



Implementación de una maleta electrónica de vuelo para visualizar un sistema FMS aeronave prototipo Skiron

Fecha de recibido: 27 de octubre 2022	Fecha de aprobado: 30 de abril 2023
Reception date: October 27, 2022	Approval date: April 30, 2023
Data de recebimento: 27 de outubro de 2022	Data de aprovação: 30 de abril de 2023

Yeison Manuel Montoya Lemus

<https://orcid.org/0000-0002-3068-2335>
 ymontoya@usbog.edu.co

Magíster en Ingeniería
 Docente e investigador – Universidad de San Buenaventura, Colombia
 Rol del investigador: teórico y escritura
 Grupo de Investigación AeroTech

Master in Engineering
 Professor and researcher – Universidad de San Buenaventura, Colombia
 Researcher's role: theoretical and writing
 AeroTech Research Group

Mestre em Engenharia
 Professor e pesquisador – Universidad de San Buenaventura, Colombia
 Papel do pesquisador: teórico e escritor
 Grupo de Pesquisa AeroTech

José Daniel Gómez Romero

<https://orcid.org/0000-0001-6528-3951>
 jdgomezr@academia.usbbog.edu.co

Estudiante de Ingeniería Aeronáutica
 Universidad de San Buenaventura, Colombia
 Rol del investigador: teórico y escritura

Aeronautical Engineering Student
 San Buenaventura University, Colombia
 Researcher's role: theoretical and writing

Estudante de Engenharia Aeronáutica
 Universidade San Buenaventura, Colombia
 Papel do pesquisador: teórico e escrito

Mateo Campos Casallas

<https://orcid.org/0000-0001-9587-770X>
 mcamposc@academia.usbbog.edu.co

Estudiante de Ingeniería Aeronáutica
 Universidad de San Buenaventura, Colombia
 Rol del investigador: teórico y escritura

Aeronautical Engineering Student
 San Buenaventura University, Colombia
 Researcher's role: theoretical and writing

Estudante de Engenharia Aeronáutica
 Universidade San Buenaventura, Colombia
 Papel do pesquisador: teórico e escrito

Edgar José Martínez Montoya

<https://orcid.org/0000-0001-7635-3599>
 ejmartinezm@academia.usbbog.edu.co

Estudiante de Ingeniería Aeronáutica
 Universidad de San Buenaventura, Colombia
 Rol del investigador: teórico y escritura

Aeronautical Engineering Student
 San Buenaventura University, Colombia
 Researcher's role: theoretical and writing

Estudante de Engenharia Aeronáutica
 Universidade San Buenaventura, Colombia
 Papel do pesquisador: teórico e escrito

Cómo citar este artículo: Montoya Lemus, Y. M., Gómez Romero, J. D., Campos Casallas, M. y Martínez Montoya, E. J. (2023). Implementación de una maleta electrónica de vuelo para visualizar un sistema FMS aeronave prototipo Skiron. *Ciencia y Poder Aéreo*, 18(2), 47-66. <https://doi.org/10.18667/cienciaypoderareero.768>



Implementación de una maleta electrónica de vuelo para visualizar un sistema FMS aeronave prototipo Skiron

Resumen: Este artículo presenta una aplicación práctica de la construcción y simulación de una interfaz de visualización para un monitor de datos de vuelo. El ejercicio tiene un enfoque analítico-sintético y se ha llevado a cabo en la aeronave prototipo Skiron-usb, diseñada por la Universidad de San Buenaventura, con el fin de complementar la taxonomía de los instrumentos de aviónica en cabina mediante la implementación de maletas electrónicas de vuelo. La inclusión del sistema se ha logrado a partir de la determinación de los parámetros tecnológicos necesarios para su integración en el futuro. Para ello, se ha definido el sistema de datos de vuelo en función de las condiciones y misiones de la aeronave, estableciendo la base de intercomunicación a través de un arreglo de circuitos entre una placa Arduino UNO y la plataforma Stewart, controlados por un *joystick*. En función de la transmisión de datos y su lectura, se ha propuesto la herramienta visualizadora adecuada. Como resultado del estudio académico, se han identificado los parámetros de actitud en relación con la fase de vuelo desarrollada en el vehículo aéreo Skiron, un modelo Conmute multipropósito según especificaciones de diseño (FAR 23, categoría transporte STOL).

Palabras clave: aeronave; Arduino; aviónica; FMS; EFB; plataforma Stewart.

Implementation of an electronic flight case to visualize an FMS system in a Skiron prototype aircraft

Abstract: This paper presents a practical application of the construction and simulation of a display interface for a flight data monitor. The exercise has an analytical-synthetic approach and has been carried out on the Skiron-usb prototype aircraft, designed by the University of San Buenaventura, in order to complement the taxonomy of avionics instruments in the cockpit through the implementation of electronic flight suitcases. The inclusion of the system has been achieved by determining the technological parameters necessary for its integration in the future. To this end, the flight data system has been defined according to the conditions and missions of the aircraft, establishing the basis of intercommunication through an arrangement of circuits between an Arduino UNO board and the Stewart platform, controlled by a joystick. Depending on the data transmission and readout, the appropriate display tool has been proposed. As a result of the academic study, the attitude parameters have been identified in relation to the flight phase developed in the Skiron air vehicle, a Conmute multipurpose model according to design specifications (FAR 23, transport category STOL).

Keywords: Aircraft; arduino; avionics; FMS; EFB; Stewart platform.

Implementação de um caso de voo eletrônico para visualizar um sistema FMS num protótipo de aeronave Skiron

Resumo: Este artigo apresenta uma aplicação prática da construção e simulação de uma interface de exibição para um monitor de dados de voo. O exercício tem uma abordagem analítico-sintética e foi realizado na aeronave protótipo Skiron-usb, projetada pela Universidade de San Buenaventura, a fim de complementar a taxonomia dos instrumentos aviônicos na cabine de comando por meio da implementação de malas eletrônicas de voo. A inclusão do sistema foi realizada por meio da determinação dos parâmetros tecnológicos necessários para sua futura integração. Para isso, o sistema de dados de voo foi definido de acordo com as condições e missões da aeronave, estabelecendo a base para a intercomunicação por meio de um arranjo de circuitos entre uma placa Arduino UNO e a plataforma Stewart, controlada por um *joystick*. Dependendo da transmissão e da leitura dos dados, foi proposta a ferramenta de exibição adequada. Como resultado do estudo acadêmico, os parâmetros de atitude foram identificados em relação à fase de voo desenvolvida no veículo aéreo Skiron, um modelo multiuso Conmute de acordo com as especificações do projeto (FAR 23, categoria de transporte STOL).

Palavras-chave: Aeronave; Arduino; aviônica; FMS; EFB; plataforma Stewart.

Introducción

El Departamento de Transporte de Estados Unidos establece que la incorporación de sistemas FDM-EFB en la cabina cumple numerosas misiones de aviónica y da lugar a la innovación tecnológica (Federal Aviation Administration [FAA], 2022). Además, estos elementos notifican posibles acciones de riesgo que puedan manifestarse durante un vuelo, garantizando la efectividad de la gestión de seguridad operacional (Fuerza Aérea Colombiana [FAC], 2018). El informe de servicios asignado al aviador debe formalizarse utilizando un *software* que interprete y ejecute trabajos precisos, operado a través de sucesiones lógicas, lo cual es posible gracias a un dispositivo denominado Electronic Flight Bag (EFB). La cubierta que envuelve este artefacto se conoce como Flight Data Monitoring (FDM).

En este orden de ideas, los parámetros de tecnología establecidos y la documentación pertinente a la integración del equipo a bordo de la aeronave Skiron proporcionarán la respectiva visualización y desarrollo del sistema FDM-EFB mediante una interfaz virtual. El tipo de metodología a utilizar es de carácter analítico-sintético: se descompone el tema en fragmentos reducidos que logran identificar errores e inconvenientes presentados (análisis); y a partir de cada fragmento integrado, se construye una visión global sólida del proyecto para dar solución a la problemática planteada (síntesis).

En primer lugar, se deben representar las condiciones y misiones que tendrá la aeronave Skiron, de modo que se pueda definir el sistema de datos de vuelo que almacene, guíe, adapte y transcriba señales analógicas, discretas y digitales. A partir de esto, se deberá designar la infraestructura de comunicación de la unidad de adquisición de datos, seleccionando el tipo de transmisión de datos que registre y almacene los datos de vuelo. Por consiguiente, se prevé proyectar las condiciones designadas del avión según la lectura de datos, por lo cual la herramienta tecnológica visualizadora deberá ser la más adecuada posible. Todo lo relacionado con esto debe llevarse a cabo mediante un proceso de ingeniería y sus etapas, como se muestra en la figura 1. La integración del sistema EFB en el esquema presentado está ubicada entre las redes conectadas por cables (Wireless Ground Link) y los servicios de navegación disponibles (Cellular Service Provider).

A grandes rasgos, se establecen tres objetivos específicos: delimitar el sistema de transmisión, identificar la web gráfica adecuada y especificar los procedimientos de conexión. Al delimitar el sistema de transmisión de datos, se presenta el desafío de identificar con precisión las necesidades de los pilotos, por lo que el *software* y el *hardware* deben ser adecuados en función de esto. En concordancia con la literatura existente sobre la aeronave prototipo Skiron, se abordan los alcances y las exigencias de cada configuración analizada.

Una vez definidas las características de cada perfil de misión, la selección del *software* y el *hardware* se

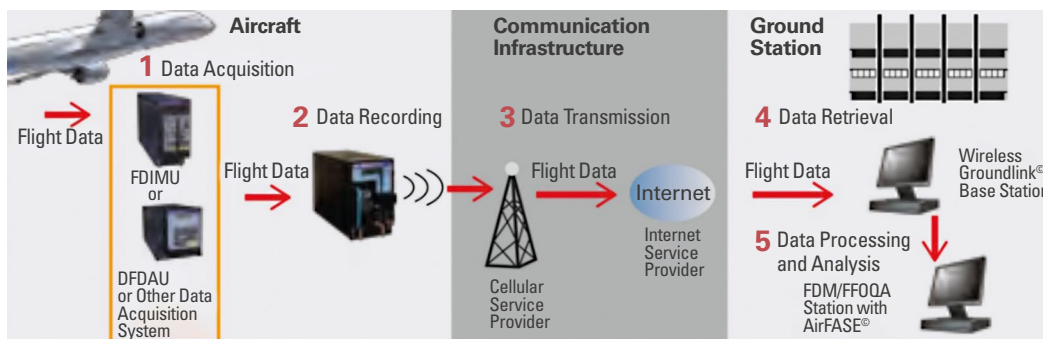


Figura 1. Fases de desarrollo para un sistema FDM
Fuente: tomado de Delhom (2014).

convierte en otro reto, ya que existen diversas aplicaciones, programas integrados, plataformas y dispositivos de FDM en el mercado, por lo cual se requiere de una investigación exhaustiva de ambos elementos que se adapten a las necesidades de la aeronave, considerando factores de compatibilidad, facilidad de uso y estándares de seguridad.

Por otro lado, identificar la web gráfica adecuada resulta complejo en términos de usabilidad, adaptabilidad, integración con los sistemas a bordo de la aeronave y certificación. Se debe considerar que la interfaz debe ser simple e intuitiva, lo cual presenta una problemática debido a la cantidad de información que se maneja en el entorno aéreo. La solución a esto puede darse realizando futuras pruebas con usuarios reales que garanticen la facilidad de uso. En concordancia con lo anterior, la interfaz debe ser capaz de adaptarse a diferentes situaciones y requisitos de los pilotos, por lo que un trabajo futuro podría ser ofrecer opciones de personalización y configuración para los usuarios.

Por último, especificar los procedimientos de conexión muestra problemáticas de ajustes y tolerancias en cabina, mantenimiento y requisitos de certificación. El sistema debe ser posicionado de manera que no obstruya la visión o interfiera con las funciones que se llevan a cabo en la aeronave. Para esto, se proporcionará un plano que demuestre la localización adecuada del sistema. Adicionalmente, los sistemas deben ser compatibles con los sistemas de comunicación y de navegación, y su diseño debe ser de fácil mantenimiento, lo cual implica un desafío en la forma del sistema que cumpla con los requisitos de certificación establecidos por las autoridades reguladoras. Debido a esto, se exponen algunos protocolos de comunicación y circulares que rigen a la FAA para garantizar la seguridad y la fiabilidad.

Diseño metodológico

Esta sección explica la estrategia de trabajo enfocada en la implementación de una maleta electrónica de vuelo (Electronic Flight Bag-EFB) para visualizar un

sistema de gestión de vuelo (Flight Management System-FMS) en la aeronave prototipo Skiron. Dado que se trata de un avance científico, se presentará la organización del trabajo en forma de un diagrama de flujo para definir la red de comunicación del sistema, facilitar el intercambio de información mediante un programa y proporcionar un apartado que permita el uso y la integración del sistema en la aeronave (ver figura 2).

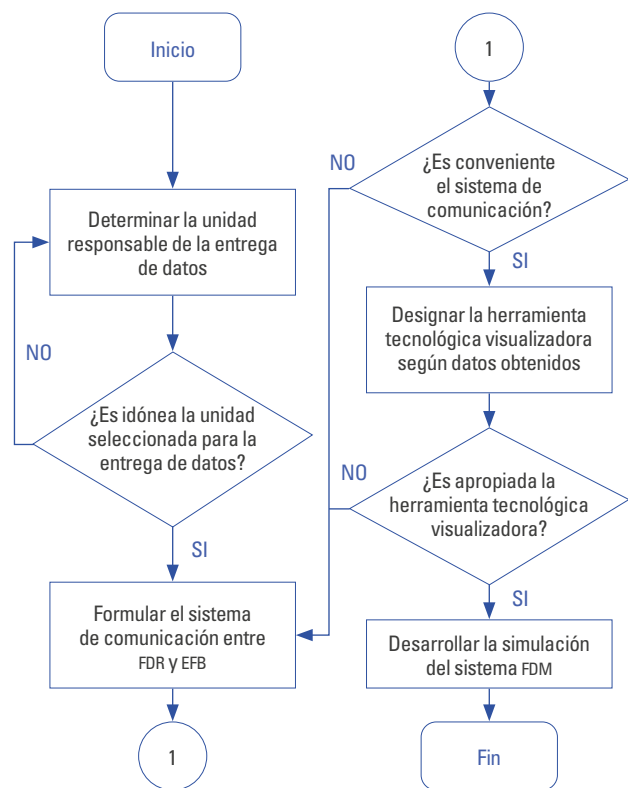


Figura 2. Diagrama de flujo orientado a la organización del trabajo
Fuente: elaboración propia.

La propuesta presentada por la Universidad de San Buenaventura, sede Bogotá, ganó la competencia de diseño general y preliminar de una aeronave STOL en el marco del Seminario Aeronáutico y Espacial de la Feria Aeronáutica Internacional celebrada en Rionegro, Antioquia, en julio de 2019 (FAC, 2018). A pesar de esto, los informes del prototipo de aeronave Skiron indican que falta un instrumento de aviónica

capaz de codificar parámetros operativos para una maleta electrónica simulada y modelada con el fin de monitorear los datos de vuelo en misiones como reabastecimiento, carga, extinción de incendios y ambulancia aérea. La solicitud presentada para el concurso solo incluyó estructuras reguladas por las directrices MIL-STD-1553 y MIL-STD-1773 para separar los instrumentos de aviónica que clasificarían y acoplarían provisionalmente a brigadas militares e intervenciones civiles (FAC, 2018). Sin embargo, estas normas no permiten la conexión de un EFB al bus de datos, debido al tipo de red que se utiliza y al registro limitado de datos que se puede almacenar

Considerando las restricciones de la FAC para un avión multipropósito tipo Commuter que pueda ingresar a áreas remotas con condiciones de pista condicionadas o no condicionadas, similar al CASA 212, y realizar numerosos vuelos cortos para realizar tareas como reabastecimiento de combustible, transporte de carga, extinción de incendios y ambulancia aérea, se ha modelado un recorrido de nueve fases que se detallan en la figura 3. Durante este recorrido, la aeronave realiza un rodaje en una pista de asfalto “taxiway” (fase 1), seguido del despegue y ascenso (fases 2 y 3), luego el vuelo en crucero (fase 4) que permitirá maniobras de despliegue de tropas (fase 5.1) o carga una vez alcanzado el techo de servicio máximo (fase 5). Después del vuelo en crucero (fase 6), se realiza un descenso controlado (fase 7) que lleva al aterrizaje temporal para el estacionamiento y la recirculación en tierra del vehículo (fase 8). Por último, se lleva a cabo el apagado total (fase 9).

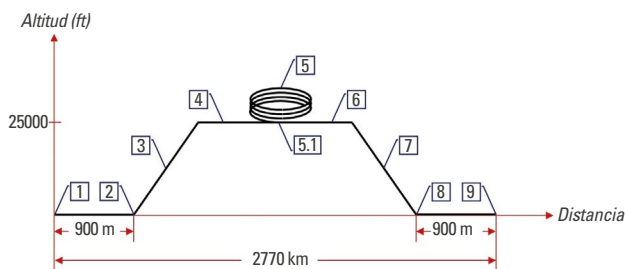


Figura 3. Plan de vuelo diseñado para la aeronave Skiron
Fuente: elaboración propia.

Para asegurar un desempeño óptimo en términos de velocidad, localización, eficiencia y seguridad en vuelo, se optó por implementar el sistema CDU-7000 como FMS (Flight Management System) para llevar a cabo las tareas requeridas durante los vuelos, tal como se muestra en la figura 4. Con el fin de mejorar aún más la operatividad, Rockwell Collins ha integrado un conjunto Proline 21 con interfaz ADS-B (Automatic Dependent Surveillance-Broadcast) para monitorear y reportar el tráfico aéreo en tiempo real. La posición de la aeronave es determinada por sistemas de navegación satelital y transmitida mediante radiodifusión a una estación receptora (Wang *et al.*, 2020).



Figura 4. CDU-7000 de Rockwell Collins. Sistema básico ADS-B
Fuente: tomado de Southeast Aerospace (2008).

Existen programas de seguridad, como el FOQA (Flight Operation Quality Assurance), que combinan la información de los registros de vuelo con otras fuentes y experiencias operacionales para mejorar la estabilidad operativa, la eficacia del entrenamiento, los métodos operativos, los métodos de mantenimiento e ingeniería, así como el control del tráfico aéreo. Según la FAA (Riggins, 2022), los datos se enrutan a un registrador de datos de vuelo, también conocido como Flight Data Recorder (FDR), para registrar parámetros específicos de rendimiento de la aeronave. El propósito del FDR es recopilar y registrar datos de una variedad de sensores de la aeronave en un medio diseñado para sobrevivir a un accidente, según lo establecido por la FAA (2010).

La Universidad de Dhaka ha publicado una investigación que detalla las fases operativas de los sistemas

de detección y el *hardware* necesario para recopilar y almacenar los datos obtenidos (Sarker *et al.*, 2016). El enfoque de la investigación es mejorar el acceso a los datos de vuelo mediante el diseño de un sistema conceptual rentable que se adapte a vehículos aéreos no tripulados (UAV-Unmanned Aircraft Vehicle), y se muestra una clara optimización en el proceso de diseño.

El diagrama presenta y describe los componentes principales del sistema FDAS (Flight Data Acquisition System), que utiliza un *software* para registrar los resultados finales en un formato de plano, a partir de la conversión de entradas analógicas en salidas digitales. Mediante el uso de sensores, es posible convertir los eventos naturales en señales digitales, ya sea con una salida de corriente o analógica, de modo que las señales filtradas y amplificadas se ajustan para que los datos adquiridos sean recibidos por el *hardware*. Todo el proceso de ingeniería se ilustra en la figura 5 (Sarker *et al.*, 2016).

Es importante mencionar que el término *hardware* se refiere al conjunto de componentes físicos que conforman un dispositivo electrónico, mientras que el término *software* se refiere al conjunto de programas y datos que permiten la ejecución de distintas tareas en un sistema computacional. El *hardware* es lo tangible del sistema, mientras que el *software* es lo intangible, pero ambos son fundamentales para el funcionamiento adecuado del dispositivo.

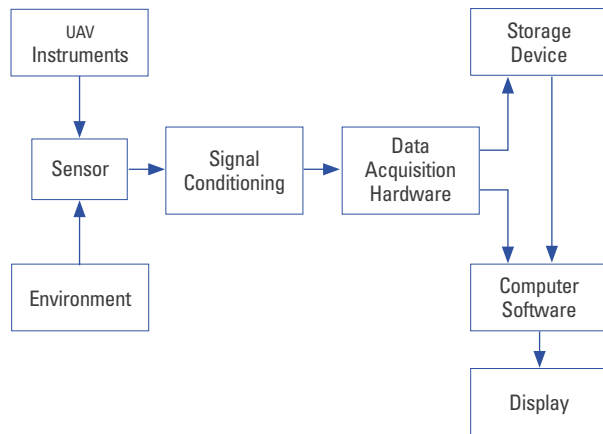


Figura 5. Esquema de datos de vuelo según su adquisición
Fuente: tomado de Sarker *et al.* (2016).

Para garantizar la ubicación exacta del avión, es fundamental utilizar un emisor GPS y de localización. La FAA solo permite el uso de sistemas de posicionamiento global en dispositivos electrónicos de vuelo si el *software* que se utiliza cumple con los requisitos de seguridad establecidos en la norma AC 120-76B. Este documento establece las directrices de certificación, aeronavegabilidad y uso operacional del EFB, que incluyen la documentación, la funcionalidad y la interoperabilidad del dispositivo. Además, se describen pruebas de interferencia, rendimiento e integridad del sistema para garantizar el cumplimiento del *software* (FAA, 2012).

La imagen de la figura 5 presenta una evaluación de costos razonable en comparación con los estándares habituales de fabricación, lo cual cumple con el objetivo principal de la investigación. Además, se ha incorporado un microcontrolador equipado con diversos tipos de sensores que permiten detectar patrones de movimiento rotativo, aerodinámica, dirección y fuerza magnética en el diseño conceptual del sistema FDA (consulte la figura 6).

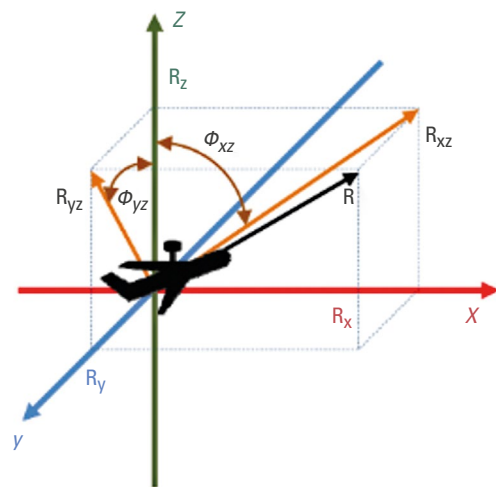


Figura 6. Marco de referencia de tres ejes para representar la velocidad de balanceo (*roll*), cabeceo (*pitch*) y guiñada (*yaw*)
Fuente: tomado de Sarker *et al.* (2016).

Las empresas aéreas y aeroespaciales han desarrollado unidades de gestión e interconexión de datos de vuelo, conocidas como Flight Data Interphase

and Management Units (FDIMU). Estas unidades tienen dos funciones principales: el procesamiento de los parámetros digitales y discretos para que el DMU (Data Management Unit) pueda convertirlos en formato numérico, que a su vez es utilizado por el QAR (Quick Access Recorder), el DFDR (Digital Flight Data Recorder) y el DAR (Digital AIDS Recorder) (FAC, 21 de junio de 2018). En la figura 7, se puede apreciar una representación del sistema.



Figura 7. Sistema Flight Data Interphase Management Unit (FDIMU)
Fuente: tomado de FAC (21 de junio de 2018).

Aquí, se presentan los Digital Flight Data Acquisition Unit (DFDAU), que funcionan como sistemas integrados para combinar las funciones de grabación y adquisición de datos mediante un Sistema de Monitoreo de la Condición de la Aeronave (ACMS-Aircraft Condition Monitoring System). Como se muestra en la figura 8, este sistema se especializa en proporcionar soluciones estandarizadas de *hardware* y *software* para la gestión de datos de aeronaves, llevada a cabo en un registrador PCMCIA interno (Montgomery, 2000). De esta manera, los sistemas de *software* ACMS de la DFDAU presentan una alta flexibilidad, lo cual permite a los usuarios definir los parámetros e informes necesarios para el mantenimiento y la supervisión de la eficiencia operativa de los datos de vuelo. Además, suministran un flujo continuo de datos digitales que permiten la detección, el monitoreo y el registro de fallos internos y externos del sistema, como lo destaca Montgomery (2000).



Figura 8. Digital Flight Data Acquisition Unit (DFDAU)
Fuente: tomado de Montgomery (2000).

La FDAU es una unidad encargada de recibir y procesar diversos parámetros (discretos, analógicos y digitales), provenientes de los sensores y los sistemas de aviónica de la aeronave, para luego enviarlos al registrador de datos de vuelo, conocido como FDR-Flight Data Recorder. Si se instala en conjunto con un Quick Access Recorder (QAR), la información de la FDAU se transfiere al FDR a través de marcos de datos específicos que dependen del fabricante de la aeronave. Actualmente, hay una tendencia hacia la integración de las funciones de la FDAU en el *software* de otros componentes del sistema de la aeronave, como en el caso de los registradores de vuelo mejorados o EAFR-Enhanced Airborne Flight Recorder (FAA, 2017). Estos dispositivos se pueden adquirir mediante un APM 2.3 “ArduPilot”, un sistema de piloto automático de código abierto que permite controlar vehículos y naves para convertirlos en aparatos autónomos capaces de realizar misiones GPS programadas con Waypoints. La placa del ArduPilot incluye un barómetro con tres ejes para el acelerómetro y tres ejes para el giroscopio, y es compatible con el sistema Arduino.

Estudios recientes realizados por la Universidad Santo Tomás demuestran que es posible determinar la disposición y elevación real de un vehículo aéreo utilizando datos de aceleración y giroscopio adquiridos. Esto se logra mediante la implementación de un detector con una sucesión de cálculos que interpreta los dígitos medidos del GPS para anticipar los resultados de trayectoria, posición y aceleración. Como resultado, se puede obtener una representación gráfica de

la aeronave en tiempo real a través del portal web IoT. Para ello, es necesario verificar el fichero del *router* inalámbrico RTAMS (Sistema de Monitoreo de Aeronaves en Tiempo Real, por sus siglas en inglés), el cual se basa en un diseño y prototipo RPAS, protocolo de comunicación y obtención de datos (Awazacko, 2021). La configuración del sistema se muestra en la figura 9.

El informe consideró varios protocolos de comunicación para garantizar un sistema de monitoreo de datos de vuelo en tiempo real, como XBEE, LORA y SIGFOX, entre otros. Debido a que el rango de transmisión de frecuencias era muy alto (VHF), se consideró que el protocolo de comunicación ideal era el internet satelital. Para obtener los datos de vuelo, se optó por el sistema ACARS (Aircraft Communication Addressing and Reporting System), que permite el intercambio de datos por medio de una red digital-terrestre. De esta manera, se pudo registrar en tiempo real el desempeño de los aeromodelos Apprentice 1500 y Renager 1600 (Awazacko, 2021).

Para la monitorización de los datos de vuelo de las aeronaves, se empleó un microcontrolador que se encargó de manejar los datos recibidos durante la trayectoria de vuelo establecida. A fin de visualizar los resultados en una página web, fue necesario diseñar y prototipar el RPAS. Para ello, se utilizó el protocolo XBEE, que permitió que los datos fueran transmitidos

al Raspberry Pi, y después se utilizaron herramientas de programación como Python para la visualización de los datos. En la figura 10, se muestra el proceso de visualización de los datos. El sistema de monitoreo desarrollado, que integra un módulo de GPS y sensores de bajo costo, permite conocer la posición y el movimiento de los aeromodelos en tiempo real. Este sistema se ha desarrollado como ejercicio académico y se han demostrado sus capacidades en los aeromodelos Apprentice 1500 y Renager 1600. En resumen, este proyecto ha permitido desarrollar una herramienta de seguridad aérea muy valiosa.

En vista de lo anterior, el propósito de este trabajo es desarrollar un lenguaje de programación que permita representar adecuadamente los resultados, siguiendo las directrices establecidas por la FAA y la Unidad Administrativa Especial de Aeronáutica Civil (Aerocivil). El objetivo es que los datos de vuelo para cada tarea necesaria se presenten de manera clara y organizada, utilizando la arquitectura de comunicación adecuada. Esto permitirá integrar el sistema como un equipo complementario a la aviónica existente en el avión Skiron.

Se espera proyectar las condiciones de la aeronave prototipo mediante la lectura de datos a partir de la plataforma Stewart, que está compuesta por un sistema de bucle cerrado conectado a una base con seis

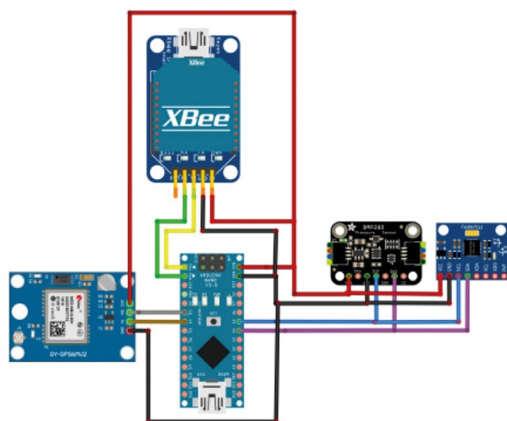


Figura 9. Circuito gráfico del sistema RTAMS
Fuente: tomado de Awazacko (2021).



Figura 10. Diseño e implementación de página web en la plataforma IoT
Fuente: tomado de Awazacko (2021).

actuadores que dirigen articulaciones prismáticas para recrear la operación en seis grados de libertad. Según el informe de Dasgupta y Mruthyunjaya (2000), la configuración del dispositivo es ampliamente aceptada en la industria aérea debido a su rango de movimiento y precisión, considerando la rigidez estructural de los eslabones que evitan fuerzas de inflexión no deseadas al resistir estados de compresión y tensión. El simulador de vuelo con forma de la plataforma se muestra en la figura 11. Cabe mencionar que la plataforma Stewart tiene diversas aplicaciones, como construcción naval, construcción de puentes, transporte y posicionamiento de antenas parabólicas de comunicación por satélite o telescopios



Figura 11. Plataforma Stewart utilizada por Lufthansa
Fuente: tomado de (Gudiño *et al.* (2018).

Las articulaciones giratorias de la plataforma Stewart se destacan por su alta precisión de posicionamiento, lo cual les permite ser controladas de manera remota. La construcción simplificada del modelo y la fabricación de sus partes posibilitan generar un ensamblaje eficiente que permite lograr las metas propuestas, tal como se muestra en la figura 12. Los movimientos del avión se realizan con base en las decisiones del piloto, en coordenadas divididas sobre uno o más ejes para emular el desplazamiento en zonas de control o primarias, como se verá más adelante. El balanceo se produce en el eje x (longitudinal), el cabeceo se da en el eje y (transversal) y la guiñada ocurre en el eje z (vertical)

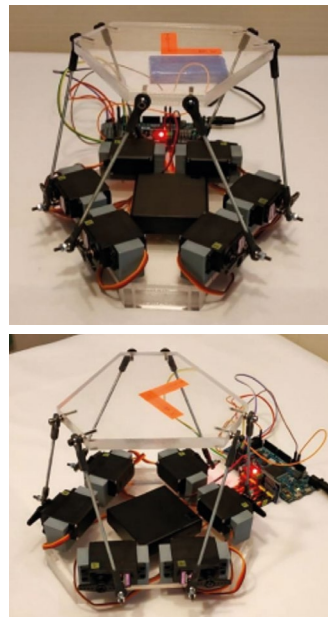


Figura 12. Modelo simplificado de plataforma Stewart
Fuente: tomado de Nieves (2020).

Cada avión tiene tres ejes imaginarios alrededor de los cuales se llevan a cabo los movimientos: el eje longitudinal, el eje transversal y el eje vertical. El balanceo, que es el movimiento alrededor del eje longitudinal, se produce cuando un ala sube mientras la otra baja; el cabeceo, que es el movimiento alrededor del eje transversal, se produce cuando la nariz del avión se eleva o baja; y la guiñada, que es el movimiento alrededor del eje vertical, se produce cuando la nariz del avión se desplaza hacia la izquierda o hacia la derecha (en inglés, estos movimientos se conocen como *roll*, *pitch* y *yaw*, respectivamente). La figura 13 muestra estos movimientos en un avión.

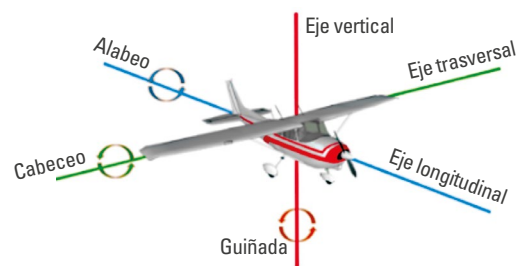


Figura 13. Ejes principales de una aeronave
Fuente: tomado de Fulgencio (24 de abril de 2018).

Diseño de ingeniería

Para simular con efectividad los movimientos principales de un avión, se utiliza una plataforma que incluye seis actuadores de referencia sg 90 con un mecanismo en forma de paralelo. Estos actuadores, también conocidos como servomotores, son motores eléctricos que están equipados con un circuito de control y son operados por múltiples conjuntos de engranajes. Pueden estar compuestos por partes neumáticas o hidráulicas, y se ingresa la señal de salida, tal como posición, velocidad, aceleración o desplazamiento. La transmisión del movimiento se lleva a cabo a través de eslabones que están unidos a la base mediante un cierre de forma y que se conectan a la plataforma, que a su vez mueve una placa superior. La función de esta placa es simular los seis grados de libertad de la aeronave: sus ejes x, y, z (lateral, vertical y longitudinal) y sus tres rotaciones (cabeceo, guiñada y balanceo).



Figura 14. Microservomotor sg 90 – 9G
Fuente: tomado de Flash Rc (2022).

Para transmitir señales y efectuar el movimiento en los microservomotores sg 90, se utiliza una placa PCA9685. Esta pieza incluye pines SCL/SDA que dirigen dieciséis servos a través de entradas y salidas PWM del microcontrolador. La placa es compatible con una fuente de 5V/5A para variar la frecuencia y resolución de 12 bits. La modulación por ancho de pulsos es una estructura idónea para estos actuadores, ya que puede enlazarse sin problemas con la PCA9685. Además, la placa dispone de pines I2C que permiten comunicarse

con la placa Arduino UNO y alimentar la sección lógica de los conectores. En la figura 15, se muestra la conexión entre estos tres elementos.

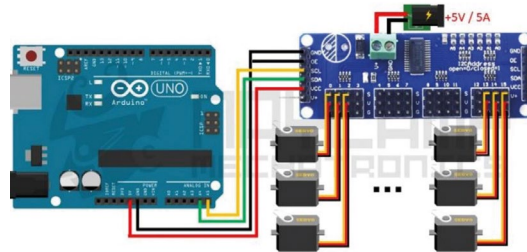


Figura 15. Control de servos con placa PCA9685 y Arduino UNO
Fuente: tomado de Naylamp Mechatronics (2016).

Es importante destacar que la placa Arduino UNO es la opción más adecuada para comenzar la programación y el control de la plataforma, ya que interactúa simultáneamente con el *software* y el *hardware*. Su utilidad radica en su código abierto, lo cual la convierte en una herramienta accesible para el público en general, con documentación y licencias gratuitas disponibles. La elección de esta placa se basó en la capacidad de ejecutar acciones precisas mediante interfaces entre ordenadores y el control de alimentación de los actuadores. La placa requiere una conexión desde un puerto USB o un transformador AC-DC para proporcionar energía al microcontrolador ATmega328P, y se distingue por su botón de reinicio, conector de audio analógico tipo Jack, entrada ICSP para programación en PIC/AVR, oscilador de cristal que genera señales eléctricas precisas a partir de la resonancia mecánica, y catorce pines digitales en modo bidireccional (entrada-salida).

Para complementar la funcionalidad del Arduino UNO, es necesario utilizar un dispositivo llamado USB Host Shield 2.0 MAX3421E Google Adk Uno Mega, el cual se muestra en la figura 16. Este dispositivo se monta encima del Arduino UNO y permite controlar periféricos que recolectan datos de entrada para ser enviados al controlador de salida (dispositivos esclavos). El módulo incluye un chip MAX3421E que implementa un acceso USB para controlar los circuitos analógicos y digitales. Este chip se comunica con los pines digitales del Arduino UNO, funcionando a una velocidad de 2.0 en el encabezado ICSP.

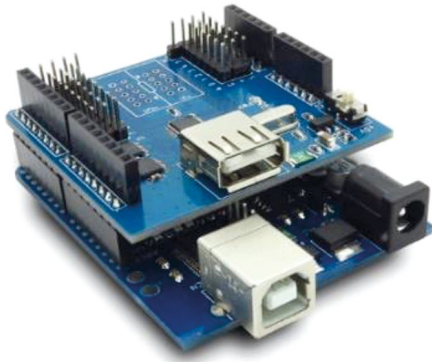


Figura 16. USB Host Shield 2.0 MAX3421E Google Adk Uno Mega
Fuente: tomado de Yorobotics (2021).

Para conectar elementos externos a la plataforma, como la unidad de medición inercial (IMU- Inertial Measurement Unit), que se explicará más adelante, se requiere utilizar un USB Host Shield. Esto permitirá conectar un control como el *joystick* Logitech Extreme 3D Pro para enviar señales a los servomecanismos, los cuales simularán con precisión los movimientos enviados por el piloto de acuerdo con las condiciones de vuelo presentes. Los botones ubicados en la parte superior del *joystick* son programables y permiten la ejecución de comandos complejos o individuales. Las figuras 17 y 18 muestran la configuración entre los dispositivos mencionados.

Es importante destacar que el *joystick* Logitech Extreme 3D Pro ha sido elegido como dispositivo controlador de la plataforma debido a su precisión, sensibilidad, durabilidad y precio accesible. Este *joystick* cuenta con sensores de alta calidad que permiten un control de posicionamiento sumamente preciso, lo cual se traduce en una respuesta óptima y en cambios significativos en la posición de la plataforma en función de los movimientos más mínimos. Además, la configuración de botones programables es capaz de alterar el movimiento o ajustar la posición deseada. En resumen, se ha elegido el *joystick* Logitech Extreme 3D Pro por su excelente desempeño, durabilidad y accesibilidad económica.

Para censar las condiciones de vuelo, se utilizará una MPU 6050, también conocida como una IMU. Esta unidad es capaz de realizar mediciones de navegación

precisas gracias a la integración de giroscopios y acelerómetros de tipo MEMS (Microelectromechanical Systems), con un procesador Host DMP que puede medir tres ejes cada uno (seis grados de libertad en total). Los pines I2C permiten la interconexión de los sensores con el Arduino UNO para capturar las salidas digitales sincrónicamente en los canales x, y, z, con una resolución de 16 bits. Además, la figura 19 muestra cómo se debe conectar correctamente la IMU a la placa Arduino UNO. Las especificaciones del fabricante indican que la MPU 6050 es capaz de procesar algoritmos de movimientos complejos de manera eficiente, lo cual la hace adecuada para la simulación de vuelo.

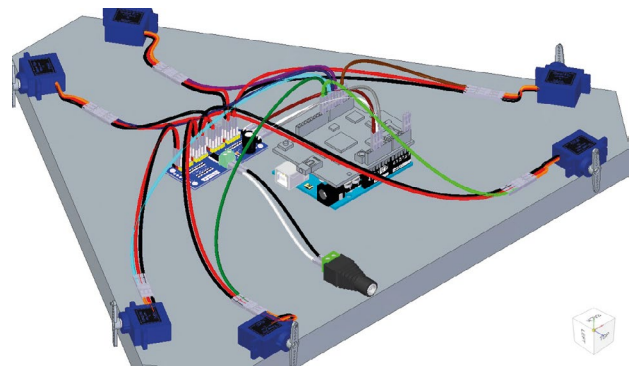


Figura 17. Conexiones de servos con placa PCA9685, Arduino UNO y USB Host Shield 2.0 MAX3421E
Fuente: elaboración propia.



Figura 18. Conexiones entre computador, Arduino UNO y USB Host Shield 2.0 MAX3421E Joystick
Fuente: elaboración propia.

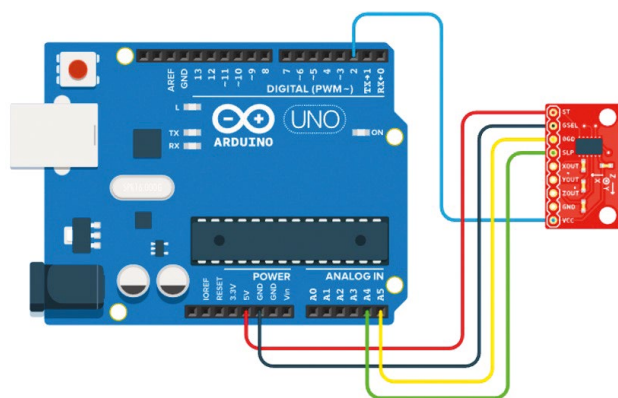


Figura 19. Conexión entre Arduino UNO y MPU6050
Fuente: elaboración propia.

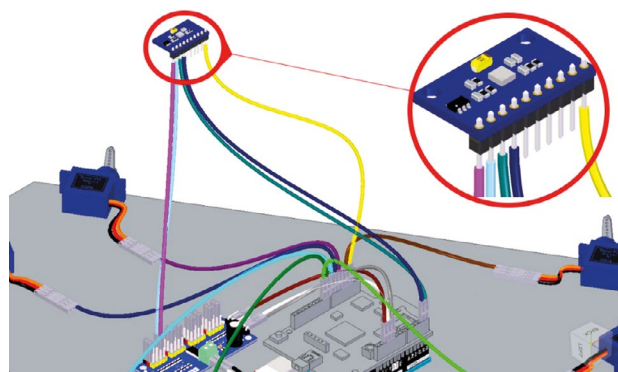


Figura 20. Detalles y ubicación de la IMU en plataforma Stewart
Fuente: elaboración propia.

Para obtener la máxima precisión en los movimientos generados, la IMU se colocará en la parte superior, tal como se indica en la figura 20. La información capturada por la IMU se utilizará para visualizar los movimientos en un sistema de gestión de vuelo (FMS) mediante una interfaz gráfica diseñada específicamente para el prototipo de aeronave Skiron, como se planteó al comienzo de este texto. Además, en la figura 21 se muestra la disposición y las conexiones de cada elemento del sistema.

Como se mencionó antes, el USB Host Shield 2.0 MAX3421E se utiliza para enviar señales a los servomecanismos y simular los movimientos enviados por una palanca de control, en este caso el *joystick* Logitech Extreme 3D Pro. Para lograr una simulación precisa de las órdenes del usuario, es necesario escribir un código en Arduino. Para hacerlo, primero se debe incluir la librería necesaria que se encuentra en la ruta: Programas > Incluir librería > Gestionar Librerías > usb Host Shield Library 2.0

Una vez incluida la librería, se realiza la siguiente ruta que permite controlar la lectura de datos en circuitos analógicos y compuertas lógicas digitales, es decir, elementos tanto digitales como físicos. Es importante destacar que, en el apartado HID, se ajusta el código a los objetivos establecidos en el documento en los módulos `le3dp`, `le3dp_rptparser.cpp` y `le3dp_rptparser.h`. Estos módulos se pueden encontrar en la ruta Archivos > Ejemplos > USB Host Shield Library 2.0 > HID > `le3dp`.

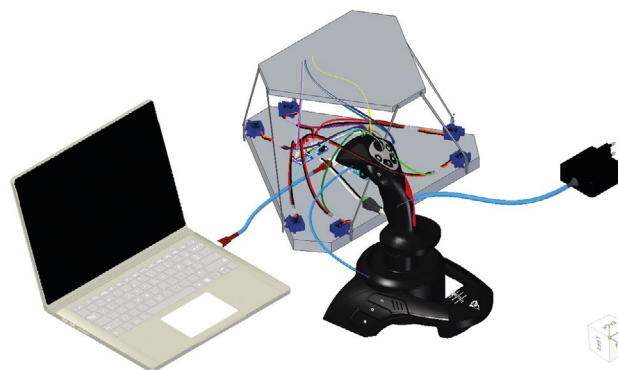


Figura 21. Disposición ergonómica de los elementos
Fuente: elaboración propia.

En la figura 22, se puede observar que la primera parte del código se encarga de declarar las librerías que se utilizarán. Para programar el sensor, obtener los datos correspondientes y seleccionar la herramienta tecnológica que mostrará los resultados, se asigna una variable "int" a los canales x, y, z de 16 bits. A continuación, se crean los objetos que contienen los atributos de las librerías, como USB, USBHub, HIDUniversal, JoystickEvents y JoystickReportParser. En el bloque *set up* se inicia el puerto serial a 57600, que es la velocidad máxima que soporta Arduino, y se informa al usuario mediante un mensaje de inicio que se espera que el USB esté disponible


```

#include <usbhid.h>
#include <hiduniversal.h>
#include <usbhub.h>

#include "le3dp_rtparser.h"

// Satisfy the IDE, which needs to see the include statement too.
("USB_Host_Shield_2.0/SPP.ino at master - Github")
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"
#ifdef dobogusinclude
#include <spi4teensy3.h>
#else
#include <SPI.h>

MPU6050 sensor;
int ax, ay, az;
int gx, gy, gz;

USB                Usb;
USBHub             Hub(&Usb);
HIDUniversal       Hid(&Usb);
JoystickEvents     JoyEvents;
JoystickReportParser Joy(&JoyEvents);

void setup()
{
  Serial.begin(57600); //Iniciando puerto serial
  Wire.begin();       //Iniciando I2C
  sensor.initialize(); //Iniciando el sensor

  if (sensor.testConnection()) Serial.println("Sensor iniciado
correctamente"); else Serial.println("Error al iniciar el sensor");

```

Figura 22. Configuración inicial y cíclica del código para enviar señales
Fuente: elaboración propia.

En la segunda parte del código, se incluye la codificación de las librerías y sus ajustes correspondientes. Aquí, se declara la variable `OnGamePadChanged` de la clase `JoystickEvents`, que permitirá mostrar los valores leídos por el *joystick* Logitech Extreme 3D Pro cada vez que haya un cambio en la señal. De tal manera, solo es necesario agregar la actualización de los valores de salida. Es importante tener en cuenta que los pines de salida deben ser configurados en el código para cada servo, ya que el Arduino UNO cuenta con catorce pines disponibles.

Después de imprimir cada variable digital correspondiente a los botones del *joystick*, se seleccionan los bits de cada una y se declara un vector llamado "mul" que discrimina estos valores para seleccionar el bit correspondiente entre las variables de 8 bits que están concatenadas. Luego, se insertan dos bucles que imprimen veinte valores digitales. A continuación, se

realiza un mapeo de los ejes x y y en el rango de 0 a 1023, mientras que el eje z va de 0 a 180 para la torsión (*twist*) y *slider*. El propósito de esto es actualizar los datos de salida de los servos. La figura 23 muestra parte de la impresión de cada variable digital.

```

#include "le3dp_rtparser.h"

JoystickReportParser::JoystickReportParser(JoystickEvents *evt) :
  joyEvents(evt)
{}

"void Joystick_Report_Parser::Parse(USBHID *hid, bool is_rpt_id,
uint8_t len, uint8_t *buf)"
{
  bool match = true;

  // Checking if there are changes in the report since
  the method was last called
  for (uint8_t i=0; i<RPT_GAMEPAD_LEN; i++){
    if (buf[i] != oldPad[i]) {
      match = false;
      break;
    }
  }

  // Calling GamePad event handler
  if (!match && joyEvents) {
    joyEvents->OnGamePadChanged((const
GamePadEventData*)buf);

    for (uint8_t i=0; i<RPT_GAMEPAD_LEN; i++)
      oldPad[i] = buf[i];
  }
}

void JoystickEvents::OnGamePadChanged(const GamePadEventData *evt)
{
  Serial.print("X: ");
  PrintHex<uint16_t>(evt->x, 0x80);
  Serial.print(" Y: ");
  PrintHex<uint16_t>(evt->y, 0x80);
  Serial.print(" Hat Switch: ");
  PrintHex<uint8_t>(evt->hat, 0x80);
  Serial.print(" Twist: ");
  PrintHex<uint8_t>(evt->twist, 0x80);

```

Figura 23. Estructura funcional del código al generar un cambio en las señales recibidas
Fuente: elaboración propia.

Finalmente, se procede a implementar la funcionalidad del código, añadiendo la librería de los servomotores. La clase `JoystickEvents` utiliza el constructor de la librería "Servo.h" para crear objetos para cada uno de los servomotores. Como se muestra en la figura 24, se le asigna un nombre a cada uno de ellos, como `servo3`, `servo5`, `servo6`, `servo9`, `servo7`, `servo8`. Estos objetos se utilizan en el método `OnGamePadChanged`, que representa una estructura que permite acceder

directamente a los canales de x y y . El canal z , por su parte, se puede acceder a través de la variable “int” en la IMU MPU 6050 que se explicó antes. En resumen, en esta sección del código se describen las conexiones, la lectura de datos, la inicialización del programa y las variables recibidas de manera general.

```
#if !defined(__HIDJOYSTICKRPTPARSER_H_)
#define __HIDJOYSTICKRPTPARSER_H_
#include <Servo.h>
#include <usbhid.h>

struct GamePadEventData
{
    union { //axes and hut switch
        uint32_t axes;
        struct {
            uint32_t x : 10;
            uint32_t y : 10;
            uint32_t hat : 4;
            uint32_t twist : 8;
        };
    };
    uint8_t buttons_a;
    uint8_t slider;
    uint8_t buttons_b;
};

class JoystickEvents
{
public:
    Servo servo3,servo5,servo6,servo9,servo7,servo8;
    virtual void OnGamePadChanged(const GamePadEventData *evt);
};

#define RPT_GAMEPAD_LEN sizeof(GamePadEventData)/sizeof(uint8_t)

class JoystickReportParser : public HIDReportParser
{
public:
    JoystickEvents *joyEvents;
};
```

Figura 24. Configuración del *joystick* que da movilidad a los servomotores
Fuente: elaboración propia.

Para obtener resultados precisos, es importante que la orientación del sensor IMU MPU6050 esté correctamente posicionada, como se muestra en la figura 25. En el siguiente apartado, se expondrán los resultados obtenidos. Además, matemáticamente se puede demostrar la primera ley de Newton al usar el *joystick* en un rango de tiempo en el que la rapidez de un objeto varía en función de su masa, como se muestra en la ecuación 1:

$$\vec{F} = m \cdot \vec{a} \quad (1)$$

La aceleración es la tasa de cambio de la velocidad respecto al tiempo y es proporcional al desplazamiento. Esto se puede expresar matemáticamente mediante la ecuación 2, que describe el cambio en la posición en los ejes x , y y z con respecto al tiempo. Es decir, la ecuación 2 expresa el cambio de velocidad en cada eje, lo cual resulta en un cambio de posición en el tiempo, por lo que es importante tener en cuenta la orientación del sensor IMU MPU6050.

$$\vec{a} = \frac{\partial \vec{v}}{\partial t} = \frac{\partial^2 \vec{r}}{\partial t^2} \quad (2)$$

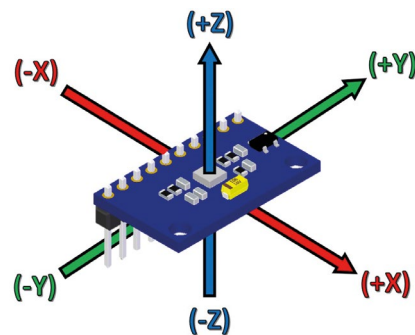


Figura 25. Posición y representación de los ejes del MPU 6050
Fuente: elaboración propia.

En la evaluación de la eficacia del sistema FMS en el avión prototipo Skiron, se utiliza una herramienta de análisis de datos de vuelo y simulación aérea llamada Tacview. Este *software* es capaz de leer, grabar y reproducir datos de vuelo en tiempo real en diferentes simuladores. Además, cuenta con diversas opciones de visualización, tales como una visualización 3D y la superposición de datos de vuelo sobre mapas y terrenos reales.

Para recrear una trayectoria de vuelo en un avión con características de patrulla marítima, se optó por utilizar Flight Simulator x como simulador de vuelo en tierra. Dicha trayectoria fue trazada según las especificaciones de la RAC 15, capítulo IV (Aerocivil, 2023). La información de los aeródromos necesarios para la ruta SKBO-SKIB (Bogotá-Ibagué), que se muestra en la figura 26, fue localizada y copiada.

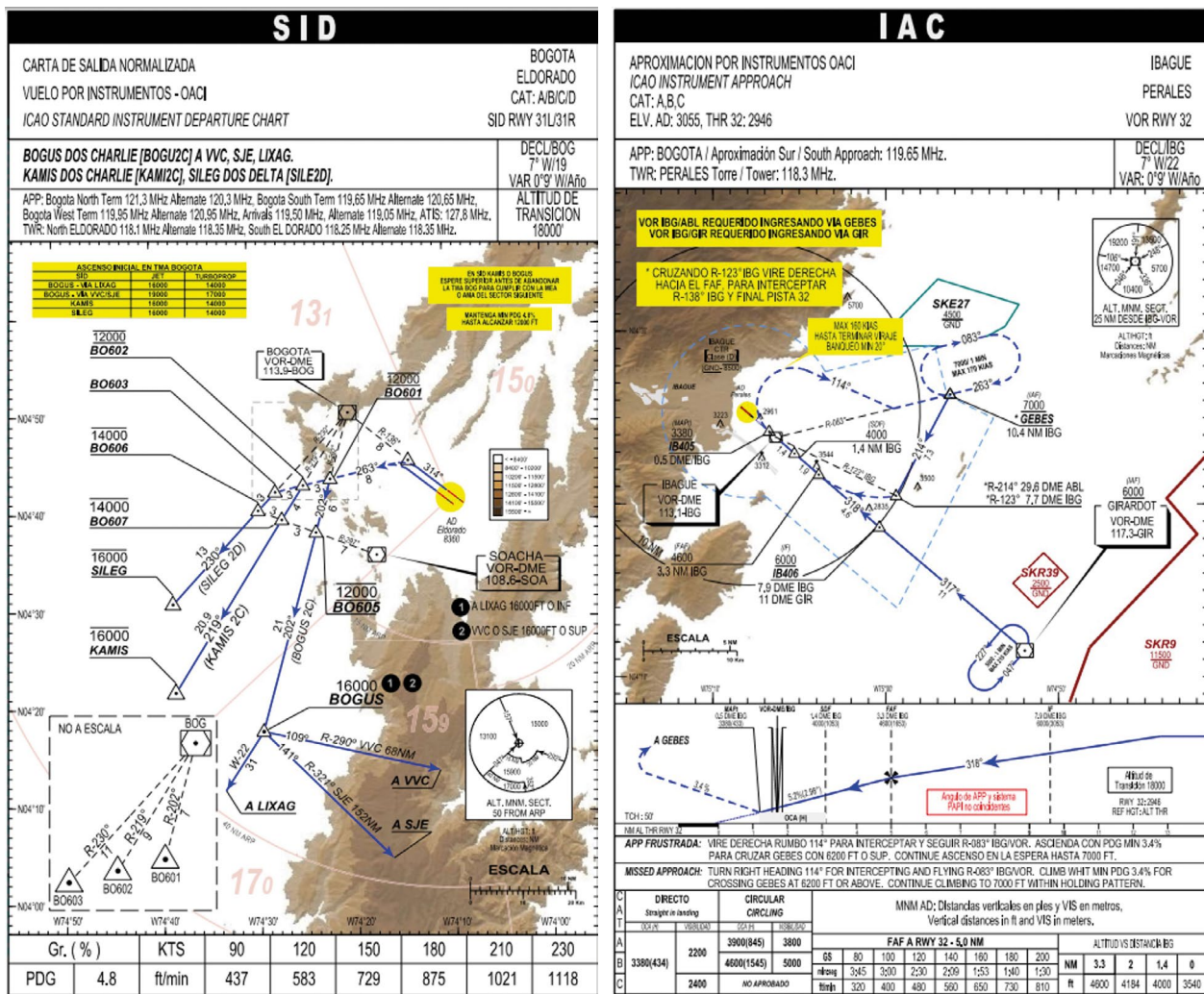


Figura 26. Cartas de navegación SID y STAR indicando trayecto Bogotá-Ibagué
Fuente: Aerocivil (2023).

El gráfico de la figura 27 muestra el movimiento rotacional durante una misión de operación máxima establecida que duró 19 minutos en total. El límite superior se alcanzó en el minuto 17 y el límite inferior fue notable en el minuto 5. Durante el despegue por la ruta 31 del Aeropuerto Internacional El Dorado, fue fundamental garantizar la estabilidad en la trayectoria del avión, por lo cual se buscó minimizar la variación del balanceo al comienzo. De acuerdo con la ruta trazada, se supone que el balanceo en el minuto 17 se debió al viraje para interceptar el Heading 263 de las cartas, mientras que el límite inferior se produjo al momento de realizar el viraje a la derecha hacia el curso 318.

En cuanto a la desviación del elevador, esta produce un movimiento conocido como cabeceo, el cual se puede apreciar en la figura 28. Para el estudio realizado, se registraron 19 minutos de vuelo, durante los cuales se alcanzó el límite superior en el minuto 4 y el límite inferior en el minuto 15. Durante el despegue no se produce ningún cambio en la sustentación, por lo cual la variación del cabeceo es nula, y el elevador y el estabilizador horizontal están alineados en esta fase del vuelo. Sin embargo, en el minuto 4 se registró el mayor valor de esta actitud, probablemente debido al ascenso de 12 000 pies que se debía realizar una vez interceptado el Heading 263. Por otro lado, el menor

valor constatado fue en el minuto 15, cuando se debía acceder al Aeropuerto Nacional Perales por la pista 32 tomando GEBES a 7000 pies.

La figura 29 muestra el movimiento de guiñada que se produce al desviar el timón. El intervalo de tiempo estudiado fue de 19 minutos, durante los cuales se registró un límite superior en el minuto 1 y un límite inferior en el minuto 18. Durante el despegue, las fuerzas asimétricas del viento pueden alterar la trayectoria del avión en la pista. Por eso, se utilizan los pedales para controlar el timón de dirección y garantizar la dirección correcta entre la ruta y la pista de aterrizaje. En este caso, el uso de los pedales se observa en la salida SILEG 2 DELTA para llegar a la ruta 31 durante el despegue, mientras que en el minuto 18 el movimiento de los pedales es breve durante el aterrizaje en la pista 32 del Aeropuerto Nacional Perales.

Debido a que este trabajo es un ejercicio académico que aún se encuentra en desarrollo y que se enfoca en la evaluación de un sistema acorde al diseño conceptual de una aeronave que no ha sido fabricada hasta la fecha, no es posible proporcionar una estimación de costos en el momento. Además, es importante tener en cuenta que el precio de las maletas electrónicas de vuelo dependerá de factores como: *hardware* y *software*

utilizados, instalación y configuración necesarias, capacitación de uso, monitoreo y análisis continuo, o mantenimiento y actualizaciones requeridas. Esto influye en el precio de las maletas electrónicas de vuelo, por lo que no es posible estipular un valor. Sin embargo, es importante señalar que en el mercado existen marcas privadas que ofrecen diferentes tarifas según la licencia o suscripción. Por ejemplo, Simplify3D, 3DPrinterOs y MatterControl son algunas de las marcas disponibles. La versión básica y personal de Simplify3D es la opción más económica, con un costo de 149 dólares, mientras que la versión empresarial de 3DPrinterOs puede llegar a costar hasta 10 000 dólares anuales para una licencia compartida.

En este momento, solamente se ha programado el Arduino y la IMU, y se ha seleccionado una pantalla tentativa para el sistema EFB; el costo de adquisición del sistema propuesto es significativamente más bajo en comparación con los sistemas existentes. Esto se debe a que el *software* utilizado es de código abierto y compatible con múltiples tabletas inteligentes. Esta aproximación permitirá mejorar el sistema propuesto en futuros trabajos y calcular con precisión el costo total de la maleta electrónica de vuelo, considerando los elementos adicionales que sean previstos, a fin de optimizar principalmente los factores de precisión.

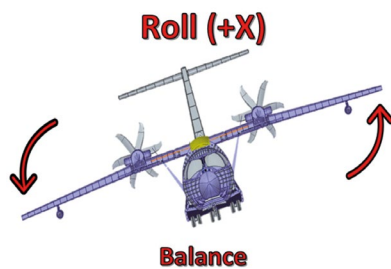
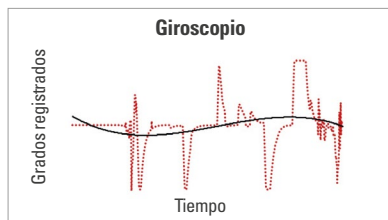


Figura 27. Gráfica presentada durante el balanceo según Tacview
Fuente: elaboración propia.

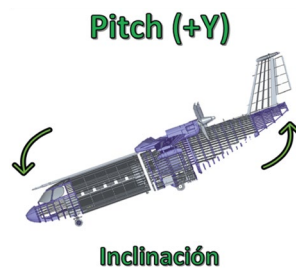
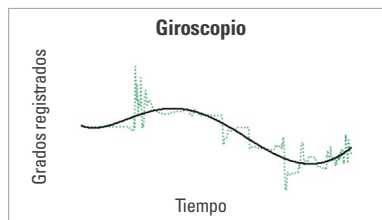


Figura 28. Gráfica presentada durante el cabeceo según Tacview
Fuente: elaboración propia.

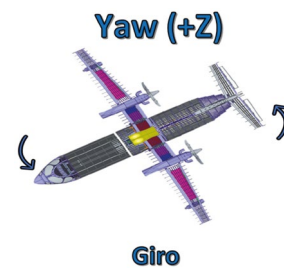
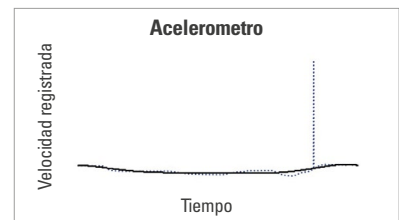


Figura 29. Gráfica presentada durante la guiñada según Tacview
Fuente: elaboración propia.

Resultados

Para el análisis correspondiente de las respuestas obtenidas por la IMU, se hicieron pruebas en la plataforma Stewart únicamente con los movimientos principales que la aeronave Skiron estará capacitada para realizar en el futuro. Las gráficas del acelerómetro y el giroscopio para los tres casos serán anexadas y explicadas en breve. Cabe aclarar que los puntos seccionados son los datos suministrados y la línea sólida es una línea de tendencia para percibir el comportamiento del fenómeno presentado.

La gráfica de la figura 30 muestra el movimiento rotacional presentado en el acelerómetro y el giroscopio sobre el eje x. Su intervalo de tiempo fue 14,2 segundos, el límite superior fue obtenido en el segundo 8,8, mientras que el límite inferior fue en el segundo 6. Al comparar la figura 27 con la figura 30, es notoria una línea de tendencia más pronunciada y constante en la gráfica de la figura 27, la razón es que el simulador de vuelo toma en consideración las cargas del avión en cada fase de vuelo, mientras que la plataforma Stewart no está diseñada para tener esa característica. Además, la tasa de tiempo tanto en la plataforma como en el simulador de vuelo usado difieren en el tiempo de muestreo, ya que el primero solo presenta segundos, mientras que el segundo presenta minutos y segundos. Pese a esto, las gráficas tienen comportamientos similares.

A partir de lo anterior, las variaciones en las gráficas de la figura 27 y la figura 30 radican en el balanceo producido durante la trayectoria de vuelo causada por la desviación de los alerones del Skiron. Esto indica que con una mayor desviación hacia abajo habrá un aumento en la sustentación, como lo demuestra al inicio de la gráfica, y posteriormente se presenta una desviación hacia arriba dando a entender la reducción en la sustentación. De ello es posible comprender que los alerones, al funcionar en pares, no distribuyen las fuerzas equitativamente, por lo que el valor del giro neto alrededor del centro de gravedad difiere del eje de balanceo (eje x).

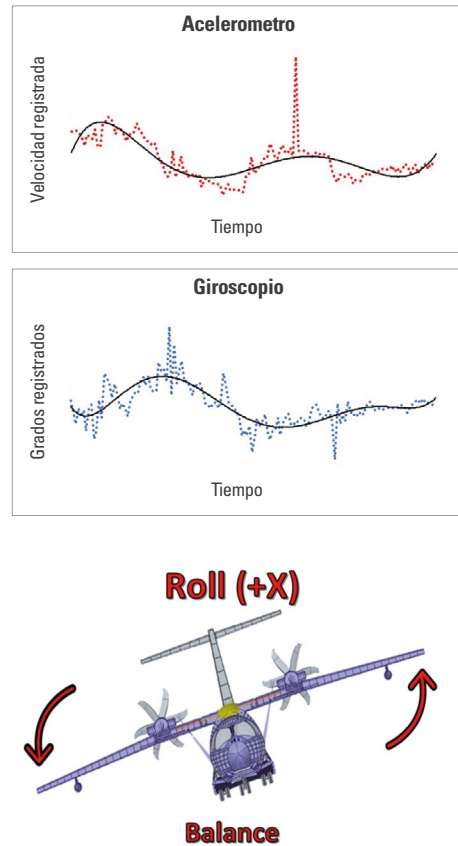


Figura 30. Datos presentados durante un balanceo en el acelerómetro y el giroscopio

Fuente: elaboración propia.

Por otro lado, a lo largo de una trayectoria de vuelo se produce la desviación del elevador. Esta acción causa el movimiento de cabeceo (ver figura 31). Para el estudio, el intervalo de tiempo fue de 13,9 segundos, el límite superior fue obtenido en el segundo 10,9, mientras que el límite inferior fue en el segundo 11,3. En la figura 28, la actitud de cabeceo es constante en gran medida, lo cual indica estabilidad y control, caso contrario a la figura 31 que presenta mayor fluctuación; se infiere que esto pudo ser causado por los movimientos mínimos involuntarios que registró el *joystick*. A pesar de todo, la línea de tendencia en ambos casos demuestra un comportamiento semejante para distintas fases de tiempo registradas.

A partir de lo presentado en las gráficas, se infiere que el elevador y el estabilizador horizontal están

alineados, lo cual no produce un cambio en la sustentación del Skiron idealmente. Sin embargo, enseguida se observa un aumento en la sustentación debido a la desviación hacia abajo causada por el *joystick* controlado, y finalmente, se planea regresar al estado inicial del avión por medio de una desviación hacia abajo causando la disminución de la sustentación. Los desvíos en el elevador permiten girar el avión sobre su centro de gravedad (eje *y*).

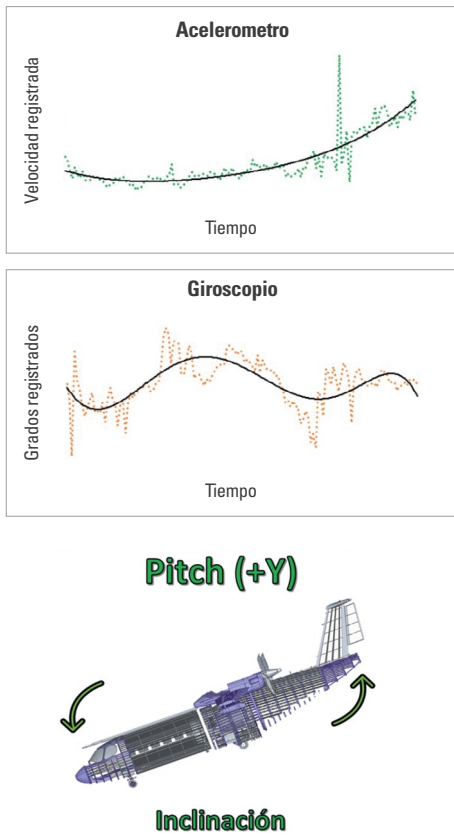


Figura 31. Resultados dados durante un cabeceo en el acelerómetro y el giroscopio
Fuente: elaboración propia.

Según representa la aeronave Skiron, se evidencia un movimiento de guiñada causado por la desviación del timón (*rudder*) en la figura 32. Para ello, el intervalo de tiempo fue de 14,5 segundos, el límite superior fue obtenido en el segundo 0,1, mientras que el límite inferior fue en el segundo 3. La variación repentina al final

del vuelo (figura 29) se produce posiblemente por el viento cruzado que se presentó en el simulador de vuelo. Además, las cargas y las velocidades del viento registradas en el simulador afectaron la dirección de ruta entre el avión y la pista de aterrizaje, siendo otro parámetro que omite la plataforma Stewart.

En este sentido, las tablas prueban que al desviar el timón alrededor de su centro de gravedad se genera un cambio en la fuerza lateral del avión. Para el estudio, se identifica un pico descendente en la lectura de los datos según lo obtenido por la IMU, lo cual corresponde a un aumento de la fuerza lateral hacia la derecha en relación con la desviación del timón dirigido hacia la izquierda. Luego de esta acción, la gráfica permanece idealmente constante, caso contrario al del giroscopio que varía a lo largo del tiempo.

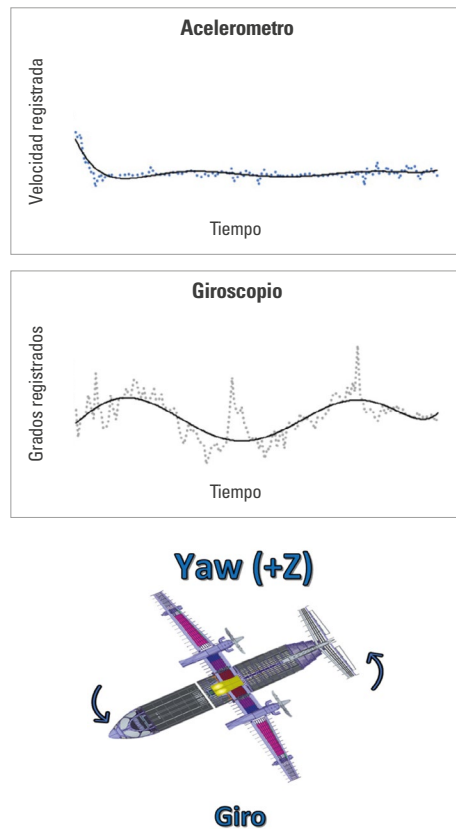


Figura 32. Datos obtenidos durante una guiñada en el acelerómetro y el giroscopio
Fuente: elaboración propia.

Conclusiones

Luego de llevar a cabo este estudio, se puede concluir que el diseño preliminar del sistema FDM para la aeronave prototipo Skiron fue capaz de identificar los parámetros de actitud en cada fase de vuelo, lo cual permite establecer los límites máximos y mínimos necesarios para una operación segura por parte del operador.

Una vez definida la arquitectura de comunicación, se puede monitorear el comportamiento del vehículo a lo largo de toda la trayectoria mediante sensores y sistemas inerciales incorporados, que están accesibles al público y son un elemento ideal para desarrollar proyectos futuros centrados en la seguridad operacional. Este estudio preliminar está sujeto a mejoras que podrían incorporar otras variables de medición y reducir los costos de operación en tierra, así como mejorar la eficiencia funcional de la aeronave. Con estas mejoras, el sistema se transformaría en un equipo trascendente para la futura integración.

Se puede aumentar la precisión de la adquisición de datos al incorporar un sensor más sofisticado. En el estudio, se utilizaron pruebas con un sistema económico con fines educativos solamente. Las relaciones entre la exactitud y la calidad de los datos establecen algunas incertidumbres. Por lo tanto, la implementación de nuevas tecnologías permitirá predecir con mayor fiabilidad los comportamientos de la aeronave en el futuro y, por ende, reducir posibles situaciones de riesgo durante su operación.

Referencias

- Aguilar Castro, L. M. y Ortiz Astudillo, C. P. (2017). *Análisis de la tecnología ArduPilot orientada a la implementación de sistemas de control automático* (tesis de grado, Universidad Politécnica Salesiana). Repositorio institucional UPS. <https://dspace.ups.edu.ec/bitstream/123456789/13480/1/UPS-CT006885.pdf>
- Allen, D. (2008). Electronic Flight Bag: Real-Time Information Across an Airline's Enterprise. *Aero Magazine*, (8), 22-27.
- https://www.boeing.com/commercial/aeromagazine/articles/qtr_2_08/AERO_Q208.pdf
- Arduino. (2022). Arduino UNO. <https://arduino.cl/arduino-uno/>
- Awazacko, J. A. (2021). *Desarrollo de un dispositivo de monitoreo en tiempo real de los sistemas de una aeronave mediante un protocolo de comunicación inalámbrica RTAMS (Real-time Aircraft Monitoring System)* (tesis de grado, Universidad de Santo Tomás). Repositorio institucional Usta. <https://repository.usta.edu.co/handle/11634/35156>
- Dasgupta, B. y Mruthyunjaya, T. S. (2000). The Stewart Platform Manipulator: A Review. *Mechanism and Machine Theory*, 35(1), 15-40. [https://doi.org/10.1016/S0094-114X\(99\)00006-3](https://doi.org/10.1016/S0094-114X(99)00006-3)
- Delhom, J. (2014). *Flight Data Analysis (FDA), a Predictive Tool for Safety Management System (SMS)*. Airbus. <https://safetyfirst.airbus.com/flight-data-analysis-fda-a-predictive-tool-for-safety-management-system-sms/>
- Federal Aviation Administration (FAA). (2010). *Advisory Circular. Airworthiness and Operational Approval of Digital Flight Data Recorder Systems*. https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_20-141B.pdf
- Federal Aviation Administration (FAA). (2012). *Advisory Circular. Guidelines for the Certification, Airworthiness, and Operational Use of Electronic Flight Bag*. https://www.faa.gov/documentlibrary/media/advisory_circular/ac%20120-76b.pdf
- Federal Aviation Administration (FAA). (2017). *Advisory Circular. Authorization for Use of Electronic Flight Bags*. https://www.faa.gov/documentlibrary/media/advisory_circular/ac_120-76d.pdf
- Federal Aviation Administration (FAA). (2021). *Automatic Dependent Surveillance-Broadcast (ADS-B)*. https://www.faa.gov/air_traffic/technology/adsb
- Federal Aviation Administration (FAA). (2022). *Flight Data Monitoring*. <https://www.faa.gov/sites/faa.gov/files/2022-01/Flight%20Data%20Monitoring.pdf>
- Flash rc. (2022). *Novedades*. <https://www.flashrc.com/es/>
- Fuerza Aérea Colombiana (FAC). (2018, 21 de junio). *Primer Concurso de Diseño Aeronáutico en Conmemoración al Centenario FAC*. Sala de Prensa. <https://secad.fac.mil.co/es/noticias/primer-concurso-de-diseno-aeronautico-en-conmemoracion-al-centenario-fac>
- Fulgencio. (2018, 24 de abril). *¿Cómo se controlan los aviones en vuelo? Sociedad Aeronáutica Española*. <https://www.sociedad aeronautica.org/control-aviones/>

- García, V. (2015). *Prácticas con el sensor de 3 ejes y Arduino*. Electrónica Práctica Aplicada. <https://www.diarioelectronicohoy.com/blog/sensor-mpu6050>
- Gudiño, J., Alcalá, J., Narrarro, H., Velez, D. y Charre, S. (2018). *Diseño y modelo cinemático de un robot delta para el diagnóstico y rehabilitación* (tesis de grado, Universidad Autónoma del Estado de Hidalgo). Repositorio institucional UAEH. <https://repository.uaeh.edu.mx/revistas/index.php/xikua/article/download/2764/2789/>
- Montgomery, D. (2000). *Manual de seguridad de vuelo para operadores*. Gain.
- Naylamp Mechatronics. (2016). *Tutorial Módulo Controlador de Servos PCA9685 con Arduino*. https://naylampmechatronics.com/blog/41_tutorial-modulo-controlador-de-servos-pca9685-con-arduino.html
- Nieves Belmar, D. (2020). *Diseño e implementación de un plataforma Stewart* (tesis de máster, Universitat Politècnica de València). Repositorio institucional UPV. <https://riunet.upv.es/handle/10251/158625>
- Riggins, D. W. (2022). *Flight Program Operations*. Federal Aviation Administration. https://www.faa.gov/air_traffic/flight_info/flight_ops
- Sarker, T., Hannan, P., Shahed, S. A., Rahman, N. y Sakib, S. N. (2016). *Conceptual Design of a Low Cost Flight Data Acquisition System for Analyzing Flight Behavior of Small Unmanned Aerial Vehicles* [ponencia]. Conference ICCIT 2016. Northsouth University, Dhaka, Bangladesh. Diciembre de 2016. https://www.researchgate.net/publication/311983505_Conceptual_Design_Of_A_Low_Cost_Flight_Data_Acquisition_System_For_Analyzing_Flight_Behavior_Of_Small_Unmanned_Aerial_Vehicles
- Southeast Aerospace. (2008). *Collins Aerospace. Model CDU-7000D*. <https://www.seaerospace.com/sales/product/Collins%20Aerospace/CDU-7000D>
- Unidad Administrativa Especial de Aeronáutica Civil (Aerocivil). (2023). *Publicación de información aeronáutica - AIP*. <https://www.aerocivil.gov.co/servicios-a-la-navegacion/servicio-de-informacion-aeronautica-ais/aip>
- Wang, P., Wang, W., Zhang, Z., Chen, M., & Yang, J. (2020). *Civil Aviation University of China*. <https://www.cauc.edu.cn/en/>
- Yorobotics. (2021). *Usb Host Shield 2.0 MAX3421E Google Adk Uno Mega*. <https://yorobotics.co/producto/usb-host-shield-2-0-max3421e-google-adk-arduino-uno-mega>