Start Date in Number For: 0.00 *Use this number for the Minimum Bound of the Horizontal Axis to set the beginning of the chart*



Anexo 5: Código de programación

```
#include <Wire.h>
#include <Servo.h>
Servo right;
Servo left;
Servo vtail;

int16_t Acc_rawX, Acc_rawY, Acc_rawZ, Gyr_rawX, Gyr_rawY, Gyr_rawZ;

float Acceleration_angle[2];
float Gyro_angle[2];
float Total_angle[2];
float elapsedTime, time, timePrev;
int i;
float rad_to_deg=180/3.141592654;

float PID1, PID2, aileron1, aileron2, VT, error1, error2, previous_error;
float pid_p1=0;
float pid_i1=0;
float pid_d1=0;
float pid_p2=0;
float pid_i2=0;
float pid_d2=0;

//////////PID CONSTANTS/////////
double kp=100;
double ki=102;
double kd=103;
///////////////////////////////

int pos=90;
float desired_angle=0;

void setup() {
  Wire.begin();
  Wire.beginTransmission(0x68);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(250000);
  right.attach(5);
  left.attach(6);
  vtail.attach(7);
```

```
  time=millis();
 right.write(pos);
 left.write(pos);
 vtail.write(pos);
 delay(1000);

}

void loop() {
  ////////////////I M U/////////////////
  timePrev=time;
  time=millis();
  elapsedTime=(time-timePrev)/1000;

  Wire.beginTransmission(0x68);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(0x68,6,true);

  Acc_rawX=Wire.read()<<8|Wire.read();
  Acc_rawY=Wire.read()<<8|Wire.read();
  Acc_rawZ=Wire.read()<<8|Wire.read();

  /*--X--*/

Acceleration_angle[0]=atan((Acc_rawY/16384.0)/sqrt(pow((Acc_rawX/16384.0),2)+pow((
Acc_rawZ/16384.0),2)))*rad_to_deg;

  /*--Y--*/

  Acceleration_angle[1]=atan(-
1*(Acc_rawX/16384.0)/sqrt(pow((Acc_rawY/16384.0),2)+pow((Acc_rawZ/16384.0),2)))*ra
d_to_deg;

  Wire.beginTransmission(0x68);
  Wire.write(0x43);
  Wire.endTransmission(false);
  Wire.requestFrom(0x68,4,true);

  Gyr_rawX=Wire.read()<<8|Wire.read();
  Gyr_rawY=Wire.read()<<8|Wire.read();
  /*--X--*/

  Gyro_angle[0]=Gyr_rawX/131.0;

  /*--Y--*/
```

```
  Gyro_angle[1]=Gyr_rawY/131.0;

  /*---X axis angle---*/

Total_angle[0]=0.98*(Total_angle[0]+Gyro_angle[0]*elapsedTime)+0.02*Acceleration_ang
le[0];

  /*---Y axis angle---*/

Total_angle[1]=0.98*(Total_angle[1]+Gyro_angle[1]*elapsedTime)+0.02*Acceleration_ang
le[1];

  /*////////////////PID alerones///////////////*/

  error1=Total_angle[0]-desired_angle;
  error2=Total_angle[1]-desired_angle;

  pid_p1=kp*error1;
  pid_p2=kp*error2;
  pid_i1=pid_i1+(ki*error1);
  pid_i2=pid_i2+(ki*error2);
  pid_d1=kd*((error1-previous_error)/elapsedTime);
  pid_d2=kd*((error2-previous_error)/elapsedTime);
  PID1=pid_p1+pid_i1+pid_d1;
  PID2=pid_p2+pid_i2+pid_d2;

  if(PID1<-90)
  {
    PID1=-90;
  }
  if(PID1>90)
  {
    PID1=90;
  }
  if(PID2<-90)
  {
    PID2=-90;
  }
  if(PID2>90)
  {
    PID2=90;
  }

  aileron1=pos+PID1;
  aileron2=pos-PID1;
  VT=pos+PID2;
```

```
  //right

int Servoright=map(aileron1, 0, 180, 0, 180);

right.write(Servoright);

 //left

 int Servoleft=map(aileron2, 0, 180, 0, 180);

 left.write(Servoleft);

 //vtail

 int Servovtail=map(VT, 0, 180, 0, 180);

 vtail.write(Servovtail);

  Serial.print("Pitch: "); Serial.print(-Total_angle[1]);
  Serial.print("\t");
  Serial.print("Roll: "); Serial.print(Total_angle[0]);
  Serial.print("\n");

}
```

Anexo 6: Presupuesto

Tipo de Recurso		Recurso	Unidades	Costo unitario	Costo parcial
Humano					
				Sub total	**S/. 0**
Material	Electrónica	Arduino UNO R3	1	S/. 30.00	S/. 30.00
		Servomotor SG90S	2	S/. 8.50	S/. 17.00
		Módulo MPU 6050	1	S/. 14.50	S/. 14.50
		Pack de cables macho-macho	1	S/.6.50	S/. 6.50
		Fuente Arduino UNO	1	S/. 13.00	S/. 13.00
		Pack de cables macho-hembra	1	S/. 6.50	S/. 6.50
		Placa multiuso	1	S/. 1.00	S/. 1.00
		Acido férrico de 60 ml	1	S/. 1.50	S/. 1.50
		Vaquela 10x10	1	S/. 1.00	S/. 1.00
				Sub total	**S/. 110.80**
Otros		Impresiones		S/. 35.00	S/. 35.00
				Sub total	**S/. 35.00**
				Total	**S/. 145.80**

Turnitin Informe de Originalidad

- Procesado el: 06-nov.-2019 11:00 a. m. -05
- Identificador: 1208324826
- Número de palabras: 14936
- Entregado: 1

Taller de investigación II Por Pedro QUISPE CURY

Índice de similitud

12%

Similitud según fuente

Internet Sources:

8%

Publicaciones:

3%

Trabajos del estudiante:

9%

1% match (trabajos de los estudiantes desde 05-dic.-2017)

Submitted to Indian Institute of Technology-Bhubaneswar on 2017-12-05

1% match (Internet desde 12-ago.-2018)

http://www.giuseppecaccavale.it/arduino/mpu-6050-gy-521-arduino-tutorial/

1% match (trabajos de los estudiantes desde 24-jun.-2019)

Submitted to Universidad Tecnologica del Peru on 2019-06-24

< 1% match (trabajos de los estudiantes desde 12-jun.-2018)

Submitted to Monash University on 2018-06-12

< 1% match (Internet desde 12-sept.-2019)

https://docplayer.info/136061940-Perancangan-digital-wireless-remote-stick-commander-untuk-pengendali-camera-crane-dan-pan-tilt-head-berbasis-sensor-accelerogyro.html

< 1% match (trabajos de los estudiantes desde 10-abr.-2013)

Submitted to UC, Boulder on 2013-04-10

< 1% match (Internet desde 19-sept.-2019)

http://repositorio.unicamp.br/jspui/bitstream/REPOSIP/331876/1/Netto_PedroGattiArtaxo_M.pdf

< 1% match (Internet desde 18-jun.-2019)

https://issuu.com/ric_noc/docs/memoria_tricop

< 1% match (trabajos de los estudiantes desde 04-dic.-2018)

Submitted to Tecsup on 2018-12-04

< 1% match (trabajos de los estudiantes desde 21-mar.-2017)

Submitted to Universidad Ricardo Palma on 2017-03-21

< 1% match (Internet desde 17-jun.-2019)

http://dspace.espoch.edu.ec/handle/123456789/6065

< 1% match (Internet desde 28-jul.-2016)

http://tdrobotica.co/categoria/113

< 1% match (Internet desde 02-may.-2019)

http://aqp-unsa.blogspot.com/2016/07/tesis-seleccionadas-2016-que-seran.html

< 1% match (trabajos de los estudiantes desde 10-ene.-2017)

Submitted to Escuela Politecnica Nacional on 2017-01-10

< 1% match (trabajos de los estudiantes desde 12-jun.-2019)

Submitted to Universidad Carlos III de Madrid on 2019-06-12

< 1% match (trabajos de los estudiantes desde 21-mar.-2019)

Submitted to Universidad Nacional de Colombia on 2019-03-21

< 1% match (publicaciones)

Ajay K Joseph, Amar Bousbaine, Abdelkader Fareha. "A Wireless communication system for a quadrotor helicopter", 2018 53rd International Universities Power Engineering Conference (UPEC), 2018

< 1% match (trabajos de los estudiantes desde 18-feb.-2013)

Submitted to Universidad Tecnologica de Honduras on 2013-02-18

< 1% match (Internet desde 19-ago.-2017)

https://www.conftool.com/iced17/index.php/ICED2017_321_a.pdf?filename=ICED2017_321_a.pdf&form_id=321&form_version=final&page=downloadPaper

< 1% match (Internet desde 02-abr.-2019)

http://aircconline.com/ijctcm/V7N2/7217ijctcm01.pdf

< 1% match (trabajos de los estudiantes desde 07-jun.-2019)

Submitted to University of Lancaster on 2019-06-07

< 1% match (trabajos de los estudiantes desde 23-nov.-2018)

Submitted to John F Kennedy, The American School of Queretaro on 2018-11-23

< 1% match (trabajos de los estudiantes desde 30-abr.-2018)

Submitted to University of Bath on 2018-04-30

< 1% match (Internet desde 29-sept.-2018)

http://www.electronoobs.com/eng_robotica_tut6_2.php

< 1% match (Internet desde 08-feb.-2018)

http://arduinom.org/uzaktan-2xservo-kontrolu/

< 1% match (Internet desde 04-sept.-2019)

https://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/

< 1% match (Internet desde 02-dic.-2018)

http://www.dspace.espol.edu.ec/xmlui/bitstream/handle/123456789/40609/D-84750.pdf?sequence=-

< 1% match (publicaciones)

Michal Podhradsky, Jarret Bone, Calvin Coopmans, Austin Jensen. "Battery model-based thrust controller for a small, low cost multirotor Unmanned Aerial Vehicles", 2013 International Conference on Unmanned Aircraft Systems (ICUAS), 2013

< 1% match (trabajos de los estudiantes desde 05-ene.-2018)

Submitted to TAR University College on 2018-01-05

< 1% match (trabajos de los estudiantes desde 12-abr.-2016)

Submitted to Escuela Politecnica Nacional on 2016-04-12

< 1% match (trabajos de los estudiantes desde 06-ago.-2015)

Submitted to MCI Management Centre Innsbruck on 2015-08-06

< 1% match (trabajos de los estudiantes desde 17-dic.-2017)

Submitted to Curtin University of Technology on 2017-12-17

< 1% match (Internet desde 15-oct.-2016)

http://produccioncientificaluz.org/index.php/cuestiones/search/authors/view?affiliation=Universidad+Sim%C3%B3n+Bol%C3%ADvar-Venezuela&country=VE&firstName=Makram&lastName=Haluani&middleName=

< 1% match (trabajos de los estudiantes desde 22-feb.-2018)

Submitted to Universidad Pontificia Bolivariana on 2018-02-22

< 1% match (trabajos de los estudiantes desde 09-jul.-2019)

Submitted to Universitat Politècnica de València on 2019-07-09

< 1% match (Internet desde 06-dic.-2018)

https://www.ntsb.gov/investigations/AccidentReports/Reports/AAR1006.pdf

< 1% match (Internet desde 28-feb.-2019)

https://www.ijedr.org/papers/IJEDR1603052.pdf

< 1% match (trabajos de los estudiantes desde 26-jun.-2019)

Submitted to Universidad de Burgos UBUCEV on 2019-06-26

< 1% match (Internet desde 26-nov.-2015)

http://www.amazon.co.uk/Diseno-Un-Sistema-Control-Gestion/dp/384656849X

< 1% match (Internet desde 21-abr.-2019)

http://cjece.ubm.ro/vol/10-2017/n2/1710.11-10203.pdf

< 1% match (Internet desde 26-nov.-2003)

http://www.museocalzado.com/modules.php?name=News&file=article&sid=42

< 1% match (trabajos de los estudiantes desde 10-jul.-2014)

Submitted to Pontificia Universidad Catolica del Peru on 2014-07-10

< 1% match (trabajos de los estudiantes desde 26-ago.-2016)

Submitted to Escuela Politecnica Nacional on 2016-08-26

< 1% match (Internet desde 24-may.-2016)

https://dspace.lib.cranfield.ac.uk/bitstream/1826/6775/1/Sunan_chumalee_Thesis_2010.pdf

< 1% match (Internet desde 06-ene.-2014)

http://www.educaweb.mx/centros/medio-ambiente-zoologia-veterinaria/centros-formacion-posgrados/

< 1% match (Internet desde 19-mar.-2008)

http://www.imem.unavarra.es/mecanica-2ii/download/ProEx2003-09-10.pdf

< 1% match (trabajos de los estudiantes desde 09-oct.-2018)

Submitted to Universidad Politecnica Salesiana del Ecuado on 2018-10-09

< 1% match (trabajos de los estudiantes desde 02-abr.-2018)

Submitted to Universidad Tecnologica del Peru on 2018-04-02

< 1% match (trabajos de los estudiantes desde 15-jul.-2015)

Submitted to Universidad de Vigo on 2015-07-15

< 1% match (trabajos de los estudiantes desde 18-jul.-2016)

Submitted to Escuela Politecnica Nacional on 2016-07-18

< 1% match (Internet desde 10-sept.-2018)

http://www.aprendiendoarduino.com/author/aprendiendoarduino/page/2/

< 1% match (publicaciones)

Ahmed Elsaadany, Yi Wen-jun. "Accuracy Improvement Capability of Advanced Projectile Based on Course Correction Fuze Concept", The Scientific World Journal, 2014

< 1% match (trabajos de los estudiantes desde 11-ene.-2016)

Submitted to University of Surrey on 2016-01-11

< 1% match (trabajos de los estudiantes desde 21-may.-2014)

Submitted to Escuela Politecnica Nacional on 2014-05-21

< 1% match (Internet desde 18-jul.-2019)

https://www.bts.gov/content/number-us-airportsa

< 1% match (Internet desde 25-oct.-2019)

https://arc.aiaa.org/doi/10.2514/1.J058002

< 1% match (Internet desde 29-oct.-2015)

http://oa.upm.es/cgi/exportview/institution/Telecomunicacion/EndNote/Telecomunicacion.enw

< 1% match (Internet desde 28-jul.-2016)

http://www.secinfo.com/dRY7g.uAe.htm

< 1% match (Internet desde 23-dic.-2006)

http://www.proteccio-civil.net/inclu.php?op=vdm009&newlang=spa

< 1% match ()

http://www.rlc.fao.org/mujer/docs/practica/Ethio-es.PDF

< 1% match (trabajos de los estudiantes desde 27-ago.-2015)

Submitted to University of Glasgow on 2015-08-27

< 1% match (trabajos de los estudiantes desde 09-jun.-2015)

Submitted to 97488 on 2015-06-09

< 1% match (trabajos de los estudiantes desde 16-jul.-2019)

Submitted to Universitat Politècnica de València on 2019-07-16

< 1% match (trabajos de los estudiantes desde 20-oct.-2015)

Submitted to Universidad Carlos III de Madrid on 2015-10-20

< 1% match (Internet desde 19-sept.-2017)

http://academica-e.unavarra.es/bitstream/handle/2454/24731/TFG_vfinal_completo.pdf?isAllowed=y&sequence=1

< 1% match (Internet desde 29-nov.-2016)

https://prezi.com/jas0dpwgvd43/mundo-arduino/

< 1% match (Internet desde 02-jun.-2017)

http://repositorio.ute.edu.ec/xmlui/handle/123456789/8624?locale-attribute=en

< 1% match (Internet desde 21-mar.-2016)

http://web.oie.int/boutique/extrait/21alleweldt619630.pdf

< 1% match (Internet desde 15-oct.-2013)

http://helvia.uco.es/xmlui/bitstream/handle/10396/2070/9788478019427.pdf.txt?sequence=3

< 1% match (Internet desde 10-jul.-2010)

http://www.slideshare.net/jveizaga/arquitectura-computadora

< 1% match (Internet desde 24-oct.-2002)

http://www.funredes.org/espanol/institucion/institucion.php3/docid/300

< 1% match (Internet desde 18-nov.-2003)

http://www.apoyopositivo.org/preguntas.html

< 1% match ()

http://www.terra.cl/noticias/noticias.cfm?id_cat=704&id_reg=239981

< 1% match (trabajos de los estudiantes desde 17-nov.-2014)

Submitted to KTH - The Royal Institute of Technology on 2014-11-17

< 1% match (Internet desde 23-jun.-2019)

http://dspace.espoch.edu.ec/bitstream/123456789/6065/1/20T00776.pdf

< 1% match (Internet desde 16-jun.-2017)

http://www.dtic.mil/dtic/tr/fulltext/u2/a545756.pdf

< 1% match (Internet desde 13-jun.-2017)

http://repositorio.ute.edu.ec/bitstream/123456789/10303/1/45065_1.pdf

< 1% match (Internet desde 24-ago.-2016)

http://www.dts.edu/dmin/dissertations/

< 1% match (Internet desde 25-jul.-2016)

https://www.scribd.com/doc/286702915/Patrones-Java

< 1% match (Internet desde 25-may.-2014)

http://archivesblogs.com/page/68

< 1% match (Internet desde 14-feb.-2007)

http://www.saa.unito.it/pdf/alfa/lobos1.pdf

< 1% match (Internet desde 07-nov.-2006)

http://chirigotadelvera.webcindario.com/historialn.htm

< 1% match (Internet desde 30-jun.-2003)

http://www.lcc.uma.es/docencia/ETSIInf/pl/ejercicios/pl2/sene96_3.html

< 1% match (publicaciones)

Antonio E. Jimenez-Cano, Pedro J. Sanchez-Cuevas, Pedro Grau, Anibal Ollero, Guillermo Heredia. "Contact-Based Bridge Inspection Multirotors: Design, Modeling, and Control Considering the Ceiling Effect", IEEE Robotics and Automation Letters, 2019

< 1% match (trabajos de los estudiantes desde 06-ene.-2017)

Submitted to CONACYT on 2017-01-06

< 1% match (trabajos de los estudiantes desde 24-oct.-2017)

Submitted to Universidad Carlos III de Madrid on 2017-10-24

< 1% match (trabajos de los estudiantes desde 27-oct.-2015)

Submitted to Patricia Test Account on 2015-10-27

< 1% match (trabajos de los estudiantes desde 10-feb.-2016)

Submitted to Pontificia Universidad Catolica del Peru on 2016-02-10

< 1% match (trabajos de los estudiantes desde 12-jul.-2017)

Submitted to Universidad Politécnica de Madrid on 2017-07-12

< 1% match (trabajos de los estudiantes desde 06-ago.-2019)

Submitted to BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA BIBLIOTECA on 2019-08-06

< 1% match (trabajos de los estudiantes desde 20-sept.-2019)

Submitted to Universidad Carlos III de Madrid on 2019-09-20

< 1% match (trabajos de los estudiantes desde 31-dic.-2013)

Submitted to Universidad de Valladolid on 2013-12-31

< 1% match (trabajos de los estudiantes desde 02-sept.-2018)

Submitted to The University of Manchester on 2018-09-02

< 1% match (trabajos de los estudiantes desde 23-jun.-2019)

Submitted to Universidad Tecnologica del Peru on 2019-06-23

< 1% match (trabajos de los estudiantes desde 14-jun.-2017)

Submitted to Unidades Tecnológicas de Santander on 2017-06-14

< 1% match (trabajos de los estudiantes desde 29-sept.-2017)

Submitted to Politécnico Colombiano Jaime Isaza Cadavid on 2017-09-29

< 1% match (trabajos de los estudiantes desde 23-jun.-2019)

Submitted to Universidad Tecnologica del Peru on 2019-06-23

< 1% match (trabajos de los estudiantes desde 08-may.-2018)

Submitted to Universidad Politecnica Salesiana del Ecuado on 2018-05-08

< 1% match (trabajos de los estudiantes desde 09-nov.-2017)

Submitted to Escuela Superior Politécnica del Litoral on 2017-11-09

< 1% match (trabajos de los estudiantes desde 24-abr.-2018)

Submitted to Escuela Politecnica Nacional on 2018-04-24